Diverse neural net solutions to a fault diagnosis problem*

A.J.C. Sharkey, N.E. Sharkey and O.C. Gopinath
Department of Computer Science,
University of Sheffield, U.K.

September 30, 1999

Abstract

The development of a neural net system for fault diagnosis in a marine diesel engine is described. Nets were trained to classify combustion quality on the basis of simulated data. Three different types of data were used; Pressure, Temperature and Combined Pressure and Temperature. Subsequent to training, three nets were selected and combined by means of a majority voter to form a system which achieved 100% generalisation to the test set. This performance is attributable to a reliance on the software engineering concept of *diversity*. Following experimental evaluation of methods of creating diverse neural net solutions, it was concluded that the best results should be obtained when data is taken from two different sensors, (eg a Pressure and a Temperature sensor), or, where this is not possible, when new data sets are created by subjecting a set of inputs to non-linear transformations. These conclusions have far reaching implications for other Neural Net applications.

Keywords: Neural Nets, Fault diagnosis, Engines, Diversity, Software Engineering.

1 Introduction

In this paper we present a neural net system solution to a problem of fault diagnosis in a four-stroke marine diesel engine; that of early detection of faulty combustion in an engine cylinder. Recognition of faulty combustion usually requires the intervention of a skilled marine engineer, to undertake the time-consuming and fallible process of comparing current indicator diagrams with

---

[0] We would like to thank the EPSRC/DTI Grant No.GR/H85427 IED4/1/9301 for funding this research.

'ideal' diagrams, a process which relies on occasional checks rather than constant monitoring. The solution we present here is based on simulated data, but shows the feasibility of developing a Neural Net system that could be used to constantly monitor combustion condition during every engine cycle and to provide immediate warnings of faults. The system consists of three nets, combined by means of a simple majority voter, and exhibits 100% generalisation to a set of previously unseen inputs. This impressive level of generalisation is achieved because the three nets fail diversely, and although each of the three nets fails on some of the inputs, there are no inputs which fail on more than one net at any one time.

In the following sections, first the method by which the nets were trained will be described. Following that, we shall discuss the relevance of the software engineering concepts of 'reliability through redundancy', and 'diversity', to neural net research, and provide a demonstration of its relevance by applying it to this particular fault diagnosis problem.

2 Diagnosis of combustion condition

In this section of the paper we shall describe the way in which neural nets were trained to recognise faults in simulated data from a diesel engine; specifically to classify combustion condition as either ideal, or as an example of one of two kinds of faulty combustion. The early detection of combustion condition in a marine engine is crucial since undetected faults can rapidly become compounded, and even result in total breakdown. Marine engineers routinely make use of indicator diagrams to monitor the condition of an engine, comparing current diagrams with ideal indicator diagrams taken during ships trials. However there are a number of disadvantages to this method. The manual process of creating indicator diagrams is time consuming, and even where the process is automated (Murayama, et al, 1979), it still relies on the costly, and fallible, expertise of an experienced engineer. By contrast, a neural net fault detector could be used to monitor combustion condition on every engine cycle, and provide a warning in case of any aberration from the norm.

The data to be used for training was generated using the MERLIN simulator[1] (which produces data almost identical to that from a real engine, Banisoleiman et al 1993). The simulator can generate two types of data during every engine cycle. Both of these measures indicate combustion condition and correspond to (i) the pressure in the engine cylinder and (ii) the temperature of the gas in the cylinder.[2] MERLIN provides a measure of pressure and temperature at fixed

---

[1] Engine simulation software developed by LLoyds Register of Shipping, London, and made available through the assistance of Dr Kian Banisoleiman

[2] The data for training was generated by Gopinath O. Chandroth, and this study relies on his expertise as a Chief ship's engineer. The idea of training neural nets on this fault diagnosis problem forms the basis of his MSc thesis, (Gopinath, 1994).

intervals of 0.5 degrees crank angle rotation, resulting in a total of 1440 values for each parameter. In addition to being used to simulate normal combustion, (*Ideal combustion*), the MERLIN simulator was adjusted to simulate two faults: (i) *Retarded injection of fuel* and (ii) *Advanced injection of fuel*. Normally fuel is injected into the cylinder towards the end of the compression stroke. Retarded fuel injection means that the fuel is injected into the cylinder later than normal. This might be caused by a slipped fuel injection cam or a faulty fuel pump delivery valve. Retarded fuel injection results in 'after-burning', whereby complete combustion of the injected fuel does not occur. The unburned fuel impinges on the turbocharger blades, causing reduced turbocharger efficiency. Reduced turbocharger efficiency implies less air available for combustion, which in turn causes more unburned fuel to leave the cylinders. Accumulated unburned fuel could even ignite in the exhaust uptake and cause a major accident. Advanced fuel injection could also be due to a slipped cam or by a faulty fuel injector. Advanced fuel injection means that the fuel is injected too early, with the result that combustion occurs earlier and high firing pressures develop. The high pressure means that the engine components are under uneven stress, and that more power is developed in the cylinder concerned, resulting in unequal power distribution between the cylinders.

