
Article:
Zhang, G., Mahfouf, M., Abdulkareem, M. et al. (6 more authors) (2016) Hybrid-modelling of compact tension energy in high strength pipeline steel using a Gaussian Mixture Model based error compensation. Applied Soft Computing, 48. pp. 1-12. ISSN 1568-4946

https://doi.org/10.1016/j.asoc.2016.06.007

Hybrid-Modelling of Compact Tension Energy in High Strength Pipeline Steel Using a Gaussian Mixture Model Based Error Compensation

Guangrui Zhang[a], Mahdi Mahfouf[b,*], Musa Abdulkareem[c], Sid-Ahmed Gaffour[b], Yong-Yao Yang[b], Olusayo Obajemu[b], John Yates[d], Sabino Ayvar Soberanis[e], Christophe Pinna[f]

[a] *College of Information Science and Engineering, Ocean University of China, 23 Xianggang Road East, Qingdao, China, 266100*
[b] *Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield S1 3JD, UK*
[c] *Department of Engineering, University of Leicester, University Road, Leicester LE1 7RH, U.K.*
[d] *docjry@gmail.com*
[e] *Advanced Manufacturing Research Centre with Boeing, The University of Sheffield, Advanced Manufacturing Park, Wallis Way, Catcliffe, Rotherham S60 5TZ, UK*
[f] *The University of Sheffield, Department of Mechanical Engineering, Mappin Street, Sheffield S1 3JD, UK*

Abstract

In material science studies, it is often desired to know in advance the fracture toughness of a material which is related to the released energy during its compact tension (CT) test to prevent catastrophic failure. In this paper, two frameworks are proposed for automatic model elicitation from experimental data to predict the fracture energy released during the CT test of *X100* pipeline steel. The two models including an adaptive rule-based fuzzy modelling approach and a double-loop based neural network model, relate the load, crack mouth opening displacement ($CMOD$) and crack length to the released energies during this test. The relationship between how fracture is propagated and the fracture energy is further investigated in greater detail. To improve the performances of the models, a Gaussian Mixture Model (GMM)-based error compensation strategy which enables one monitor the error distributions of the predicted result

*Corresponding author. Tel.: (+44)(0)114 222 5607
Email address: m.mahfouf@sheffield.ac.uk (Mahdi Mahfouf)

is integrated in the model validation stage. This can help isolate the error distribution pattern and to establish the correlations with the predictions from the deterministic models. This is the first time a data-driven approach has been used in this fashion on an application that has conventionally been handled using finite element methods or physical models.

Keywords: Pipeline, steel, Gaussian Mixture Model, fuzzy, Neural Networks, prediction.

---

1. Introduction

High strength steel is one of the most commonly used materials in engineering works and the modelling, prediction and prevention of failure of steel materials is a key issue in engineering because of safety concerns and to prevent the huge costs incurred during failures. It is thus no surprise that there is a plethora of materials science studies aim at developing new methods of analysis as well as improving existing techniques.

Fracture toughness relates to the ability of a material with intrinsic cracks to resist failure.

Existing analysis on the fracture toughness of steel used in the design of pipeline steel is the calibrated empirical method based on finite element analysis. This method, although returning good modelling results on the test set, have unfortunately been found to have poor generalisation results across steel specimens. As illustrated in [1], using the charpy upper shelf energy which is predicted by the old application ultimately leads to a large error in determining the pipeline fracture resistance.

Physical based-modelling combined with the Finite Element Methods (FEM) are popular for ascertaining fracture characteristics in metals. For example,[2] used the Gurson-Tvergaard-Needleman (GTN) model for the prediction of the ductile failure of 22NiMoCr37 and SA-333 Gr-6 Carbon steel. Also, Karabin et al. in [3], developed a constitutive model based on the Gurson-Tvergaard (GT) and Leblond-Perrin-Devaux (LPD) model [4] for 7085-T7X akluminium alloy

plate samples.

Unfortunately as found in [5], the very high dimensionality and complexities of the process variables may incur high computational cost when trying to analyse the models from first principles.

As illustrated in [6] and [7], mathematical models which are based on data-driven approaches may prove a better solution to this problem. These modelling approaches include fuzzy systems, artificial neural networks, Gaussian processes and support vector machines among others. These approaches have proved to be popular in materials engineering because of their interpolating and generalising capabilities.

