

## A Turnkey Solution for A VSTLF Model Using Artificial Neural Networks and Microcontrollers

G. Manuel

*Nuclear Energy Corporation of South Africa South Africa*

---

**Abstract:** - Neural networks are by nature parallel computational algorithms used to address two types of problems. Namely the classification problem and forecast problem. The availability of low cost and high speed microcontrollers with peripheral rich features, specifically peripherals in support of communication mediums have led to a more feasible implementation of the neural network paradigm to address the very short term load forecast problem. The paper proposes a turnkey solution incorporating a neural network algorithm onto a microcontroller platform to address a very short term load forecast problem.

**Keywords:** - *Artificial Neural Networks, Feedforward, Backpropogation, Microcontroller, Short Term Load Forecast*

---

### I. INTRODUCTION

Neural networks are used two address two forms of paradigms. Namely the classification problem and forecast problem (1) (2). For the forecast paradigm, the future load is a correlation of the loads most recent behaviour and factors that affect the load (3). This gives rise to the necessity for real time data processors that are able to process the most recent data. By accounting for the loads most recent behaviour, the future load may be predicted with an acceptable error. Neural networks do not have the same short comings of parametric or statistical techniques whereby the future load is determined by a correlation of the magnitude of the load elements for which the load is derived. Rather the final behaviour or rhythmic cycles is used to correlate the future load. Depending on the type of parametric technique, the load is modelled by a series of mathematical equations (most commonly in terms of a Fourier series: sines and cosines) (4). The load may be a stochastic or deterministic time variant load and is modelled accordingly (5). Non parametric equations eliminate the need of evaluating the load to the degree of knowing whether or not it is a stochastic load in which case it has to be brought a stationary process (6), deterministic load, a linear load or nonlinear load curve. By categorising the load for instance in the case of a regression approach formulated by Equation 1, the load elements for which the load comprises of is modelled in order to forecast a predetermined time ahead window (forecast) (7).

$$z(t) = b(t) + \sum_{i=1}^n a_i y_i(t) + \varepsilon(t) \quad (1)$$

Where

$b(t)$ : Potential load

$\varepsilon(t)$ : White noise aspect

$y_i(t)$ : Weather variable

Modelling the load elements using the regression technique requires more data to be acquisitioned than what is required when only the load and not necessarily the elements for which defines the derivation of the load is used for the forecast model as in the case of artificial neural networks.

This emphasise two types of approaches, namely a load that is modelled using the source and a load that is modelled using the factors that determines the load changes. In order to provide a real time forecast, the computational requirements such as processing speed play a significant role for the Artificial Neural Network (ANN) algorithm in terms of accounting for the loads most recent behaviour. The ANN model is inherently parallel in characteristic and therefore devices such as Programmable Logic Devices (PLD's) are ideal for meeting the processing requirements (8). However the ability to communicate with data acquisition devices serves the utmost importance. Data acquisition devices log and monitor physical entities (power, current, power factor, weather variables, etc.), These physical parameters are necessary when performing a future prediction of trend or forecast. Several microcontrollers (Atmel, Microchip, and Infineon) are available at a relatively low cost with many supporting peripheral features (9). Most importantly are the peripheral attributes that support various communication mediums. Common communication mediums supported by such devices include the Universal Synchronous Asynchronous Receive Transmitter (USART) used for communicating serially in

RS485 and RS485, Universal Serial Bus (USB), Inter-Integrated Circuit (I<sup>2</sup>C), Serial Peripheral Interface (SPI) and TCPIP. Peripheral attributes found within an embedded device minimise the need for additional components (therefore increasing the size and footprint of electric circuit boards) by including the functional circuit and registers within the wafer of the silicon or germanium doped device. Having a peripheral rich device eliminates the need for additional components as in the case of employing an FPGA or similar high speed devices of parallel type architecture.

