

# Prediction of vehicle reliability performance using artificial neural networks

S. Lolas, O.A. Olatunbosun \*

*School of Mechanical and Manufacturing Engineering, University of Birmingham, B15 2TT, UK*

## Abstract

Product development is an important but also dynamic, lengthy and risky phase in the life of a new product. The optimisation of the product development phase through extensive knowledge of the involved procedures is believed to reduce the risks and improve the final product quality. Artificial intelligence and expert systems have been used successfully in optimising the development phase of some new products as it will be demonstrated by the first sections of this publication. This paper presents the first module of an expert system, a neural network architecture that could predict the reliability performance of a vehicle at later stages of its life by using only information from a first inspection after the vehicle's prototype production. The paper demonstrates how a tool like neural networks can be designed and optimised for use in reliability performance predictions. Also, this paper presents an optimisation methodology that enabled the neural network to deal with the limited amount of available training data, common during new product development, and to finally achieve acceptable prediction performance with small error. A case example is presented to demonstrate the methodology.

© 2007 Elsevier Ltd. All rights reserved.

**Keywords:** Neural networks; Product development; Reliability prediction; Knowledge

## 1. Introduction

It is widely accepted that the modern industries are under immense pressure to deliver high quality and reliable products at the minimum possible cost. In other words, manufacturers must seek to shrink, significantly, the product development phase, from design concept to complete product release. Higher quality levels, higher service levels, customisation and continuous innovation paired with lower production costs have been deemed as necessary since the 1980s but are no longer enough in the modern era. The extra requirements for the industries of today and the future are the creation and domination of new markets which can be achieved by the development of new products (Cohen & Levinthal, 1994; Hamel & Prahalad,

1991; Ozer, 1999). However, as Ozer (1999) has commented new product development is a dynamic, lengthy and risky business.

The optimisation of the new product development phase requires extensive knowledge about all the various procedures that are involved with it. Knowledge, in general, is a very important resource for presenting valuable heritage, learning new things, solving problems, creating core competences and initiating new situations for both individuals and organisations now and in the future (Liao, 2003). The management community, both academic researchers and practitioners, has developed a large number of technologies and applications to address the issue of knowledge acquisition, management and application. According to Liao (2003) there is a huge number of publications under the knowledge management index. All this literature presents a number of different approaches.

Artificial intelligence and expert systems is the method of choice for this research project, whose objective is to provide solutions to issues inherent to the product development

\* Corresponding author.

E-mail address: [o.a.olatunbosun@bham.ac.uk](mailto:o.a.olatunbosun@bham.ac.uk) (O.A. Olatunbosun).

of automotive vehicles. This paper only discusses early parts of the expert system development, currently under way by the Novel Vehicle Technology and Future Power System Group of the University of Birmingham.

Expert systems are knowledge based systems that contain expert knowledge, according to the definition given by Kasabov (1996). Expert systems have been applied successfully to most fields of science including engineering, science, medicine, agriculture and many more. The main advantages that make them so attractive for such a large variety of applications are that they can capture human knowledge in the form of a set of rules. Expert systems are not only capable of capturing knowledge but they also retain this knowledge therefore they form the organisational memory or in other words they store the learning of the organisation. This provides a very attractive benefit to all the organisations that employ expert systems. These organisations no longer rely solely on their human workforce to carry and pass the knowledge and the experience gained from older projects to their successors in the company. The organisation's expert systems incorporate this knowledge in the form of tools and therefore the acquired knowledge can be retrieved automatically during a future project when it is required by a different employee. Consequently expert systems eliminate the danger of wrongly understood or forgotten knowledge, something that is very possible when knowledge is retained by humans. Examples of more conventional expert systems are presented in Cunningham and Bonzano (1999) and Doyle et al., 1996. Examples of using artificial intelligence methods such as genetic algorithms, neural networks and fuzzy logic can be found in (Al-Thabtabai, 1998; Mohan & Arumugam, 1997). More applications of expert systems can be found in table 5 in (Liao, 2003).

This paper presents the first steps towards the construction of an expert system. The complete expert system will combine different types of artificial intelligence methods. The scope of this paper will concentrate on one of the modules of the expert system. This module comprises of a neural network system which was designed to process reliability data, learn it and then use this knowledge in forecasting and analysis.

## 2. The problem

Fig. 1 describes the real situation in the automotive industry for the reliability problems of a vehicle model during its development phase. The figure is created using the following methodology. For every problem noticed on a development test vehicle, points are awarded according to the significance, severity and other contributing factors, creating a final score that quantifies the reliability performance of the given test vehicle. Models tested represent time or stage in the development phase. The models closer to the concept vehicle are towards the left side of the axis and the models that are closer to job#1 can be found towards the right end. In real life (actual curve) vehicles

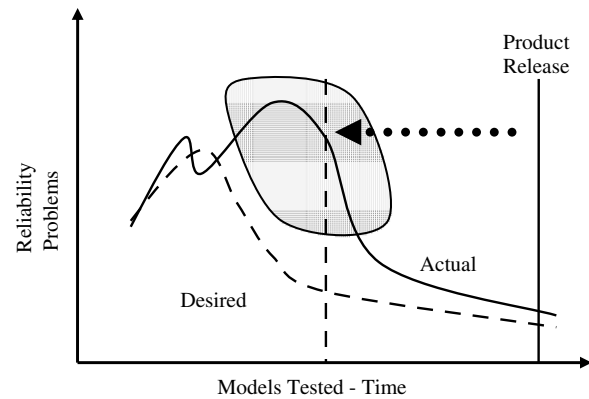


Fig. 1. Development of a new vehicle in terms of reliability. Reality versus Ideal.

towards the left side of the figure give a fairly good reliability performance because many of the components installed in them are carried over, without any modifications, from the older already reliable version. As the new components are introduced and installed in the test vehicles the reliability problems increase. Then, the reliability performance of the test vehicles improves gradually after a critical point is reached, mainly because the engineers react to the reliability problems that are pointed out from the testing procedure and correct them. From there the reliability performance gradually reaches the required levels and the vehicle is released to the market. This approach to the development phase of a vehicle can be characterised as passive, because corrective actions are made after the faults have been identified.