For example, [8] predicted the impact energy of API X65 micro alloyed steel using the Artificial Neural Networks (ANN) The fuzzy modelling approach was used for modelling the hysteretic behaviour of CuAlBe wire from experimental data in [9]. The literature is replete with different types of computational intelligence techniques applied to materials modelling. They have shown to provide good accuracy on the specific experimental data. However, these methods tend to be 'biased' and are not able to provide a high degree of confidence in predictions. In this work, we provide a data-driven approach of modelling and consequently predicting materials failure in high strength X100[1] pipeline steel. The research examines two types of modelling framework on the steel crack propagation process during the compact tension test on the steel prototypes. The first is based on fuzzy modelling with hierarchical clustering for initial structure determination and the gradient descent optimisation to improve on the accuracy of the model. This method follows directly from that developed in [7]. The second framework is based on a double loop neural networks. The accuracies in predictions of both methods are compared. To further improve on the accuracy of the two elicited models, an error compensation scheme based on Gaussian Mixture models was developed for the two techniques. The care-

[1] X100 are high grade steel with yield strength greater than $690MPa$ and are usually used for high distance engineering projects.

Element	C	Si	Mn	P	S	Cu
Wt %	0.06	0.18	1.84	0.008	0.001	0.31
Element	Ni	Cr	Mo	Nb	Ti	Al
Wt %	0.5	0.03	0.25	0.05	0.018	0.036

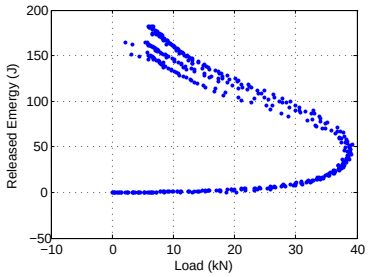
Table 1: Composition of the steel specimens used in the CT experiment.

ful design of this error compensation scheme is not only shown to improve on the performances of the two modelling paradigm but also provides a confidence band in the predictions of each model systematically. Finally, the modelling performance of the proposed modelling framework is compared with 55 that of the adaptive neuro-fuzzy inference system modelling framework (*AN-FIS*). The remainder of the paper is organised as follows: Section 2 analyses the X100 steel data used in the paper explaining the input variables the composition of the steel prototypes. Section 3 briefly describes the proposed fuzzy modelling approach. Section 4 discusses the Neural Network approach used in 60 the paper before the error compensation scheme is used on both models which is described in section 5. Section 6 concludes the paper and recommends direction for future research.

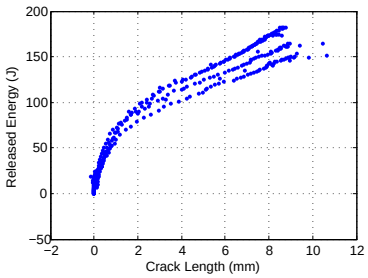
2. Data and Analysis

The experimental data used in this research originated from the works carried out in the Department of Mechanical Engineering, the University of Sheffield [10]. At room temperature, tests were carried-out on six compact tension specimens with longitudinal direction initial crack. This is the direction of shear fracture in cases of real burst pipelines. The steel specimens were side-grooved on each side by up to 20% of the original thickness of the specimen. This ensures a straight crack front and that shear lip formation are reduced. A low displacement control rate of $0.01mm/s$ was used during the tests. Table 1 shows the composition of the *X100* pipeline steel used in the experiments.

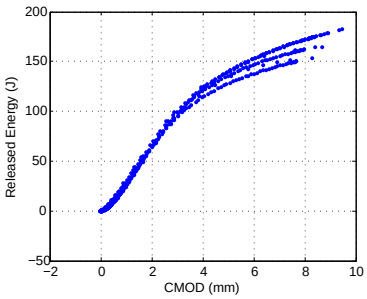
In the experiments the explanatory variables are the load, *CMOD* and crack-



(a) Load v.s. Released Energy



(b) Crack Length v.s. Released Energy



(c) *CMOD* v.s. Released Energy

Figure 1: Distribution of data used in the study. The plots show that the relationship between the input variables against the output variable is strongly non-linear.

length. The output variable is the released flat fracture energy during the tests, which is indicative of the strength of the steel. Six test data sets contain a total of 432 data points which were used in developing the models. Of the 432 data points, 70% was used in the training the two models (fuzzy and neural networks) and the remaining 30% for testing the generalization capabilities of the elicited models. Fig. 1 shows the distributional characteristics of the data.