MERLIN was used to produce 588 examples of each of these three classes of data, each example corresponding to one engine cycle. On each engine cycle, measures of pressure and temperature were recorded for every 0.5 degrees of crank rotation. The 1440 values taken on each engine cycle were then reduced, using statistical sampling techniques, such that each pressure example consisted of 52 input values and each temperature example consisted of 71 input values. No explicit information about the position of the crank was included in the input; the temporal characteristics of the data were implicitly represented. The two sets of data, based on either pressure or temperature, were treated seperately: the training regimes for each are described below. As is typical in neural net training regimes, nets were trained on a subset of the available data, and then tested for generalisation on a previously unseen set of data. Thus, both data sets of pressure and temperature were divided into a test set of 414 example pairs (138 from each class) and 9 training sets of 150 example pairs (50 from each class).

2.1 Pressure data

Nets were trained on nine different training sets, from nine different random initial conditions; resulting in a total of 81 trained nets. Each training set consisted of 150 examples of pressure data, each containing 52 values taken during one engine cycle. Each net had 52 inputs, 2 hidden units, and 3 output units, and was trained with a learning rate of 0.6, a momentum of 0.2 and an error tolerance of 0.1. All nets converged, and the average training time was 343 cycles. The nets were trained to produce an output of 100 if the example was one of Ideal Combustion, 010 if it was an example of Retarded Injection, and 001 if it was an example of Advanced Injection. When the trained nets were tested on a test set of 414 previously unseen examples, they produced the correct classification for, on average, 98.04% of the data.

2.2 Temperature data

A second set of 81 nets was produced by training on nine different training sets from the starting point of nine different random seeds. Each training set contained 150 examples of temperature data, each of which consisted of 71 values taken during one engine cycle. Individual nets had 71 inputs, 2 hidden units and 3 output units and were trained with learning rates of 0.6 a momentum term of 0.2, and an error tolerance of 0.1. All nets converged (Mean no. of cycles to train, 576). As in the case of nets trained on pressure data, nets were trained to output 100 for an example of Ideal Combustion, 010 for an example of Retarded Injection and 001 for and example of Retarded Injection. When tested on the test set of 414 previously unseen examples, they produced the correct classification for, on average, 98.15% of the data.

2.3 Combined Pressure and Temperature data

Nine new training sets were produced by combining the pressure and temperature training sets. Each example in the combined training sets consisted of 123 input values (52 pressure and 71 temperature), all taken during the same engine cycle. Nets were trained on these sets, from the starting point of nine different initial conditions, resulting again in a total of 81 trained nets, with architectures of 123 inputs, 2 hidden units and 1 output units. The same learning rate of 0.6 and momentum of 0.2, and error tolerance of 0.1 were used to train the nets; all nets converged and the mean training time was 307 cycles. When tested for generalisation on a combined test set of 414 examples, the mean generalisation percentage was 98.45%.

2.4 Interim conclusions

The results reported here indicate the feasibility of applying neural nets to the monitoring of combustion condition, and as such, compliment the results of other neural net applications to fault diagnosis and condition monitoring (eg Boek, 1991; Duyar & Merrill, 1992; Lihovd & Ramussen, 1993; Macintyre et al 1993). Duyar and Merrill (1992) report a system, based on simulated data, which uses two levels of neural nets for fault diagnosis of the Space shuttle main engine. A classifier level classifies faulty input data as two types of fault, and a second 'severity' level classifies the severity of the fault. Boek (1991) reports experiments in which real faults were introduced into the operation of a desk top fan, and the resulting vibration signals used to train a neural net classifier. Lihovd and Rasmussen (1993) similarly introduce faults into a centrifugal pump, and use resulting performance and vibration data to train a classifier that consists of six neural nets, each trained to recognise a particular type of fault. MacIntyre, Smith, Harris and Brason, (1994) propose a hybrid architecture for condition monitoring in Blyth power station; whereby inputs are preprocessed by means of a self-organizing net before being used to train a multi-layer perceptron.

The current application differs from others in that it makes use of measures of pressure and temperature, rather than the more commonly used vibration signals, and is applied to the new problem of diagnosing combustion faults in a diesel engine. The classification it performs is a simple one; one that can also be performed by a single layer net trained using the Delta rule, but at a cost in terms of an increased number of training cycles and reduced generalisation. Although the training and testing sets were limited, they are comparable in size to those used in the other applications referred to above. Under the current methods, the level of generalisation achieved was above 98% for nets trained on either Pressure data, Temperature data, or a combination of the two. The best results, (98.46% generalisation) were achieved when Pressure and Temperature data was combined. Nevertheless, even though these results are encouraging, what they mean is that for at least 1.54% of the examples the difference between the output, and the correct output was greater than 0.1. This indicates a margin of error, the reduction of which provides a target for the application of software engineering techniques.

3 Reliability through redundancy

The interrelated issues of testing and reliability are extensively discussed in the software engineering literature; but have not yet received much attention in the Neural Net community. In software engineering there is a general recognition of the difficulty of establishing the reliability of a software program. The problem is that a program may appear to be error-free after extensive testing; but has necessarily only been tested under a finite number of circumstances, with the result that there still remain unforeseen situations which may result in errors. Similarly, a program may produce error-free performance in terms of its software specification, but that specification itself may be wrong. In the light of these problems, there are certain software engineering techniques which can be employed to improve the fault-tolerance of programs. One that is frequently employed is that of diversity of design, or N-version programming. Here the traditional approach is to undertake the independent development of alternative versions of a piece of software. These versions can then be executed in parallel, and the outputs compared. Depending on the application, disagreement could lead to fail-safe shutdown, or, where non-stop operation is required, a majority vote could be chosen.