The “desired” curve in Fig. 1 represents the ideal scenario for the development of a new vehicle. Improvements in reliability can be achieved through a more pro-active approach to the procedure. A pro-active approach through the development phase requires good knowledge about it. Knowledge about the particularities of the new development procedure can be acquired by analysing past development phases of predecessors or similar products and also by using the experience of the experts that have been involved in the development of the predecessor systems. A detailed analysis can accentuate trends and effects of the various changes in the design and other parameters of the process which can then be monitored and learned, using artificial intelligence tools such as neural networks. By using neural networks the engineers would be able to simulate the development procedure for the new product and as a result they can identify the areas which need to receive extra attention and also the areas that do not contribute to the reliability problems of the new product. Therefore engineers can concentrate the available resources on significant reliability problems that have great contribution to the overall reliability performance, thereby optimising the development procedure. Another important benefit of using a pro-active approach for the new development procedure is that requirements and

reliability targets can be imposed on the sub-system and component suppliers. The “everybody plays” concept that has been created by General Electric Appliances (Antony & Banuelas, 2002) explains the positive effects the supplier involvement can have in the reliability improvement efforts.

There are many examples of world leading companies in manufacturing and engineering fields that have realised that the improvement of reliability and quality of their products can only be achieved through optimisation of all of their operations. This is the reason why a number of leading companies world wide use techniques like total quality control and six sigma to improve all aspects of their structure management and operation (McClusky, 2000; Antony & Banuelas, 2002; Catherwood, 2002; Johnson & Swisher, 2003; Kwak & Anbari, 2004). Despite the commitment of many companies to such programs the difficulty arises from the fact that reliability improvement of the product during the product development requires continuous reliability estimation and updating (Yadav, Singh, Chinnam, & Goel, 2003), that requires detailed knowledge about all of the specific details of the procedure. The lack of knowledge creates what is known as epistemic uncertainty, which according to Cizelj, Mavko, and Kljenak (2001, Cizelj) is defined as the lack of knowledge about a system or phenomenon.

### 3. Neural networks for reliability prediction

#### 3.1. Background and applications

The use of ANNs has grown in popularity during the last few years. The reason for this is that neural networks represent a novel and modern approach that can provide solutions in problems for which conventional mathematics, algorithms and methodologies are unable to find a satisfactory and acceptable solution. These problems are usually very complex and some of the mechanisms involved have not been fully understood by the researchers dealing with them.

ANNs are inspired by the human brain functionality and structure, which can be imagined as a network that comprises of densely interconnected elements called neurons. Despite this fact the ANNs’ objective is not to model it. Instead their purpose is to be useful models that can be used for problem solving and knowledge engineering, in a way that resembles the human process for problem solving and knowledge acquisition. Both biological and artificial networks have the following main and important features; learning adaptation, generalisation, massive parallelism, robustness, associative storage information and spatiotemporal information processing (Kasabov, 1996).

The operational manner of ANNs is that when inputs are applied to the input neurons the network performs a summation of the weighting factors and then it activates one or more specific output neurons that are capable of providing the most suitable answer for the given problem.

By training the neural network, implicit knowledge can be built into it (Pham & Pham, 1999). The training of a supervised neural network occurs by presenting typical input patterns and the corresponding expected output patterns. The strength or otherwise the weights of the connections between the neurons are modified by using an error measurement between the actual and the expected results, until the results of the network are satisfactory. For this procedure a backpropagation algorithm is used. It propagates the error, between the expected and actual results, backwards through the structure of the network and then it computes the weight modifications necessary to improve the actual results of the network’s outputs in order to provide the most correct solution to the problem. Multilayer perceptron is the most popular neural network. A multilayer perceptron is a feed forward network, which can perform static mapping between input space and an output space. It consists of neurons organised in a number of layers that can be categorised into three parts. The first part is the input layer that allows that network to communicate with the environment, the second part is commonly known as the hidden part, where one, two or more layers of neurons exist depending on the problem’s demands and generalisation requirements. The last part of the multilayer perceptron is the output layer through which the network presents its answers to the user. A schematic diagram of a basic multi-layer perceptron and its learning procedure can be seen in Fig. 2.

Some examples of ANN applications in various industries are Zhang and Friedrich (Zhang, 2003), who used an ANN in the composite material science in order to assist in the selection of suitable composites according to the requirements of the application. Yin, Rosendahl, and Luo (2003) used ANN in the coal industry, to model the relationship between the ultimate analysis data and its proximate analysis data, which are both used in order to classify coals. Other interesting ANN applications of multi-layer feed forward backpropagation neural networks include (Nascimento et al., 2000; Pleune and Chopra, 2000; Venkatesh and Rack, 1999; Zhang, 2003). All these applications demonstrate the superiority of the backpropagation ANNs in solving difficult and complex problems, which could not be solved at all, or could be solved only by expensive and time consuming testing.