It is worth noting that the figure shows that the same load value corresponds to two different released energies. This is because the experiment was carried out using a crack speed controlling procedure, meaning that when the elastic property of the metal was broken in the middle of the crack propagation, the load was lowered to maintain the crack speed. Additionally, the figure only shows the released energy as a function of only the load variable. The released energy have been influenced by other input variables..

Variables	Load	CMOD	Crack Length	Released Energy
Load %	1	-4.865E-1	-5.721E-1	-4.015E-1
CMOD	-4.865E-1	1	9.785E-1	9.725E-1
Crack Length %	-5.721E-1	9.785E-1	1	9.552E-1
Released Energy %	-4.015E-1	9.725E-1	9.552E-1	1

Table 2: Correlation Coefficient between the variables (input and output).

2.1. Correlation Coefficient Analysis

Table 2 shows the correlation coefficient analysis for the variables (input and output) to identify the effects the inputs have on the outputs.

The corresponding analysis shows that the correlation between the load and the energy is negative. This is due to decreasing load in the middle of fracture which is caused by the crack controlling procedure. The correlation between the crack length and *CMOD* is high which agrees with the intuition of crack length and *CMOD* increasing simultaneously during fracture. Finally, it may also be concluded that *CMOD* and crack length affect energy more than load.

3. Fuzzy Model on Compact Tension Energy

The use of fuzzy logic modelling in material science is widespread because of its ability to find very accurate linguistic representation of very complex non-linear systems thus enhancing interpretability (transparency) and simplicity of the process [11]. Fig. 3 shows a typical structure of a fuzzy logic system (*FLS*). The fuzzifier component maps a real input in R^D into a fuzzy set. A fuzzy set (*FS*) extends the capabilities of a crisp set by allowing elements have degree of membership in the set. So the **fuzzifier** provides the degree of membership that the real input belongs to a particular fuzzy set. The **Fuzzy inference engine** (*FIS*) is the heart of the *FLS* and it determines how the fuzzified input is combined with the rules contained in the **Rule Base** to produce a fuzzified output. Finally the **Defuzzifier** produces a crisp output.

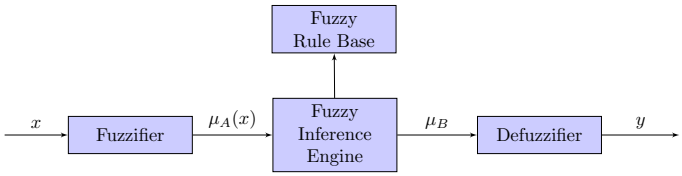


Figure 2: Structure of Fuzzy Systems. $\mu_A(x)$ is the membership function value of input x and μ_B is the fuzzy output to be defuzzified after rule aggregation.

The rules of a fuzzy system is usuallly of the form:

$Rule^m$: IF x_1 is A_1^m AND ... AND x_n is A_n^m THEN y^m is B^m.

Where m is the number of rules, n is the number of inputs. $A_1^m, ...and A_n^m$ are fuzzy sets in the input space and B^m is a fuzzy set in the output space. In a Takagi-Sugeno-Kang (TSK) FLS, the B^ms are replaced by $g^m(x_1, ..., x_n)$ which represents a function of the inputs variables. Usually this function is just a linear function of inputs.

Expert knowledge is required to build a fuzzy model but mechanisms for automatic rule generation from data may be used when only data is available. Several types of adaptive fuzzy modelling may be found in the literature [12][13]. The approach used in this work in eliciting the first part of the fuzzy model is similar to that of [14] and [15], whereby hierarchical clustering is used to determine the initial number of clusters (rules) and then the initial structure of the fuzzy logic model. Data clustering has been shown to be an effective initial fuzzy logic model generation. To improve prediction accuracy, this initial model is optimised using the gradient descent algorithm. The next subsections explain this process of model elicitation in greater detail.