It has been suggested, (Sharkey and Partridge, 1992; Partridge and Sharkey, 1994) that the idea of N-version programming is one that could be usefully applied in the domain of Neural Computing applications. Although sophistication in this area is increasing, in in the past Neural Computing researchers have been content to divide their data into a training set and a test set, and to accept a measure of correct generalisation to the test set as the main indicator of the net's performance. A performance measure of 97% correct generalisation to previously unseen inputs would be considered impressive. However, it is easy to imagine situations in which a generalisation level of 97% would not be adequate; for example in the case of a neural net controlled component of a Nuclear Power reactor. In addition, the level of generalisation attained depends entirely on the contents of the test set; but our concern in this paper is not with the development of an improved test methodology for neural nets. The idea that is explored here is the application of the technique of N-version programming to Neural nets.

Before considering its application to Neural Computing, the concept of N-version programming will be further explained. Conventional N-version programming is made use of in real systems. For example, the Airbus Industry A310 aircraft makes use of dual programming in the slat and flap control system (Martin, 1983). Similarly, Taylor (1981) describes the application of dual programming to point switching, signal control and traffic control in the Gothenburg area by Swedish State Railways; as does Hagelin (1987). More recently, however, the emphasis has been on diversity of design, rather than mere redundancy.

Although it has been assumed that N-version programming techniques provide increased software reliability (eg Avizienis and Kelly, 1984), there is evidence (Knight and Leveson, 1986), that even when working independently, people tend to make the same mistakes when solving a difficult intellectual problem. What this means is that even when programs are written by different people, when they do fail they do not fail independently; thus if one program fails on a particular input the probability of other programs also failing on that input are increased. Eckhardt and Lee (1985) developed this idea, provided a formal definition of independence of versions, and presented results which imply that, even when true independence of the versions is achieved, they will still exhibit dependent behaviour. On the basis of these findings, Littlewood and Miller (1989) argue that the important idea is that of *diversity* rather than independence.

Littlewood and Miller (1989) introduce the idea of promoting diversity through the employment of different methodologies, and argue that it is possible to do better than achieving independence of failures. That is, to achieve a situation in which there is a low, or even *negative* correlation between the failures of the two methodologies, such that if a particular input fails on several programs from one methodology, that input succeeds in programs from the other methodology. They also argue that greater diversity of design results in a greater chance of freedom from coincident failures. Their argument is reinforced by the empirical results presented by Adams and Taha (1992). Littlewood and Miller (1989) define the degree of methodological diversity in terms of the size and the sign of the (product moment) correlation between probability of failures for two contrasting methodologies. They make the point that reliability can be increased not only by employing contrasting methodologies, but also by creating several versions within a methodology. To this end, they provide a statistical method for calculating the correlation between failures of several versions of one methodology and several versions of a second methodology.

In the experiments which follow, we make use of Littlewood and Miller's statistical model, creating different Neural Net methodologies through the use of 'Extensional programming' techniques. The notion of *extensional programming* (raised by Cottrell, Munro and Zipser, 1989, and expanded on by Sharkey and Sharkey, in press), is that the performance of nets is determined by a number of factors which are in the control of the researcher. These include the following; (i) the architecture of the net, or nets, and their initial structure; (ii) the learning technique; (iii) the learning parameters (eg. the learning rate); (iv) the input and output representations and (v) the content of the training sample. The factors manipulated in the current experiment were the random initial conditions, the content of the training sample, and the type of input representation. The relevance of these factors to the creation of diversity was investigated by selecting different sets of nets from the entire set trained to classify combustion condition (as described in the preceding section). The factors varied here differ from the manipulations previously employed in the standard symbolic paradigm to decrease the number of coincident failures, which have included the following; (a) working from different specifications (Ramamoorthy et al, 1981); (b) using different programmers (Knight and Leveson, 1986); and (c) using different types of programming language (procedural versus logic programming), (Adams and Taha, 1992).

In our investigations of diversity in Neural Nets, our aim is to create nets that are diverse in the sense that their failures are negatively correlated. However, mere negative correlation is not sufficient; a particular kind of negative correlation is required, for we also need the nets to be reasonably accurate, and to cover the function. This point can be made clearer if we look at an example of contrasting methodologies which *are* negatively correlated, but which do *not* provide good coverage of the function. In Figure 1, the probability of failing (number of inputs to fail on that number of versions) in one methodology is plotted against the probability of failing in a different methodology. The results on which this plot is based were taken from a different study, not reported here, in which nets were trained on a simple function. The nature of the function is irrelevant for our present purposes; the illustration is included because it provides an example of a negative correlation accompanied by a high incidence of coincident failures. The two methodologies plotted in this study correspond to one set of nets trained on the function (Methodology 1), and a second set not trained on any function (Methodology 2). Although the failures in the two methodologies were negatively correlated (ρ = -0.6381), examination of Figure 1 should make it clear that the combined performance of the two nets would be poor, due to the lack of combined successes, and the presence of inputs which fail on all versions of both methodologies.