A turnkey solution is ideal, more specifically one that supports the algorithm, processing requirements and communication medium necessary to obtain data from peripheral devices. In the context of a Very Short Term Load Forecast (VSTLF) algorithm, it easily concluded by simply evaluating the regression formulae (Formulae 1) that weather stations are needed to acquisition data for weather variables as in the case of a parametric techniques. When considering the real time data scenario, speed plays a critical factor. Evaluating a system level design whereby single board computers and servers incorporate operating systems that limit the full potential speed in terms of the execution of the algorithm by committing to threads that do not necessarily support the algorithm is not ideal. Field Programmable Gate Arrays (FPGA) and microcontrollers do not suffer the high overheads in execution of the ANN STL algorithm in comparison to processing devices that harbour operating systems. In this paper, a microcontroller is subscribed to executing a VSTLF ANN algorithm with the full computational resources available to the execution of the algorithm. For this test, data from a railway operator was used to predict the next hourly load. A Microchip microprocessor from the 24F family series (16 bit) was used to execute the algorithm. Back propagation was used to train the supervised ANN model to reduce the error and offer a more accurate next hour forecast. The model outputs were compared to actual values and yielded acceptable results. The percentage error varied between 0% and 19%.

## II. ARTIFICIAL NEURAL NETWORKS

For Figure 1, the basic fundamental topology of a multi layer perceptron artificial neural network is illustrated. The ANN network illustrated in Figure 1 consists of a four nodes within the input layer (denoted by  $i_1..i_n$  where  $1 \leq i \leq 4$ ), four within the hidden layer (denoted by  $h_1..h_n$  where  $1 \leq i \leq 4$ ) and one node in the output layer. Bias nodes are illustrated by  $b_{h1}$  and  $b_{o1}$  within the hidden and output layer respectively. These nodes offer stability to the artificial neural network when all inputs are zero. Bias nodes have no inputs and therefore the output of the bias node are the product of the weight and high logic (one). The forecast model works by formulating a correlation between the previous load values and the future load values. In order to obtain the next load value at time t, sample data is used in order to train the ANN, to perform an acceptable correlation. This is done by adjusting the value of all weights found between the input and hidden layer and hidden to output layer. The neural network is initialised by setting all weights to random values. It is assumed at this stage that there is a variance between the expected load outcome and what will be predicted by the untrained ANN. The error exist somewhere on a curve or plane. This value of this error is determined by taking the input values and in a feed forward manner the output of the network is determined. For each node, the net output is the product of all inputs and weights. This is formulated by Equation 2 (10). For each node found within the hidden and output layer, the actual output is derived by passing the net output through an activation function.

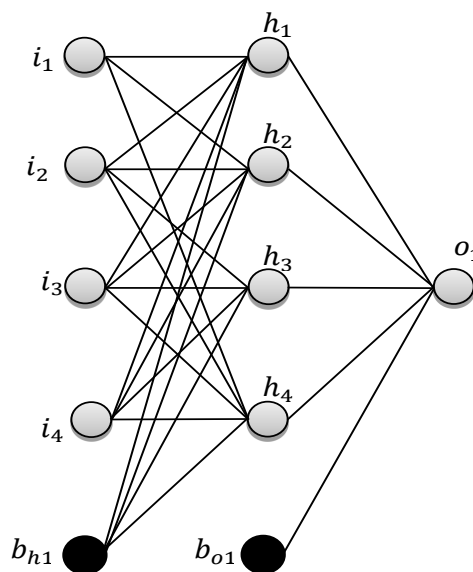


FIGURE 1 THE ARTIFICIAL NEURAL NETWORK TOPOLOGY

$$x = \sum_{i=0}^{i=n} x_i w_i \tag{2}$$

Where:

- $x_i$ : The associated node input with reference to the corresponding weight
- $w_i$ : The corresponding weight
- $f(x)$ : The net output of the node

For this particular model, the most common function was used, namely a sigmoid function formulated by equation 3 (11) as an activation function.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

Where

- x: node net output

The sigmoid function is asymptotically bounded to between one and zero. For this reason all data inserted into the neural network has to be normalised. The data is usually normalised in terms of percentage change from the base demand. The problem experienced in the load curve illustrated by Figure 5 was that the minimum load was zero kw due to the fact that the traction power was switched off during maintenance periods. In order to normalise the data, the incoming values into the neural network was divided by a constant. In this case, the constant was 10000.