3.1. Model Structure

The initial structure of the FLS was found using an improved hierarchical clustering scheme. The parameters of this initial model are then optimised using the gradient descent algorithm. The procedure for initial and final structures

determination is subsequently explained in detail.

3.1.1. Clustering

Partitional and hierarchical clustering are the two most popular clustering techniques. Partitional clustering involves associating each data point to some pre-specified number of clusters [16]. While partitional clustering is computationally fast, the usually suffer from the problem of reproducibility and the need to specify the number of clusters. Hierarchical clustering on the other hand can optimally select the number of clusters [17]. In this work we employ the improved hierarchical clustering algorithm developed by [14]. This methodology exploits the accuracy and reproducibility of hierarchical clustering and the relatively computationally efficient partitional clustering. The clustering methodology is described as follows:

1. The desired number of clusters N_c and the maximum allowed threshold N_{max} are chosen. Usually we choose $N_{max} \geq N^{1/2}$.
2. if $N \leq N_{max}$, begin the agglomerative complex-link algorithm (ACL) as described in [17] to classify the data into the pre-specified N_c clusters and then end clustering. If $N > N_{max}$, go to the next step.
3. Separate the data randomly but equally into i groups. Where $i = \lceil (N/N_{max}) \rceil$[2].
4. The data in every group is classified into j sub-clusters using the normal ACL algorithm. $j = \lfloor (N/i) \rfloor$[3].
5. A representative data from every sub-cluster is selected. This selected data is the data point closest to the centre of every sub-cluster.
6. A representative data set is constructed to include all the $i \times j < N_{max}$ data points.
7. The representative data set is now clustered using the normal ACL clustering algorithm.

[2] $\lceil x \rceil$ is called a ceiling function and returns the smallest integer value greater than x.
[3] $\lfloor x \rfloor$ returns the largest integer less than x.

8. Every representative data point is replaced with original data set in its corresponding sub-cluster.

The clustered data points is used to construct the initial fuzzy model. The elicited model is composed of N_c fuzzy rules. If C_n represents the nth cluster, DN_n the number of data points in C_n, then fuzzy rule (R_n) corresponding to the C_n fuzzy rule is given as:

$$R_n\text{: IF } x_1 \text{ is } A_1^n \text{ AND } x_2 \text{ is } A_2^n \text{ AND ... } x_D \text{ is } A_D^n \text{ THEN } y \text{ is } Z_n. \quad (1)$$

Where for $n = 1, 2, \cdots, DN_n$; $x = [x_1, x_2, \cdots, x_D]$ is the input variable to be fuzzified, A_i^n is the ith antecedent fuzzy set (FS) for the nth rule for $i = 1, 2, \cdots, D$ and Z_n is the consequent FS of the nth rule.

3.1.2. Fuzzy Modelling

The membership function (MF) selected for each FS is the Gaussian MF because a Gaussian MF allows for easy exploration of the whole data-space and produces a smooth model surface which can improve model generalisation. Additionally, clustering results can easily be mapped into the Gaussian MF. The centre of the MF, c_n^i, is the centre of the corresponding dimension which is gotten from the cluster centres. The width of each FS, σ_i^n, is calculated by solving the following equation:

$$\min_j(\mu_{A_i^n}(x_i^{nj})) = \min_j(\exp(-\frac{(x_i^{nj} - c_i^n)^2}{(\sigma_i^n)^2})) = Th \quad (2)$$

Where $j = 1, 2, \cdots, DN_n$. The generality of the MF is guaranteed by setting a suitable threshold. A threshold value ($Th = 0.5$) is selected because it produces a not too wide nor not-too narrow MFs which may ensure the generaility of the initial MF. The initial MF (defined by its width and centre) is then optimised to produce a more accurate model.

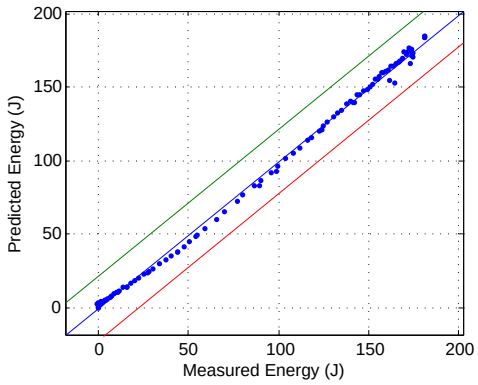


Figure 3: Fuzzy modelling prediction results on the training data without error compensation.