FIGURE 1 ABOUT HERE

By contrast, Figure 2 shows an idealised result, where two methodologies are negatively correlated, but the majority of inputs are correct on both methodologies. In this example, where failures occur in one methodology, they have a zero chance of failing on the other methodology. Thus, if the two methodologies were combined, there is an increased chance of a correct response being produced by one of the two methodologies.

FIGURE 2 ABOUT HERE

4 Diversity Experiments

We shall employ two methods of illustrating the results we obtained; (i) tables of the correlations between failures in contrasting methodologies and (ii) 3-dimensional plots of the probability of failing on a pair of methodologies. In both cases, the failures are the failures of trained nets tested on the previously unseen test set (for which the average generalisation for pressure was 98.04%, and for temperature was 98.15%.). As mentioned earlier, four different methods for creating diverse solutions were explored; these being the manipulation of (i) initial conditions; (ii) training set; (iii) contrasting measures and (iv) combining measures.

Manipulation of Initial conditions:

The effect of variations in initial conditions on diversity was investigated, using the Littlewood and Miller (1989) statistical model, by treating each initial condition as a methodology, and each training set trained with that initial condition as a version within that methodology. By 'initial condition' we mean the set of random weights used to initialise the net. Using the statistical methods of Littlewood and Miller (1989), it was then possible to calculate the correlation between the failures of pairs of methodologies, testing across several versions. In Table 1, it is possible to see the correlations between nets trained in different methodologies based on different initial conditions. All of these nets were trained on Pressure data, (the same comparisons were also undertaken for Temperature data. The results followed a similar pattern, but for the sake of brevity are not displayed here). Each entry in the table corresponds to the correlation between two methodologies (two different random seeds), where each methodology consists of 9 different versions, or training sets. As is apparent from the table, there is little evidence here of any resulting diversity. The failures between the methodologies are all highly correlated. Figure 3 shows a plot of one of these comparisons (between Random Initial Condition 1, and Random Initial Condition 2). As should be apparent from the figure, as the probability of failing on one methodology increases, so does the probability of failing on the other.

TABLE 1 ABOUT HERE

FIGURE 3 ABOUT HERE

Manipulation of training sets:

The effect of variations in training sets was investigated by treating each training set as a methodology, and each net trained on that set from the starting point of a different set of random weights as a version within that methodology. In Table 2, the correlations between nets trained on different training sets are shown. Each entry in the table corresponds to the correlation between nine versions of one training set methodology, and nine versions of a second training set methodology. All of the entries in Table 2 are based on Pressure data only (results based on Temperature data showed a similar pattern, but for brevity are not displayed). Examination of the table shows that the correlations here are more variable, and mostly lower than was the case where the Initial Conditions were manipulated. Nonetheless, the level of diversity achieved is still not ideal, as can be seen by the illustration provided in Figure 4. Here it is evident that even in a case where the correlation was relatively low, there is still one input which fails on all versions in both methodologies, and other inputs which fail on several versions in both methodologies.

TABLE 2 ABOUT HERE

FIGURE 4 ABOUT HERE

Contrasting Measures:

Our third manipulation was to look at the extent to which the failures of nets trained to classify faults based on pressure data, would correlate with the failures of nets trained to classify faults based on temperature data. Here corresponding training sets, based on either Pressure or Temperature data, were treated as methodologies, with different random seed versions being treated as versions within that methodology. Correlation between the failures of nets trained on these two kinds of data is computed with reference to their performance on the generalisation set. Two corresponding versions of the generalisation set are used; one for Pressure, and one for Temperature. Each input vector in the Pressure generalisation set has a corresponding vector in the Temperature generalisation set that is based on the *same* engine cycle. It is important to emphasise this correspondence, for it is this that makes it meaningful to look at the correlations between failures. Thus, in both cases, the inputs were based on the same engine cycle; an engine cycle which could be classified as either an example of ideal combustion, or of advanced fuel injection, or of retarded fuel injection. However, when contrasting measures are employed, the actual inputs on which this classification is based differ, since they are either based on the Pressure in the cylinder, or the Temperature in the cylinder. Each entry in Table 3 corresponds to a comparison between a pair of methodologies; the methodology being determined by the training set, and the data which it contains. Each cell in the table represents the correlation between a Pressure and Temperature training set, where for each methodology nine versions were created by training from different initial conditions. Examination of Table 3 shows that the correlations obtained were much lower than when either initial conditions or the training sets were manipulated. The lowest correlation achieved as a result of varying the initial conditions (Table 1) was 0.9273, and as a result of varying the training sets (Table 2) was 0.1732. In Table 3, the highest correlation is 0.6704, but the lowest is -0.0347; evidence that the negative correlations discussed by Littlewood and Miller (1989) can occur in practice. Figure 5 provides an illustration of the correlation between the first Pressure training set, and the third Temperature training set. Examination of this figure shows that the distribution of the failures is close to the ideal, since there are no inputs which fail on all versions in both methodologies, and as the probability of failing on one methodology increases, the probability of failing on the other remains close to zero.