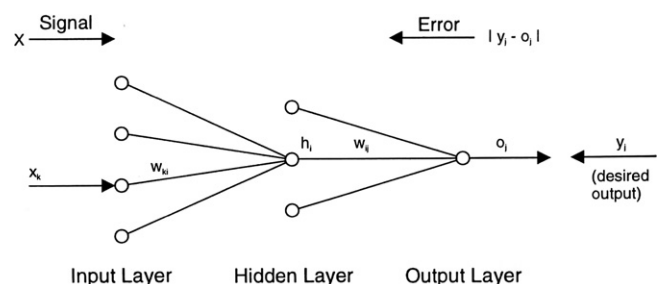


Fig. 2. Multi-layer perceptron learning example (Kasabov, 1996).

### 3.2. Objectives and description of the proposed ANN

During the current project an ANN was created in order to assist the engineers in predicting the reliability behaviour of an automotive vehicle during its development phase. Due to the complexity of the procedure due to the high number of parameters and influencing factors that are involved, an accurate mathematical model without a great number of assumptions is not possible. Many assumptions usually result in reduced accuracy.

In more detail the proposed network was designed in such a way that it could predict the behaviour of the vehicle at 6000 km by using only information about the vehicle's reliability issues at 0 km. This practically means that an initial inspection, during which the initial reliability and quality issues of vehicle are recorded, will provide enough information to simulate this vehicle's performance at 6000 km. If data for higher mileage ranges is included then the trained network will be able to predict the reliability degradation for the vehicle for a later stage of its life.

### 4. Data collection and data nature

One of the most important stages in the design of a supervised ANN is the data collection and data preparation, thus the examples for training must be representative of all the possibilities concerning the application. Researches that have used ANNs with supervised learning support the previous statement (Yin et al., 2003; Zhang, 2003).

The data used for this work has been extracted from a number of databases, which record reliability and quality issues. The context of the data used was in terms of reliability evaluation audit (REA), which records problems on the vehicle's systems and assesses points according the severity and importance of the problems. Four inputs into the network which represent the behaviour of the vehicle at 0 km were initially extracted from the databases. More precisely the inputs into the system are the code number of the vehicle, which could preserve the chronological point in the development phase at which the vehicle was tested, the total number of issue score points for this vehicle, the number of issues that appeared on this vehicle and finally the average score for each issue.

Three target parameters or in other words three desired answers will be used to demonstrate to the network what the correct answer is to any given input parameter. These parameters are the total REA points, the number of issues and the average number of points per issue all at 6000 km.

At the end of the data collection there were 29 vehicles that have been tested and had satisfactory information available in terms of the parameters stated above. The amount of the available data could be judged to be very small and could potentially cause network generalisation problems. Adding to the problems was the fact that not all the data from the 29 cars could be used to train the

network. A small amount of this data was reserved to create validation and testing data, whose use is described later. Thus a very thorough network optimisation procedure, including training, backpropagation, neuron numbers, hidden layer numbers and other was undertaken in order to achieve the best possible results.

## 5. Network design and optimisation method

### 5.1. Overview

The main objective of the work described in this section was carried out to discover which one of the many different feed-forward backpropagation neural network combinations and architectures would be most suitable for the task and also the network architecture that would be able to deal with the small amount of available training data. The design and optimisation of the network was carried out in three main stages. In brief, during the first stage the fundamental parameters for the network, like the transfer functions and the training functions have been tested. During the second stage the type of pre-processing method more suitable for the network has been tested. Finally, the third stage of the network optimisation has been carried out to discover the most suitable number of hidden layers and the number of suitable neurons in each. It has been suggested by a number of researchers that have used ANNs that the selection of all the parameters that have been described above is a trial and error procedure (Lee, Almond, & Harris, 1999).

### 5.2. Network optimisation – stage 1

During the first stage of network optimisation the fundamental parameters of the network have been tested to investigate which one gives the best performance. The fundamental parameters for the network are the transfer functions for each layer and the training functions.

The transfer functions that have been used for this group of networks were the *tansig* function, for the hidden layer and the *purelin* function, for the output layer. Both are very common choices for this type of ANN (Vogl, Mangis, Rigler, Zink, & Alkon, 1988; Hagan, Demuth, & Beale, 1996; MathWorks, 2004).

The training algorithm selection is a more complicated task than the selection of the transfer functions. There is a large number of training algorithms that can produce a faster, more accurate and more reliable network. The main complication in the selection of an appropriate training algorithm arises from the fact that the performance of all the available training algorithms depends on the specific problem under consideration (Hagan et al., 1996; MathWorks, 2004). A training algorithm that performs well in a function approximation problem might not perform very well when applied to a pattern recognition problem. Therefore, all the main training algorithms that, according to reference (MathWorks, 2004), have the best performance over



Table 1  
Network code names and the corresponding training algorithm name

Network code name	Training algorithm full name	MATLAB code	Acronym
Net1	Levenberg–Marguardt	Trainlm	LM
Net2	Baayesian Regularisation	Trainbr	BR
Net3	Resilient Backpropagation	Trainrnp	RP
Net4	Scaled conjugate Gradient	Trainscg	SCG
Net5	BFGS Quasi-Newton	Trainbfg	BFG
Net6	Variable Learning Rate Backpropagation	Traingdx	GDX

a number of different problems, were tested in order to discover the ones that performs better for this specific application.