3.1.3. Gradient Descent

To improve on the accuracy of this initial model, the gradient descent optimisation algorithm is used to fine tune the parameters (c_i^n and σ_i^n) and the Root Mean Square Error ($RMSE$) is chosen as the performance index. The parameter learning algorithm for the kth iteration is given by the following set of equations:

$$\Delta c_i^n = \lambda_c \cdot (y_k - y_{dk}) \cdot (Z_n - y_{dk}) \cdot \frac{x_i^{nj} - c_i^n}{(\sigma_i^n)^2} \cdot \frac{\mu_n}{\sum \mu_n} \tag{3}$$

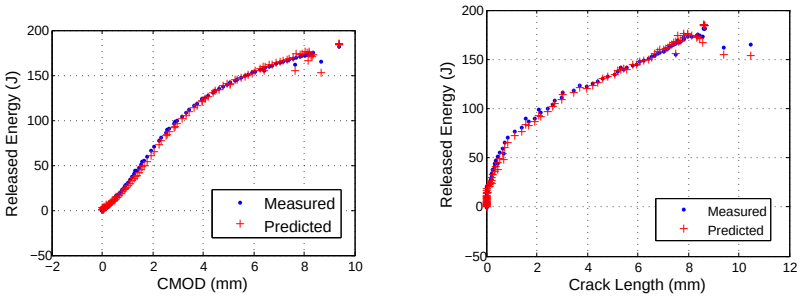
$$\Delta c_i^n = \lambda_\sigma \cdot (y_k - y_{dk}) \cdot (Z_n - y_{dk}) \cdot \frac{x_i^{nj} - c_i^n}{(\sigma_i^n)^3} \cdot \frac{\mu_n}{\sum \mu_n} \tag{4}$$

$$\mu_n = \exp\left(-\frac{(x^n - c^n)^2}{(\sigma^n)^2}\right) \tag{5}$$

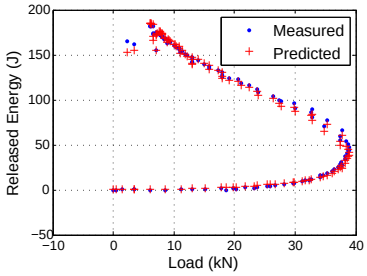
λ_c and λ_σ are the learning rates of centre and width parameters respectively.

3.2. Results

Fig. 3 shows the modelling results of the fuzzy model with 15 rules on the training data. The $RMSE$ is 3.0864. We observe that at the beginning of the fracture propagation process (lower energy region), the fuzzy model predicts the released energy very well but the performance of the model deteriorates at the end of fracture (high energy region).



(a) CMOD v.s. Released Energy (b) Crack Length v.s. Released Energy



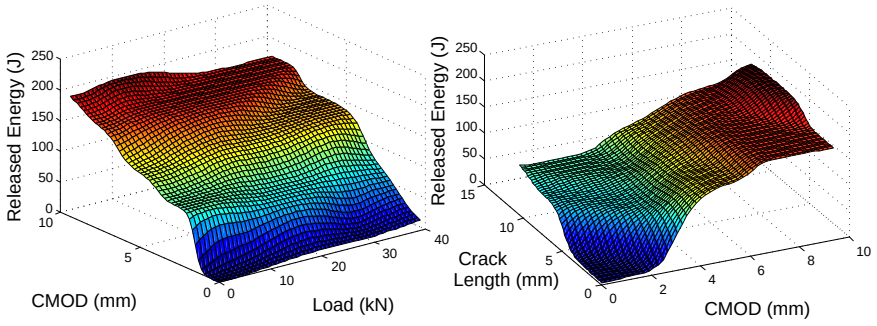
(c) Load v.s. Released Energy

Figure 4: The distribution of data (inputs v.s. predicted and measured released energy) of fuzzy modelling on training data.

Fig. 4a shows the data distribution of *CMOD* against real and predicted energy. This corroborated our claim that when the fracture is completed at the high energy regions, performance of the elicited fuzzy model deteriorates. The increased prediction error in these high energy regions may be due to the fast change in process variables observed during the fracture propagation process. Fig. 4b shows the crack length against real and predicted released energy. Similar deductions may be found as in the *CMOD* plot of 4a.