FIGURE 5 ABOUT HERE

TABLE 3 ABOUT HERE

Combining measures:

Our final manipulation involved looking at the correlations between nets trained on a combination of Pressure and Temperature, and nets trained on either Pressure or Temperature data alone. In Tables 4 and 5, correlations are shown between Combined Temperature and Pressure nets and either Pressure, (Table 4) or Temperature (Table 5) nets alone. The data type was treated as the Methodology, and the use of different initial conditions was used to create nine versions for each cell of the two tables. As can be seen in the tables, this manipulation, like that of using contrasting methodologies, also provides evidence of both low and negative correlation of the failures.

TABLE 4 ABOUT HERE

TABLE 5 ABOUT HERE

In summary, the results reported here indicate that of the four methods investigated, it is the use of contrasting measures and of combining measures which provide the greatest diversity of failures. Manipulation of the initial conditions produced nets where failures were correlated, (confirming the detailed investigations of initial conditions reported by Sharkey, Neary and Sharkey, 1995). Lower correlations were obtained when training sets were manipulated; an improvement noted elsewhere, (Sharkey and Partridge, 1992). However the best results were obtained when contrasting, or combined measures were employed. Under these circumstances, several of the resulting comparisons correspond to the ideal situation, in which failures in one methodology are backed up by successes in the other methodology. In addition, our results provide evidence of negative correlations; negative correlations which can also be shown to cover the function, and which are not just indications of poorly performing nets. These negative correlations provide an illustration of what Littlewood and Miller (1989) referred to as the possibility of doing better than 'independence of performance'.

5 Neural net system solution

When neural nets were trained using three different types of training data, the resulting correlations between their generalisation failures were often low, or even negative. It was therefore possible to identify several sets of three nets, each trained on a different type of data, which are based on negatively correlated methodologies. Thus, Temperature training set 9 is negatively correlated with Pressure training set 4, and also with Combined Temperature and Pressure set 3. At the same time, Pressure training set 4 is also negatively correlated with Combined Temperature and Pressure set 3. Guided by such correlations, it was possible to choose three nets, one from each of these sets, which never fail on the same inputs. Of the three selected nets, each fails on some of the test set (5 errors in the Temperature net, 4 errors in the Pressure net, and 10 errors in the Combined Pressure and Temperature net), but for any one input in the test set there is only ever one of these nets which fail; the other two make the correct classification. Therefore, when these three nets are combined by means of a majority voter, they produce 100% generalisation performance.

The final neural net system solution, therefore, consists of three neural nets, combined by means of a majority voter. Each net was trained on the same input data, but in each case, the data has been preprocessed in a different way, with the result that different patterns of generalisation are obtained. For one net, combined Pressure and Temperature data is used, and for the other two nets, Pressure data, and Temperature data, respectively, are used. The methodology for choosing three nets such net needs further development, but relies on the computation of pairwise correlations between nets, and the selection of negatively correlated nets with no shared failures.

6 Discussion and Conclusions

The case study described in this paper provides a clear indication of the feasibility of using neural nets for the purposes of fault diagnosis of a marine engine. The particular engine in question is a typical prime mover for a power generator on land, or at sea on a ship. The point is that the method is applicable to any diesel engine, and could even be applied to petrol engines (implications for the automobile industry). Nets were trained, using the backpropogation algorithm, on the basis of three types of input data (pressure, temperature and combined pressure and temperature), to classify engine cycles as examples of (i) Ideal combustion, (ii) Advanced fuel injection or (iii) Retarded fuel injection. Following this training, a set of three nets, one based on each type of data, was selected which when combined by means of a majority voter, formed a system which achieved 100% generalisation performance on the test set. The system is simple, but effective and there would be obvious advantages to its use to monitor combustion condition without the need for the continual presence of a skilled ship's engineer.

As referred to earlier, there are other examples of applications of neural nets to fault diagnosis (eg Boek, 1991; Duyar & Merrill, 1992; Lihovd & Ramussen, 1993; Macintyre, J. et al 1993). The most interesting novel aspect of the present study is its reliance on the concept of diversity; a reliance which has interesting implications for neural net applications in general. First, the implication is that the reliability of a neural net solution to a problem can be increased through creating a system that incorporates several different nets which fail diversely, and are combined by means of a voter. The question then arises as to the best way to create such diverse solutions. Our results indicate that the best way to create diversity is to train nets on different types of data. Negative correlations were obtained when nets trained on Pressure data were compared to nets trained on Temperature data; and when nets trained on either of these were compared to those trained on a combination of Pressure and Temperature data. On the basis of this, it could be concluded that, where possible, this effect should be recreated by training nets on data from two different sensor readings. This would seem to be equivalent to training nets on either a Pressure sensor, or a Temperature sensor, or on a combination of the two inputs.

In practice, it turns out that, as yet, sensors for reading the temperature of the gases in the cylinders of ship's engines do not exist. What this means is that the temperature measure produced by the MERLIN simulator is, in effect, a theoretical construct, and is created by means of the application of thermodynamic equations to the Pressure data. This does mean that, in the absence of a real temperature sensor, if the Pressure sensor failed, no data would be obtained. However, the interesting aspect of this is that it implies an important alternative method of creating diverse solutions, in those situations in which a second sensor is not available. That is, the idea of taking a set of data and creating a further set through the application of non-linear transformations of the original input set; equivalent to the application of thermodynamic equations to the pressure data. On the basis of the results reported here, this should have the effect of promoting diversity. Further research is currently being undertaken to explore the effect on diversity of applying non-linear transformations of the inputs.