Six training algorithms were tested. The different networks created with each different algorithm were given code names such as net1, net2... net6. The training algorithms with their corresponding code names are presented in Table 1. Initially, for the testing of all the networks the same number of neurons in the hidden layer has been used. There is no specific rule for the selection of the number of neurons in the hidden layer or the number of hidden layers that will produce the optimum results for a given problem. The number of neurons in the hidden layer is usually determined empirically, despite the fact that there have been numerous methods proposed by various researchers (Kaminskyj & Materka, 1995; Kang & Song, 1997; Lee et al., 1999). Due to the obvious variance in the opinions regarding the number of neurons in the hidden layer and the number of hidden layers these numbers will be decided after trials. For stage 1 the selection of the neurons in the hidden layer was decided by using the rule of thumb, therefore twelve neurons in the hidden layer were selected. The initial network configuration will be 4-[12]-3. This code means four neurons in the input layer, twelve in the hidden layer and three in the output layer. This configuration has been used for the trials of the six different training algorithms. The results of the trials are depicted in Fig. 3, in terms of mean absolute error. X-axis represents the number of trials that have been carried out for each one of the six different networks. Fifteen trials for each network have been carried out because the responses of the networks were not stable, meaning that the performance of the same network was varying from one training session to another. In Fig. 3 net1 appears to be the one that produces the more consistent results when compared with the other networks. It also gave the lowest error values. The second best performance in terms of singular minimum error has been achieved by net3, but this network appeared to have a great difficulty in achieving this performance consistently. Net2 appeared to be fairly consistent and was very close in terms of performance with net1. Overall, the values of error in all the three assessing parameters were not very good and even the best performing networks were not able to predict the behaviour of the vehicles at 6000 km at a satisfactory level.

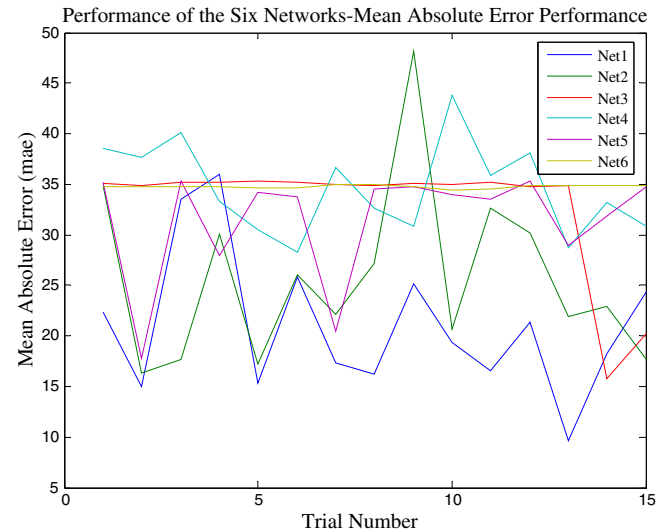


Fig. 3. Performance of Networks in terms of mean absolute error.

Consequently, there was the need for improvement in the networks that would increase their prediction quality and would reduce the error values.

### 5.3. Network optimisation – stage 2

Suggestions in a number of publications such as (Hagan et al., 1996; Al-Assaf & El Kadi, 2001) reinforced the opinion that the instability and the poor performance of the networks tested in stage 1 was due to the fact that the input values were high, ranging from 0 to 500. Al-Assaf and Kadi used a combination of normalisation techniques for the inputs. For this project a technique called minimum and maximum technique, has been used as in (Hagan et al., 1996; MathWorks, 2004). In the later two references another two data pre-processing techniques have been proposed. These are the mean and standard deviation method (MMM) and the principal component analysis (PCA) (Ishihara, Ishihara, Nagamachi, & Matsubara, 1997; Di Natale et al., 1999; Briquieu, Gottlieb-Zeh, Ramadan, & Brulhet, 2002; Lacassie, Roser, Ruiz Del Solar, & Herve, 2004).

Net1 and net2 have been selected for training with the pre-processed data from stage 1 because they performed better and more reliably during this first trial session of stage 1. Net2 with the Bayesian regularisation algorithm was expected to work much better with pre-processed data

Table 2  
Network code description for stage 2

Pre-processing method	Net1	Net2
Min–Mix Method (MMM)	Net1_1	Net2_1
Mean and standard deviation method (MSDM)	Net1_2	Net2_2
Principal component analysis (PCA)	Net1_3	Net2_3

according to (MathWorks, 2004). A new coding system was created to facilitate the separation of the different pre-processing methods. This coding system is presented in Table 2. It must be mentioned that the networks presented in Table 2 preserved the main characteristics that have been explained in Table 1. The networks have the same training algorithms and their structure remained the same, 4-[12]-3.

In Fig. 4 the response of the Levenberg–Marquardt network or net1 to the three pre-processing methods is demonstrated. The network responded positively to the pre-processed data. The performance of the network has been greatly improved. The MMM pre-processing method has resulted in a much better performance, in terms of consistency and the lowest error performance.

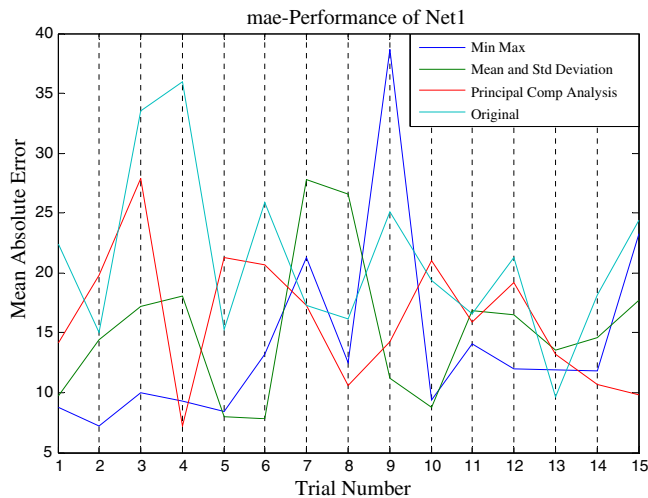


Fig. 4. Mean Absolute Error performance of net1 with all the pre-processing methods.