Finally, the error is then calculated by the difference between the output value (output of neuron  $O_1$ ) and ideal value given by (here the absolute value is formulated):

$$E = |d - f(O_1)| \tag{Error! Bookmark not defined.}$$

Where:

- $E$ : The Error
- $d$ : Ideal or desired output
- $f(O_1)$ : Actual output derived from node  $O_1$  (illustrated in Figure 1)

The error is reduced by means of the Back Propagation Algorithm (BPA) that seeks to reduce the error by adjusting the weights. The BPA employs the delta rule that is relevant to the method of gradient descent. The first stage in the BPA is to calculate the layer delta for the output node (equation 5) and hidden nodes (equation 6). These are calculated respectively by:

$$\delta_{O_1} = -E f'(x) = -E \frac{e^x}{(1 + e^x)^2} \tag{4}$$

$$\delta_{h_n} = f'(x) \cdot \sum w_{ki} \cdot \delta_{O_1} \tag{5}$$

Where

- $\delta_{O_1}$ : The output delta
- $\delta_{h_n}$ : The hidden layer delta
- $E$ : The error derived by equation 4.
- $f'(x)$ : The derivative of the output function derived by equation 3.
- $x$ : The net output value derived by equation 2
- $w_{ki}$ : The respective hidden layer weights where  $k$  refer to the hidden layer number (when multiple

hidden layer is used) and  $I$  which serves as the node reference number. (For illustration 1:  $1 \leq i \leq 4$ )

For the next phase, the gradient is calculated for each node:

$$G = \delta_{ki} \cdot f_{ki}(x) \tag{6}$$

The amendment to each weight is then calculated by:

$$\Delta w_t = \epsilon G \tag{7}$$

Where

$\epsilon$ : The learning rate

The learning rate implies a fractional weight adjustment of the full degree of weight change recommended by equation 8. A high learning rate value will result in the ANN learning to rapidly and as a consequence maintain an inadequate magnitude of error. If too small learning rates are used, the ANN algorithm will learn too quickly. For STLf a sequence of values is presented as inputs to the ANN. A context layer is added to the recurrent neural network model as illustrated in Figure 2.

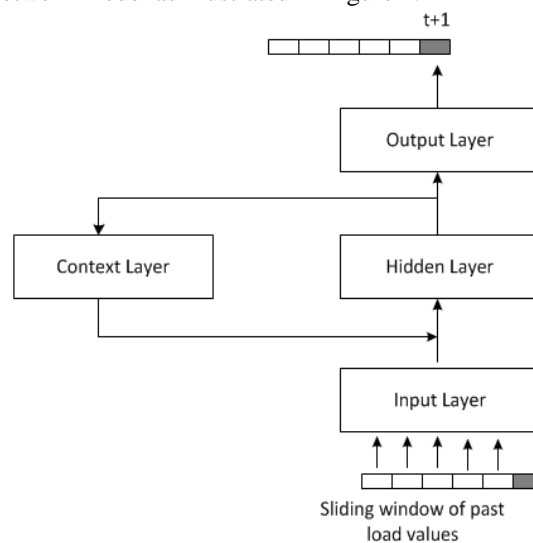


FIGURE 2 CONTEXT LAYER WITHIN THE ANN MODEL

This layer takes the inputs from the output of the hidden layer and in turn provides this historic data into the hidden layer when the next pattern is presented. This leads to previously presented inputs to the neural network to affect the values of the nodes when the new inputs are presented, providing a form of context for the neural network. Within the forecast model there exist two windows. The first window captures the current and past behaviour of the load. This data is used to predict data found within the forecast window or period. A window of past values provides the correlation of future values. The most recent forecast data will be shifted into the present time as denoted by  $t$ . With regards to the railway operator load data, after each hour the forecast is made and corrected when the time series shift to make future values current values. The corrective action is achieved by means of the back propagation learning algorithm. Figure 3 illustrates the variance or error between the forecast data for a twenty four period and actual data. High error values were investigated (in excess of  $E > 10\%$ ) and was discovered that unpredictable behaviour within the historic window led to these high error values (erratic behaviour not within past time series). Figure 4 illustrates actual values for a weekday. For Day1 (D1), Day2 (D2) and Day3 (D1) the trend is fairly similar. In the first few hours in the morning, the upwards trend for D4 does not explicitly or inexplicitly resemble D4. The major advantage of artificial neural networks is the ability to learn patterns not explicitly presented. However the historic data has to form the most optimum correlation between the time series and load values. The inputs for the artificial neural network may be derived from data captured from the communications with standalone systems or sub systems. There are two forms of data acquisition systems, namely, online and offline systems. Online systems are connected to data acquisition devices that capture data in real time. Offline systems receive data through an USB port, Ethernet port or any other form of communication medium from a device used to store data.