In conclusion, the results we report here are interesting in two respects. First, for the demonstration they provide of the potential of a neural net solution for fault diagnosis in a ship's engine. And second, for their wider implications to neural net applications in general, and for any situations in which increased accuracy is required. Clearly, the software engineering concept of diversity is one that can be usefully employed in Neural Computing. On the basis of our investigation into methods for creating diverse solutions, we conclude that the best methods for doing so are those which rely on changes in the input-output relationships in the training set; either as a result of employing different sensors, or through the use of non-linear transformations to preprocess the data.

7 References

Avizienis, A. & Kelly, J.P.J. Fault diagnosis by design diversity: Concepts and experiments. IEEE Comput. 17, 8, 67-80.

Adams, J.M. & Taha, A. (1992) "An Experiment in Software Redundancy with Diverse Methodologies," *Proc of the Twenty-Fifth Hawaii International Conference on Systems Sciences.*

Banisoleiman, K., Smith, L.A., & Matheieson, N. (1993) Simulation of diesel engine performance, *Transactions of the Institute of Marine Engineers*, 105 (3) 117-135.

Boek, M.J. (1991), 'Experiments in the application of neural networks to rotating machine fault diagnosis. Paper presented at IEEE International Joint Conference on Neural Networks, Singapore 1, 769-74.

Cottrell, G.W., Munro, P. & Zipser, D. (1989) "Image compression by back propagation: An example of extensional programming", In (Ed.) Noel E. Sharkey, *Models of Cognition: a review of Cognitive Science*, Ablex, New Jersey, 208-241.

Duyar, A. & Merrill, W. (1992) "Fault diagnosis for the space shuttle main engine." *Journal of Guidance, Control and Dynamics*, 15 (2) 384-9.

Eckhardt, D.E. & Lee, L.D. (1985) "A theoretical basis for the analysis of multiversion software subject to coincident errors". *IEEE Trans. Software Eng.*, vol SE-11, no 12, pp 1511-1517.

Gopinath, O.C. (1994) "A neural net solution for diesel engine fault diagnosis", MSc thesis, University of Sheffield.

Hagelin, G. (1987) "Safety systems for railways", In (Ed) B. Littlewood, *Software reliability: Achievement and assessment*, Blackwell Scientific Publications, Oxford, pp 213-218.

Knight, J.C. & Leveson, N.G. (1986) An experimental evaluation of independence in multiversion programming. *Trans on Software Eng*, Vol SE-12 no 1.

Littlewood, B. & Miller, D.R. (1989) Conceptual modeling of coincident failures in multiversion software. *IEEE Trans. on Software Engineering*, 15,(12).

Lihovd, E. & Ramussen, M. (1993) "Neural networks in condition monitoring and diagnosis on rotating machinery", Paper presented at the 10th Ship Control Systems Symposium, Ottawa, October, 1993.

MacIntyre, J., Smith, P., Harris, T. & Brason, A. (1993) "Application of neural networks to the analysis and interpretation of off-line condition monitoring data", Paper presented at the Sixth International Symposium on Artificial Intelligence, Monterrey (Mexico), 1993.

Martin, D.J. (1983) Dissimilar software in high integrity applications in flight controls," In *Software for Avionics* (AGARD Conf. Proc. 330), Jan, 1983, pp36-1-36-9.

Murayama, T., Miyamoto, N. & Gotoh.S. (1979) 'Estimation of an indicator diagram-method by the stresses measurement of cylinder head bolts', *Bulletin of the Marine Engineering Society in Japan*, 7 (2), 81-90.

Partridge, D. & Sharkey, N.E. (1994) Neural computing for software reliability. *Expert Systems*, 11, 3, 167-175.

Ramamoorthy, C.V., Mok, Y.R., Bastani, E.B., Chin, G.H. & Suzuki, K. (1981) "Application of a methodology for the development and validation of reliable process control software" *IEEE Trans. Software Eng.*, vol SE-7, pp 537-555.

Sharkey, N.E., Neary, J. & Sharkey, A.J.C. (1995) Searching weight space for backpropogation solution types. In Proceedings of The Second Swedish National Conference on Connectionism.

Sharkey, N.E. & Partridge, D.P. (1992) The statistical independence of network generalisation: an application in software engineering. In P.G. Lisboa & M.J. Taylor (Eds) *Neural Networks: Techniques and Applications* Chichester, UK: Ellis Horwood.

Sharkey, A.J.C. & Sharkey, N.E. (in press) Cognitive Modelling: Psychology and Connectionism. In (Ed.) M.A. Arbib*The Handbook of Brain Theory and Neural Networks*, Bradford Books/MIT Press.

Taylor, J.R. (1981) "Letters from the editor", *ACM Software Eng Notes*, vol 6, no1, pp 1-2.