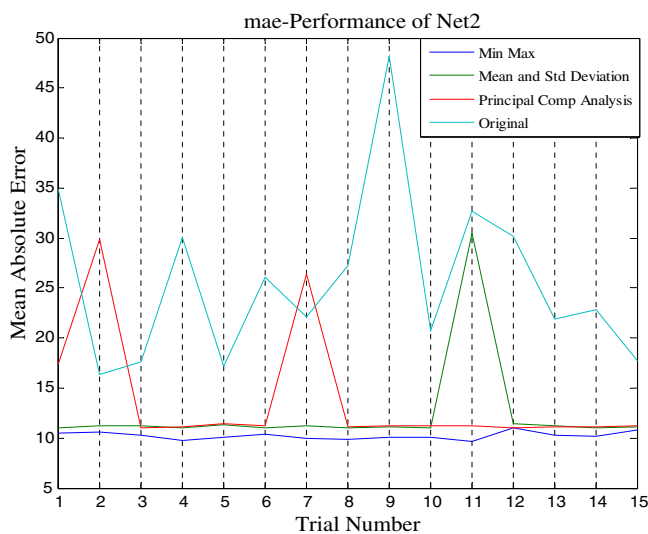


Fig. 5. Mean Absolute Error performance of net2 with all the pre-processing methods.

The effects of the three pre-processing methods on the Bayesian Regularisation method, namely net2, are illustrated in Fig. 5. It is apparent that the data manipulation prior to its application to the network has a very positive effect on all aspects of its performance. The error values dropped and in almost all the trials the response of the network was stable and consistent with small variations. The best performance in terms of lowest error values and consistency have been achieved by the MMM. Therefore this method has been chosen for the rest of the trials in the following stages.

Fig. 6 depicts the comparison between the two networks, net1 and net2, when the MMM is applied. The lowest error value is achieved by net1 but its response in terms of consistency is not as good as that of net2.

Summarising the results from stage 2, the two networks, net1 and net2 have responded very well to the introduction of the pre-processed data. The error values have been greatly reduced in both cases and also the consistency of the networks with respect to the network response has been improved. Despite the great improvements that have been achieved by the introduction of the pre-processed data there is the need for further improvements in order to reduce the error further.

#### 5.4. Network optimisation – stage 3

This optimisation stage has focused on testing different neural network architecture structures by altering the number of neurons in the hidden layer and the number of the hidden layers.

The network that was used to assess the differences in the performance of the network with respect to its different architectures was net2\_1 selected from stage 2. The reason behind this selection is that this network had a very consistent and stable response at every training trial.

##### 5.4.1. One hidden layer

The default error assessment parameter for the Bayesian Regularisation training algorithm is the sum squared error (sse). Net2 uses this training algorithm and as a result the assessment parameter that has been used in this stage is the sse. There have been two different values for sse. The first was the minimum sse value achieved during the training, according to which the training is stopped. The second was the total sse value, which calculates all the results collectively. One new assessment parameter has also been used in this stage. It is called regression performance of the network which is the result of data post-processing, where a regression analysis between the network's response and the desired response is performed. The regression results are represented by values between 0 and 1 where a value of 1 indicates perfect correlation between the targets and the actual outputs of the networks and 0 indicates the opposite. After the end of the trials it appeared that the regression values for two of the output parameters, the total number of REA points and the number of issues,

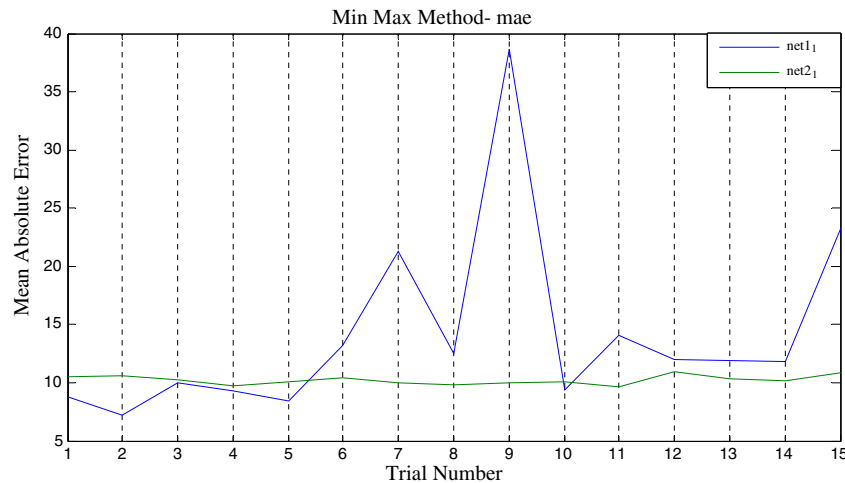


Fig. 6. Comparison of the error performance between net1 and net2 when using the MMM.

were always above 0.9 and more precisely 0.95 and above. The regression value for the third parameter, which is the average number of REA points per issue, was not very good taking values of 0.4–0.6. Therefore, the regression value for the third output parameter is the important one that indicated the best performing combination.

Starting from the network with one hidden layer the number of neurons has been changed taking values from 1 up to 30 neurons. In Fig. 7 the lowest min sse value is achieved by the network with 10 neurons. In terms of regression performance, Fig. 8, the network with fifteen neurons in the hidden layer manages to correlate the values of the third parameter better than any other. The difference in the sse values of the networks that have between 6 and 15 neurons in the hidden layer is not as great, which means that overall the network with a number of neurons in its hidden layer within the range of 10–15 will give much better results. These results suggest that the rule of thumb for the number of neurons in the hidden layer was correct.