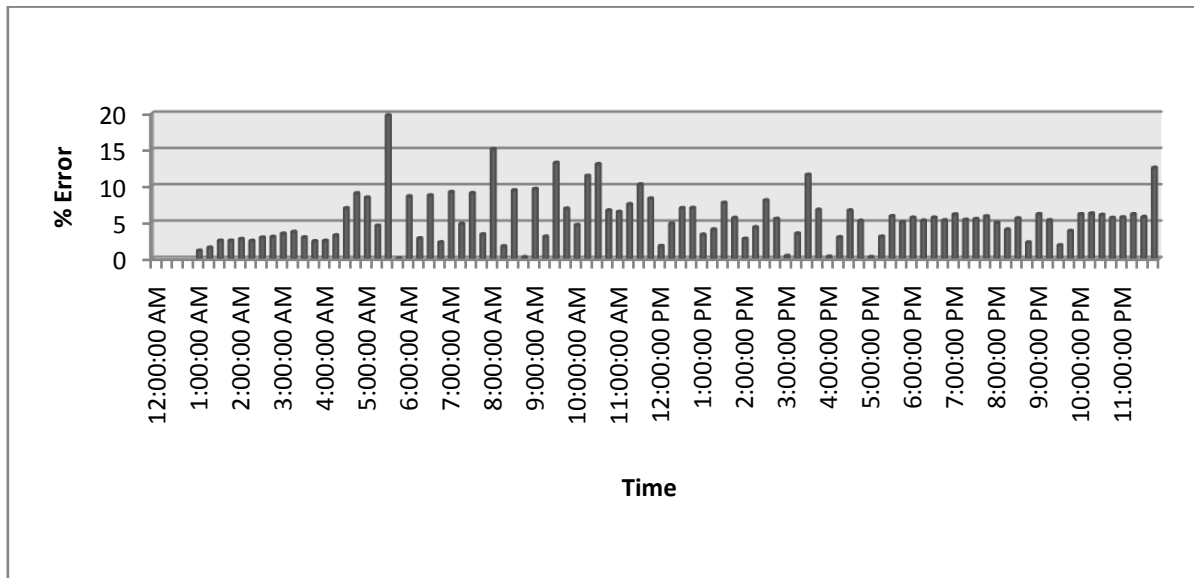


FIGURE 3 RESULTS OF ANN FORECAST MODEL USING A MCU (WEEK END PROFILE)