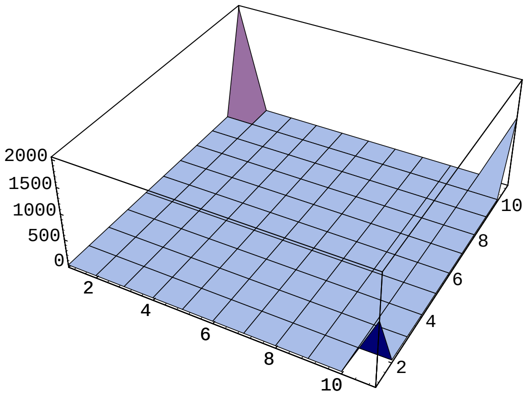

Figure 1: Three dimensional plot of the probability of failure of one methodology against another. The **x** axis shows the number of versions on which failures occurred in Methodology 1, and the **y** axis shows the number of versions on which failures occurred in Methodology 2. The height of the plot represents the number of failures. This plot represents a high negative correlation (the result of plotting trained nets against untrained nets), but there are still a large number of inputs which fail on 10/10 versions in both methodologies, and no inputs which are correct on both methodologies.

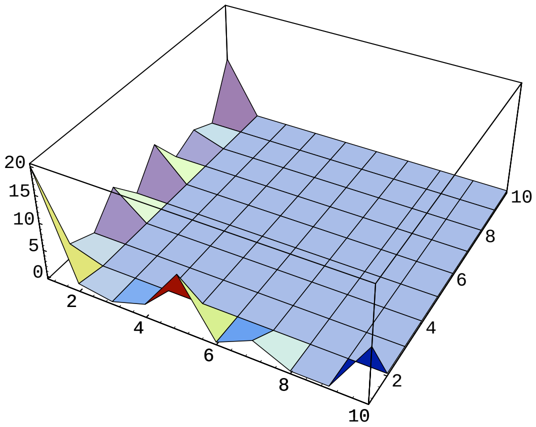

Figure 2: Constructed example of an idealised pattern of negative correlation. The failures occur along the axes. That is, as the probability of failing on one methodology increases, the probability of failing on the second methodology remains at zero. In this illustration, there are 20 inputs which do not fail in either methodology (0 on both the x and y axes), and there are no inputs which fail on both methodologies.

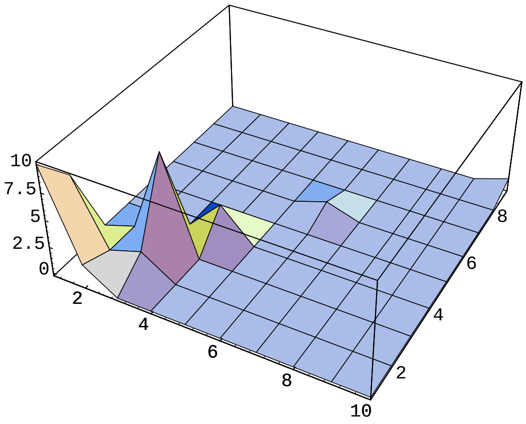

Figure 3: Number of inputs failing on two methodologies, (methodologies corresponding to nets trained from two differents sets of Random Initial Conditions, here RIC 1 and RIC 2). The x and y axes show the 10 versions of each methodology, and the height of the plot shows the number of inputs failing on that number of versions in each methodology. Thus, the plot shows that there are two inputs which fail on six versions in both methodologies. To increase the visibility of the results, the number of shared successes (zero failure on both methodologies) was truncated at 10 from 384. It can be seen that as probability of failing on one methodology increases so does the probability of failing on the other methodology.

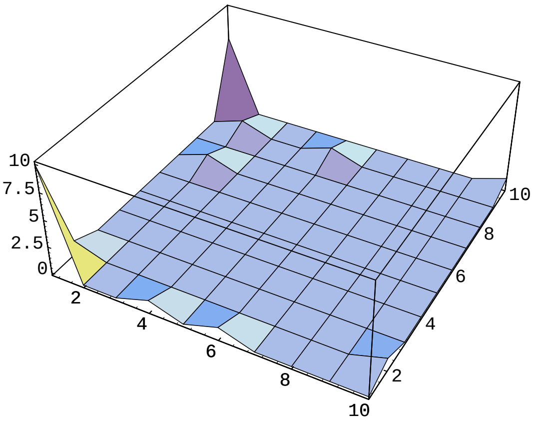

Figure 4: Number of inputs failing on two methodologies, based on two different training sets (T 1 vs T 2). The x and y axes show the 10 versions of each methodology, and the height of the plot shows the number of inputs failing on that number of versions within each methodology. To aid visualisation, the number of shared successes was truncated at 10, from 399. The correlation between these methodologies was 0.2676, but there is one output which fails on all versions in both methodologies, and some inputs which fail on several versions in both methodologies.

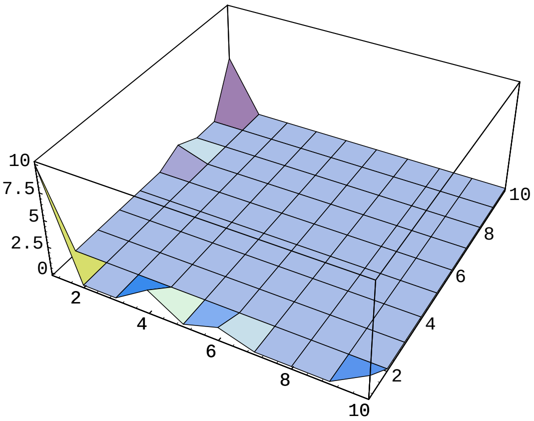

Figure 5: Number of failures per version for Pressure methodology (P 1) plotted against number of failures for Temperature methodology, (T 3). Actual number of shared successes was 403/414, but that number is truncated to 10 for this diagram. The two methodologies are negatively correlated (ρ = -0.012). As can be seen, as the probability of failing on one methodology increases, the probability of failing on the other methodology remains at zero.