The curve in Fig. 4c is the released energy as a function of the load applied. The failure starts from the left corner and ends at the top left corner (high energy).

The response surfaces for the elicited fuzzy model are shown in Fig. 5. The surfaces provide one with an idea of the interactions between the inputs and output variables. We observe that the energy released is low when crack length and *CMOD* are both small. The released energy increases non-linearly



(a) CMOD & Load v.s. Released Energy

(b) CMOD & Crack Length v.s. Released Energy

Figure 5: Surface of the elicited fuzzy model with 15 rules but without error compensation.

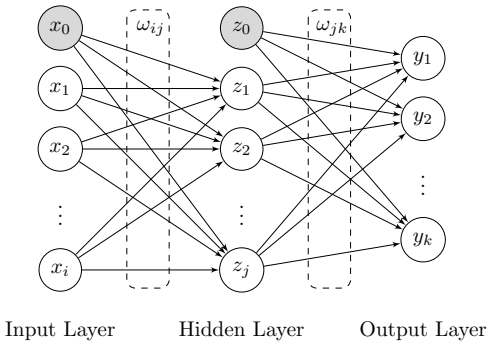


Figure 6: Neural Networks Structure.

as the crack length and *CMOD* increase. The increase in load does not seem to significantly increase the released energy keeping other variables constant because of the fixed crack propagation speed. The crack length does not also have significant effect on the released energy because there cannot be too much released energy without large shape change in the specimens.

4. Artificial Neural Network Modelling of Compact Tension Energy

The second modelling framework used in this paper is the Double-Loop Neural Network Training procedure. The structure of the neural networks used in this research is shown in Fig. 6.

4.1. Model Structure

The training procedure of the neural networks model is done in 3 steps namely:

1. With x_o and z_o set to 1, all the weights (ω_{ij} and ω_{jk}) are initialised randomly.
2. In the **forward process**, the network outputs are calculated according to the input values as defined by the set of equations below:

$$z_j = f_j\left(\sum_i \omega_{ij} x_i + b_j\right) \tag{6}$$

$$y_k = f_k\left(\sum_j \omega_{jk} z_j + b_k\right) \tag{7}$$

   Where ω_{ij} is the weights of the connection from the ith input neuron to the jth hidden neuron. ω_{jk} is the weights of the connection from the jth hidden neuron to the kth output neuron. z_j is the jth neuron output. f_j and f_k are the activation functions of the hidden and output neurons respectively and y_k is the output from the kth neuron.
3. The **Backward Process** changes the weights according to a pre-specified error performance. The training procedure used in this research is based on the Levenberg-Marquardt optimisation which has proven to have very fast convergence to an optimum solution for the weights.

A neural networks with 8 hidden neurons is trained in this paper. The training procedure is implemented via a double loop training process as given in Fig. 7 [18]. Where $iMax = 10$ and $jMax = 50$ are the inner loop and outer loop number of iterations respectively.

The advantage of the double loop training procedure is that it is able to monitor the training process while recording the optimal network structure in the process. The inner loop represents the BP training progress of a NN, where i is the training step, Each inner loop will lead to a new trained network, whose

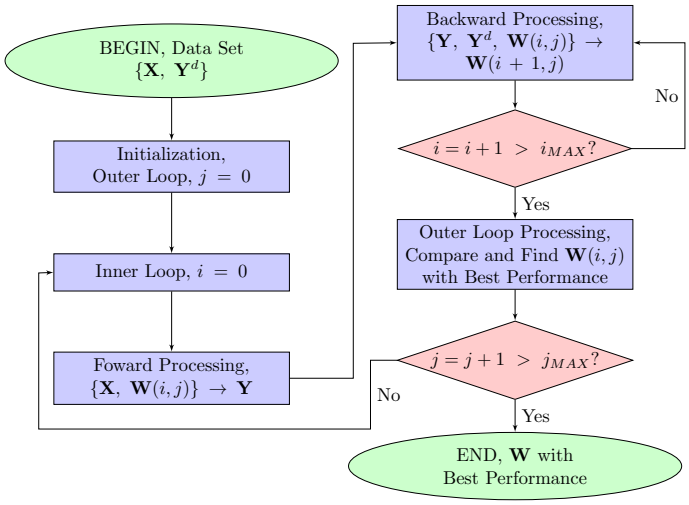


Figure 7: Neural Network Double Loop Training procedure

Trials	1	2	3	4	5	Average
RMSE	5.619	5.073	5.804	6.979	6.305	5.956

Table 3: Performance of the neural networks model across five (5) runs. We note the variability in performance across each of the runs.

performance will be recorded and compared in the outer loop according to the pre-defined performance criteria.