#### 5.4.2. Two hidden layers

The networks have been modified accordingly to incorporate two hidden layers in their structure. Some random trials to specify approximately the best combination of neurons in each hidden layer were carried out. The results of the different layer and neurons combinations are presented in Figs. 9 and 10. In the figures the three final layers of the networks are represented because the number of inputs is fixed. For example 5\_2\_3 means that the network is a 4-[5-2]<sub>2</sub>-3 network. The figures indicate that the network 4-[5-5]<sub>2</sub>-3, (5\_5\_3), with five neurons in the first hidden layer and five neurons in the second hidden layer, had the best performance with all the assessment parameters i.e. both min sse value and high regression. Overall it was apparent that the error values were reduced by a great amount and the regression values, especially for the third output parameter, was improved and reached values over 0.8. In the one layer networks the regression values for this parameter was not more than 0.78. These results

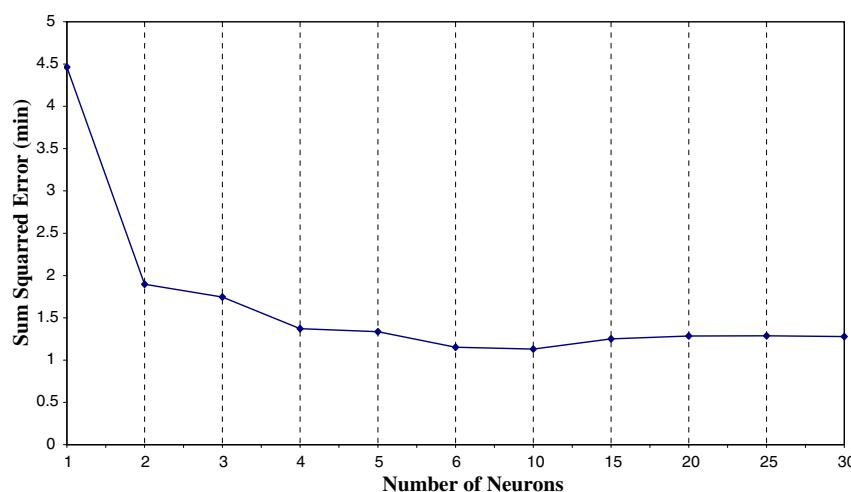


Fig. 7. The performance of the network for a range of neurons in the hidden layer.

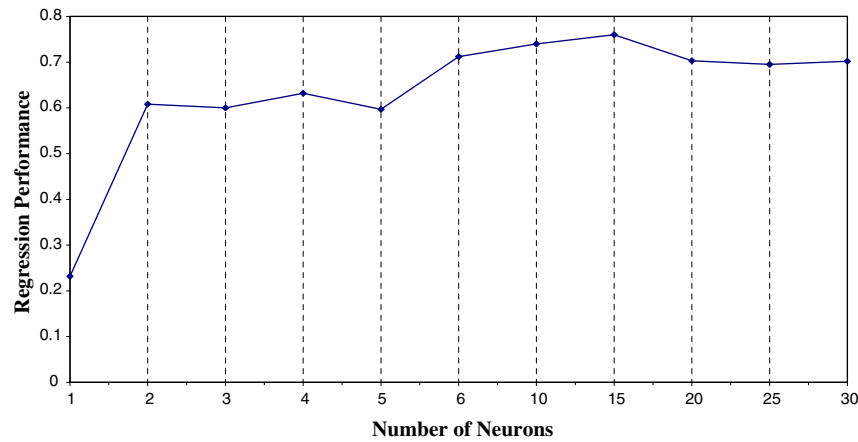


Fig. 8. Regression performances for the third network output value (Number of average points).

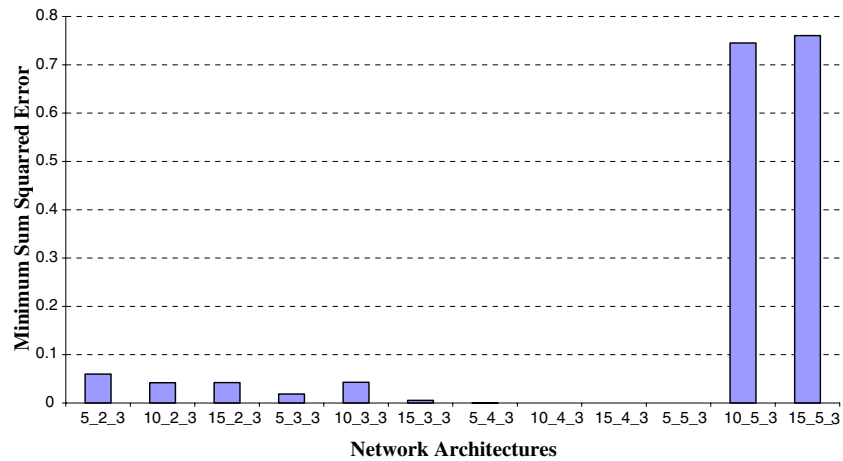


Fig. 9. Collective results for the minimum sse performance of networks with two hidden layers.

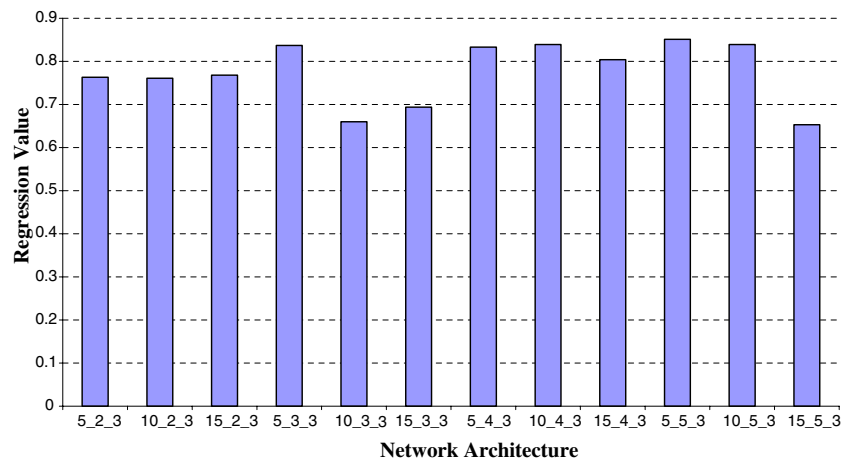


Fig. 10. Collective results for regression performance of networks with two hidden layers.

demonstrate that the target of achieving a very good prediction level has been reached. This means that the 4-[5-5]<sub>2</sub>-3 network will be used to perform similar tasks in

the future. Figs. 11–13 present the predicted results for this network, which gave results that are much closer to the expected behaviour of that of the real vehicle.