	RIC 2	RIC 3	RIC 4	RIC 5	RIC 6	RIC 7	RIC 8	RIC 9
RIC 1	0.971	0.948	0.974	0.943	0.959	0.963	0.963	0.945
RIC 2		0.971	0.969	0.963	0.979	0.971	0.972	0.951
RIC 3			0.969	0.951	0.959	0.972	0.975	0.974
RIC 4				0.943	0.963	0.983	0.985	0.976
RIC 5					0.972	0.950	0.936	0.927
RIC 6						0.978	0.960	0.954
RIC 7							0.990	0.977
RIC 8								0.975

Table 1: Correlations between Random Initial Conditions methodologies. Each cell corresponds to the correlation of failures between two methodologies, created by training from different sets of Random Initial Conditions.

	Tset2	Tset3	Tset4	Tset5	Tset6	Tset7	Tset8	Tset9
Tset1	0.268	0.319	0.675	0.536	0.368	0.467	0.446	0.760
Tset2		0.832	0.477	0.684	0.173	0.705	0.727	0.499
Tset3			0.567	0.563	0.196	0.902	0.486	0.608
Tset4				0.629	0.262	0.574	0.594	0.871
Tset5					0.626	0.551	0.733	0.703
Tset6						0.250	0.453	0.314
Tset7							0.465	0.650
Tset8								0.649

Table 2: Correlations between training set methodologies. Each entry corresponds to correlation between two methodologies, each created by training on a particular training set.

	T 1	T 2	T 3	T 4	T 5	T 6	T 7	T 8	T 9
P 1	0.079	0.0155	-0.012	0.101	0.145	0.098	0.017	0.034	0.185
P 2	-0.032	0.360	0.392	0.133	0.218	0.129	0.294	0.039	0.049
P 3	-0.028	0.498	0.571	0.125	0.211	0.120	0.412	0.038	0.067
P 4	0.022	0.009	-0.020	0.108	0.157	0.104	0.008	0.027	0.088
P 5	-0.035	0.071	0.040	0.159	0.232	0.154	0.064	0.049	-0.009
P 6	-0.032	0.020	-0.020	0.153	0.220	0.586	0.023	0.286	-0.017
P 7	-0.030	0.593	0.670	0.166	0.264	0.161	0.494	0.056	0.056
P 8	-0.034	0.018	-0.021	0.148	0.214	0.472	0.020	0.223	-0.018
P 9	0.013	0.009	-0.013	0.085	0.123	0.082	0.010	0.024	0.056

Table 3: Correlations between Temperature and Pressure methodologies.

	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9
T&P 1	0.068	-0.030	-0.026	0.017	0.079	0.143	-0.028	-0.032	0.010
T&P 2	-0.014	0.352	0.512	-0.022	0.031	-0.023	0.604	-0.024	-0.015
T&P 3	-0.015	0.388	0.489	-0.024	0.067	-0.025	0.578	0.026	-0.017
T&P 4	-0.013	0.037	-0.019	-0.021	0.020	-0.022	-0.021	0.035	-0.015
T&P 5	-0.010	0.641	0.225	0.025	0.410	-0.017	0.137	0.515	0.033
T&P 6	0.322	0.244	0.262	0.255	0.308	0.683	0.284	0.576	0.275
T&P 7	-0.015	0.335	0.490	-0.025	0.026	-0.025	0.578	-0.027	-0.017
T&P 8	0.038	0.148	0.045	0.037	0.131	0.081	0.063	0.210	0.042
T&P 9	0.040	0.011	0.025	0.000	-0.029	-0.033	0.016	-0.035	-0.001

Table 4: Correlations between Pressure and Combined Temperature and Pressure methodologies.

	T 1	T 2	T 3	T 4	T 5	T 6	T 7	T 8	T 9
T&P 1	0.485	0.028	-0.022	0.054	0.034	-0.018	0.048	0.013	0.698
T&P 2	0.205	0.672	0.028	-0.022	0.054	0.034	-0.018	0.048	0.013
T&P 3	-0.029	0.658	0.899	-0.015	0.019	-0.017	0.682	-0.030	0.073
T&P 4	-0.025	-0.020	-0.016	-0.013	-0.012	-0.015	-0.026	-0.027	-0.014
T&P 5	-0.019	0.114	0.145	-0.010	0.022	-0.012	0.087	-0.020	0.075
T&P 6	-0.021	0.066	-0.013	0.268	0.351	0.908	0.065	0.455	-0.011
T&P 7	0.065	0.658	0.900	-0.015	0.019	-0.017	0.682	-0.031	0.255
T&P 8	-0.020	0.020	-0.013	0.111	0.149	0.147	0.017	0.057	-0.011
T&P 9	0.300	0.019	0.035	-0.019	-0.006	-0.022	0.022	-0.040	0.515

Table 5: Correlations between Temperature and Combined Temperature and Pressure methodologies.