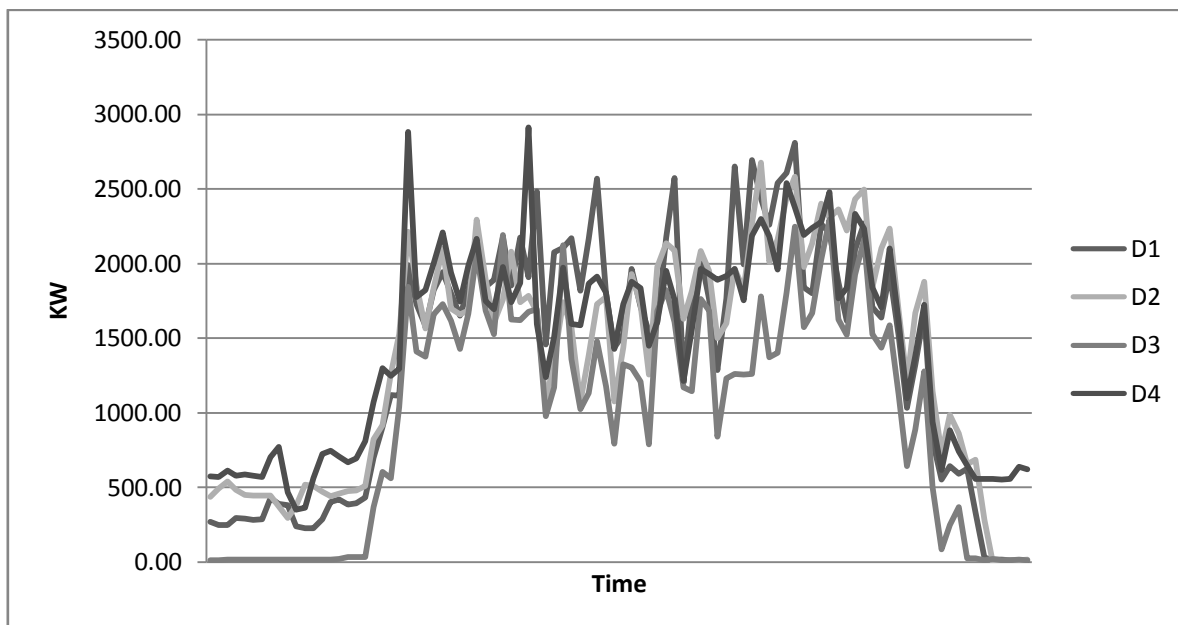


FIGURE 4 ACTUAL LOAD DATA

### III. MICRO CONTROLLERS AND ARCHITECTURES

Microcontrollers are inherently sequential in nature. The sequential timing of the execution of instructions is undertaken by an oscillator. This is referred to as the machine cycle or clock cycle. An instruction cycle may take up to two or four clock cycles to execute an instruction.

$$t_{instruction\ cycle} = P \times t_{clock\ cycle}$$

Where

Clock cycle: For which time is derived from the oscillator

Instruction cycle: The minimum amount of time taken to execute an instruction requiring the least amount of clock cycles. Hence the reason why some instruction may take more than one instruction cycle.

P: (Pipeline flow) An integer that is usually specified by the manufacturer for the amount of clock cycles to complete an instruction cycle.

Microcontrollers are designed based on either of the Von Neumann type architecture (Illustrated in Figure 6) or the more common Harvard Architecture (Illustrated in Figure 5). For a Harvard Architecture, separate busses are used for data transfers and instruction fetches. For the Von Neumann architecture a single bus is used for both data transfers and data fetches. The data paths, widths, integer sizes and addresses are available in 8 bit, 16 bit, 32 bit and 64 bit.

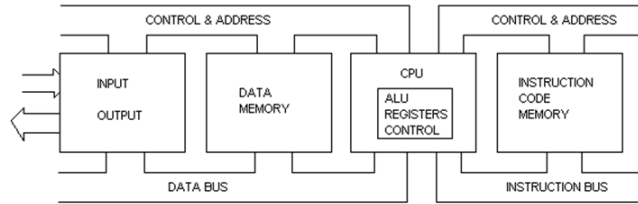


FIGURE 5 THE HARVARD ARCHITECTURE (12)

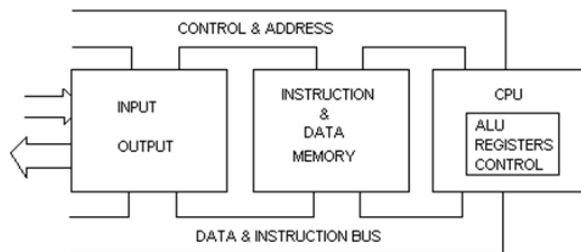


FIGURE 6 THE VON NEUMAN ARCHITECTURE (12)

Due to the sequential nature of microcontrollers, the execution of an instruction will have a greater latency in comparison to devices that are not sequential. Programmable logic devices are examples of devices that are parallel in nature and do not suffer from a pipeline flow attribute (illustrated in Figure 7).

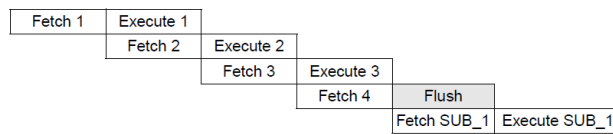


Figure 7 Pipeline flow for a microcontroller (13)

Hence why PLD's would process the data faster than a microcontroller would. Code optimization also plays a vital role. A poorly PLD's synthesized solution may match the efficient operational speed of an optimally written embedded code targeted for a microcontroller executing the ANN algorithm. Below are the test criteria and results of the Microcontroller executing the ANN forecast algorithm:

Random Access Memory (RAM) Used:	16.384 Kbytes (7% of total memory)
Flash/Program memory	87.548 Kbytes (2% of total memory)
Processor Frequency:	48 Mhz
Simulated device:	PIC24FJ256GA110
Optimization methods:	None
Compiler:	XC16 C Compiler
Software:	Mplab X
Highest percentage error forecast value:	19.72135
Lowest percentage error forecast value:	0.055289
Learning Rate:	0.7
Program execution time:	25.736354 ms

Table 1 Test Results and Criteria

The performance of the microprocessor to execute the algorithm was simulated within the Mplab simulation environment. The clock cycle was simulated at 48Mhz and the total time to execute the learning

phase and the forecast was 25.736ms. The percentage error varied from between 0 to 19%. The Mplab is An Integrated Development Environment (IDE). Within the simulator and In Circuit Debugger, the microchip performance was tested. The railway operator required data to be logged every fifteen minutes (sample time). However the microprocessor was able to process the data within 25ms.