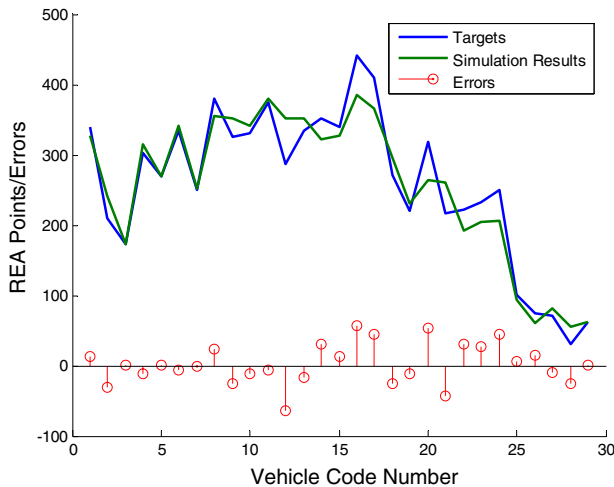


Fig. 11. Prediction results for the total REA points at 6000 km.

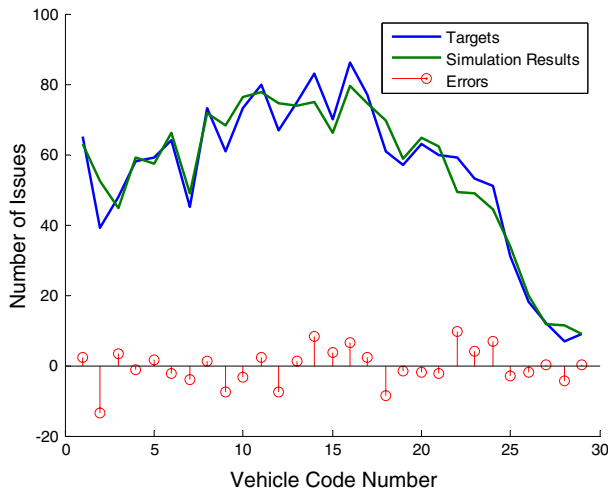


Fig. 12. Prediction results for the number of issues at 6000 km.

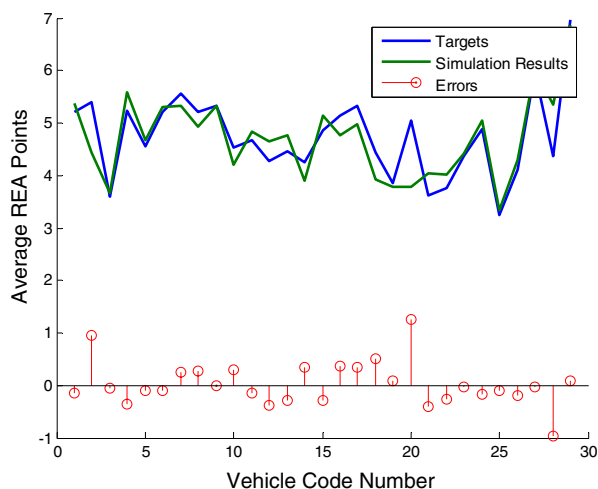


Fig. 13. Prediction results for the average REA points at 6000 km.

## 6. Conclusions

In the work presented here it has been demonstrated how an ANN system can be developed in order to predict accurately enough the reliability behaviour of an automotive vehicle at 6000 km by only knowing information from the initial vehicle's inspection at 0 km. The ANN managed to learn the associations between the input parameters at 0 km and the target values of the reliability performance at 6000 km. The ANN has therefore managed to recognise and identify the degradation mechanism from 0 to 6000 km and has used this knowledge in order to predict the reliability behaviour of the test vehicle models through their development phase. The overall error for the whole network and all the three output parameters was almost 9% which is quite acceptable. The reason for this error value was due to the fact that there was a limited amount of data and paradigms to train the network and improve its ability to generalise well. The immediate effect of this system is that a smaller number of vehicles can be tested at 6000 km. An initial inspection at 0 km can indicate what its reliability behaviour will be at higher mileage. The system can be easily adapted to calculate reliability performance for a number of higher mileages, given that tests to this mileage have been carried out.

The first step after this stage is to enrich the system with other parameters that describe changes during the development stage of the vehicles. These new parameters can be, for example, in the form of improvement factors from one version of the model to another. This will assist in the investigation and prediction of the vehicle's reliability performance when changes and improvements are introduced to it. For example the engineers will be able to visualise what will happen to the vehicle if a certain type of improvement is introduced to it. This way they will be able to apply to the vehicle those improvements that will have the best effect on the reliability performance and also those improvements that will be the cheapest. Resource management could be another application of the system.