#### **IV. CONCLUSION**

The study evaluated a VSTLF using a microprocessor as a processing element for the execution of the algorithm. Generally microcontrollers are available at a relatively low cost with peripheral rich features that make implementing an embedded approach to solve the forecasting problem feasible. A turnkey solution consists of two aspects, the algorithm and the hardware to run the algorithm. In terms of the algorithm, the forecast of an hour ahead used a three layer ANN. The ANN learnt from the data presented from the previous hour to predict the next hour data. The back propagation algorithm was used and the highest percentage error obtained was near 20%. The lowest being zero. The ANN VSTLF model was capable of predicting the hour ahead with an acceptable degree of accuracy. In terms of the hardware, the ANN executed the program within 25ms. The service provider required that data be provided for the next hour in intervals of 15 minutes. Therefore the processing power that the processor offered was more than adequate. The global trend is to have parallel devices such CPLD's and FPGA's that exhibit parallel architectures to execute an ANN based VSTLF model. However for low sampling time requirements, microcontrollers are a more feasible option in addressing the solution.

#### **BIBLIOGRAPHY**

- [1] 1. Neural classification approach for short term forecast of exchange rate movement with point and figure task. Stahlbock, R. s.l. : IEEE IJCNN, 2008.
- [2] 2. Forecasting with artificial neural networks: The state of the art. Guoqiang, Zhang, Patuwo, B Eddy and Hu, Michael Y. 1, s.l. : International Journal of Forecasting, 1998, Vol. 14.
- [3] 3. Weather sensitive short term load forecasting using nonfully connected artificial neural network. Chen, S, Moghaddamjo, A R and Yu, D.C. 3, s.l. : IEEE Transactions in Power Systems, Vol. 7.
- [4] 4. Forecasting Portugal global load with artificial neural networks. Fidalgo, J Nuno and Matos, A Manuel. s.l. : ICANN 2007: Lecture notes in Computer Science, Vol. 4669.
- [5] 5. Short term load forecasting for the holidays using fuzzy linear regression method. Song, Kyung-Bin, et al. 1, s.l. : IEEE Transactions on Power Systems, Vol. 20.
- [6] 6. The time series approach to short term load forecasting. Hagan, T Martin and Behr, Suzanne M. 3, s.l. : IEEE Transactions on Power Systems, 1987, Vol. 2.
- [7] 7. Analysis and evaluation of five short term load forecasting techniques. Moghram, I and Rahman, S. 4, s.l. : IEEE Transactions on Power Systems, 1989, Vol. 4.
- [8] 8. Neural Networks and Fuzzy systems. Kosco, Bart and Burgess, John C. 6, s.l. : Acoustical Society of America, Vol. 104.
- [9] 9. Trends in embedded-microprocessor design. Schlett, M. 8, s.l. : IEEE Computer Society , 1998, Vol. 31.
- [10] 10. Output feedback control of nonlinear systems using RBF neural networks. Seshagiri, S and Khalil, H.K. 1, s.l. : IEEE Transactions on neural networks, 2000, Vol. 11.
- [11] 11. On Fuzzy modelling using fuzzy neural networks with the back propogation algorithm. 5, s.l. : IEEE Transactions on neural networks, 1992, Vol. 3.
- [12] 12. Computer Architecture. [Online] [Cited: 25 Feb 2012.] [http://www.electronics.dit.ie/~staff/tscarff/architecture/computer\\_architecture.htm](http://www.electronics.dit.ie/~staff/tscarff/architecture/computer_architecture.htm).
- [13] 13. Microchip. 8Pin, 8 bit CMOS Microcontrollers. PIC12C5XX: DS40139E. s.l. : Microchip, 1999.
- [14] 14. The significance of relevance trees in the identification of. Manuel, G and Pretorius, JHC. 1, s.l. : Journal of Energy in Southern Africa, 2013, Vol. 24.