## References

- Al-Assaf, Y., & El Kadi, H. (2001). Fatigue life prediction of unidirectional glass fiber/epoxy composite laminate using neural networks. *Composite Structures*, 53, 65–71.
- Al-Thabtabai, H. (1998). A framework for developing an expert analysis and forecasting system for construction projects. *Expert Systems with Applications*, 14(2), 259–273.
- Antony, J., & Banuelas, R. (2002). Key ingredients for the effective implementation for six sigma program. *Measuring Business Excellence*, 6(4), 20–27.
- Briqueu, L., Gottlieb-Zeh, S., Ramadan, M., & Brulhet, J. (2002). Inferring lithology from downhole measurements using an unsupervised self-organizing neural network: study of the Marcoule silty clayish unit. *C.R. Geoscience*, 334, 331–337.
- Catherwood, P. (2002). What's different about six sigma. *Manufacturing Engineer*, 81(8), 186–189.
- Cizelj, R. J., Mavko, B., & Kljenak, I. (2001). Component reliability assesment using quantitative and qualitative data. *Reliability Engineering and System Safety*, 71, 81–95.

- Cohen, W. M., & Levinthal, D. A. (1994). Fortune Favors the Prepared Firm. *Management Science*, 40(2), 227–251.
- Cunningham, P., & Bonzano, A. (1999). Knowledge engineering issues in developing a case-based reasoning application. *Knowledge-Based Systems*, 12(7), 371–379.
- Di Natale, C., Paolesse, R., Macagnamo, A., Troitsky, V. I., Berzina, T. S., & D'Amico, A. (1999). Pattern recognition approach to the study of the interactions between metalloporphyrin Langmuir–Blodgett films and volatile organic compounds. *Analytica Chimica Acta*, 384, 249–259.
- Doyle, M. D., Ang, C. S., Martin, D. C., & Noe, A. (1996). The visible embryo project: embedded program objects for knowledge access, creation and management through the world wide web. *Computerized Medical Imaging and Graphics*, 20(6), 423–431.
- Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). *Neural Network Design*. PWS Publishing Company.
- Hamel, G., & Prahalad, C. K. (1991). Corporate imagination and expeditionary marketing. *Harvard Business Review*, 69(4), 81–92.
- Ishihara, S., Ishihara, K., Nagamachi, M., & Matsubara, Y. (1997). An analysis of Kansei structure on shoes using self-organizing neural networks. *International Journal of Industrial Ergonomics*, 19, 93–104.
- Johnson, A., & Swisher, B. (2003). How Six Sigma Improves R&D. *Research Technology Management*, 46(2), 6–17.
- Kaminskyj, I., & Materka, A. (1995). Automatic source identification of monophonic musical instrument sounds. In IEEE International Conference on Neural Networks.
- Kang, J. Y., & Song, J. H. (1997). Neural network applications in determining the fatigue crack opening load. *International Journal of Fatigue*, 20, 57–69.
- Kasabov, N. K. (1996). *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. MIT Press.
- Kwak, Y. H., & Anbari, F. T. (2004). Benefits obstacles and future of six sigma approach. *Technovation*, 1–8.
- Lacassie, J. P., Roser, B., Ruiz Del Solar, J., & Herve, F. (2004). Discovering geochemical patterns using self-organising neural networks: a new perspective for sedimentary provenance analysis. *Sedimentary Geology*, 165, 175–191.
- Lee, J. A., Almond, D. P., & Harris, B. (1999). The use of neural networks for prediction of fatigue lives of composite materials. *Composites: Part A*, 30, 1159–1169.
- Liao, S.-H. (2003). Knowledge Management Technologies and Applications – Literature Review from 1995 to 2002. *Expert Systems with Applications*, 25, 155–164.
- MathWorks, I. (2004). Neural Network Toolbox User's Guide. MATLAB 7.0, Release 14, MATLAB Manual.
- McClusky, R. (2000). *The Rise, Fall, and Revival of Six Sigma. Measuring Business Excellence*, 4(2), 6–17.
- Mohan, S., & Arumugam, N. (1997). Expert system applications in irrigation management: an overview. *Computers and Electronics in Agriculture*, 17, 263–280.
- Nascimento, C. A. O., Giudici, R., & Guardani, R. (2000). Neural network based approach for optimization of industrial chemical processes. *Computers and Chemical Engineering*, 24, 2303–2314.
- Ozer, M. (1999). A survey of new product evaluation models. *Journal of Product Innovation Management*, 16, 77–94.
- Pham, D. T., & Pham, P. N. T. (1999). Artificial intelligence in engineering. *International Journal of Machine Tools and Manufacture*, 39, 937–949.
- Pleune, T. T., & Chopra, O. K. (2000). Using artificial neural networks to predict the fatigue life of the carbon and low-alloy steels. *Nuclear Engineering and Design*, 197, 1–12.
- Venkatesh, V., & Rack, H. J. (1999). A neural network approach to elevated temperature creep-fatigue life prediction. *International Journal of Fatigue*, 21, 225–234.
- Vogl, T. P., Mangis, J. K., Rigler, A. K., Zink, W. T., & Alkon, D. L. (1988). Accelerating the convergence of the backpropagation method. *Biological Cybernetics*, 59, 257–263.
- Yadav, O. P., Singh, N., Chinnam, R. B., & Goel, P. S. (2003). A fuzzy logic based approach to reliability improvement estimation during product development. *Reliability Engineering and System Safety*, 80, 63–74.
- Yin, C., Rosendahl, L., & Luo, Z. (2003). Methods to improve prediction performance of ANN models. *Simulation Modelling Practice and Theory*, 11, 211–222.
- Zhang, Z., Barkoula, N. M., Karger-Kocsis, J., & Friedrich, K. (2003). Artificial neural network predictions on erosive wear polymers. *Wear*, 255, 708–713.
- Zhang, Z., & Friedrich, K. (2003). Artificial neural networks applied to polymer composites: a review. *Composites Science and Technology*, 63, 2029–2044.