4.2. Results

In the course of training the network, the data set was divided into 3 portions: training data (60%), Validation data (25%) and testing data (15%). The training data is used in the weight updating process, the validation data is used to prevent the model from overfitting so that optimisation process is stopped when the error increases, and the testing data is used to assess the performance of the elicited model. Due to the variability in the performance of a neural networks model, several training runs were performed and the result of some of 5 of the runs is given in table 3.

Figs. 8 and 9 show the results of using one of the trained neural networks models. The explanation of the figures follows directly from those obtained in

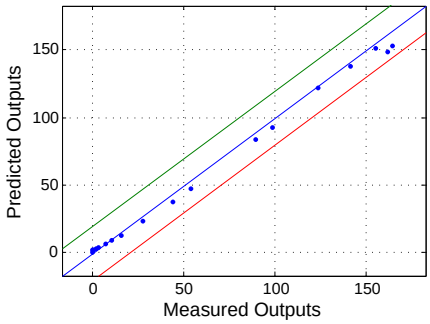
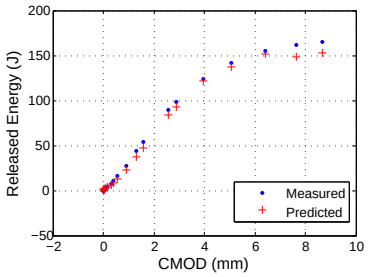


Figure 8: Neural networks prediction results on the testing data set.

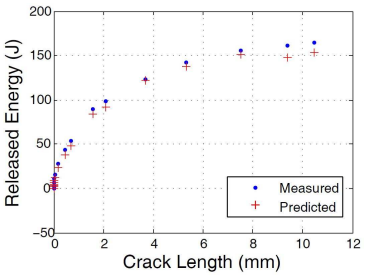
the fuzzy modelling results. Fig. 10 shows the surface plot of the inputs against the output. It is perhaps worth noting that the surface is not as smooth as that obtained from the fuzzy model in Fig. 5 which may be due to the fact that the Neural network model is fitting the model according to the training data while the fuzzy model with 15 rules can only provide limited inference. It is worth noting at this stage that to improve the robustness of the elicited models, the training algorithms used for the neural networks and fuzzy models were performed several times using randomly selected subsamples of the training data at each training run in a manner similar to k-fold cross validation. The models with the best performances were selected.

5. Error Compensation Using A Gaussian Mixture Model

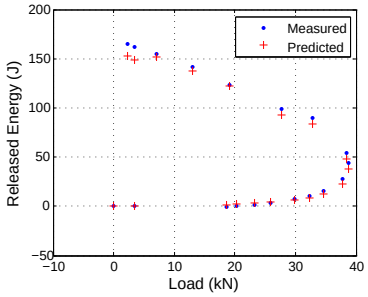
The previous sections have shown and compared the results between proposed the modelling frameworks. We observe that in both models, there seems to be certain regions of the data space where the performances of the model seem to deteriorate. It is the intention of this section of the paper to show how we may improve on the performances especially in the low-accuracy regions using an error compensation technique. The error compensation strategy tries to compensate for the errors in prediction when new factors/new environment are introduced into the prediction process (which may not be observable during data collation), or when eliciting a new model is prohibitively computationally



(a) CMOD v.s. Released Energy

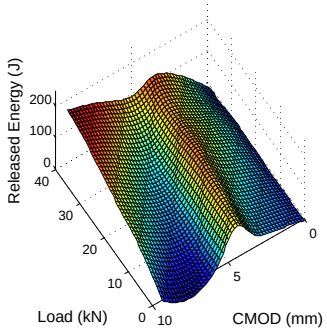


(b) Crack Length v.s. Released Energy

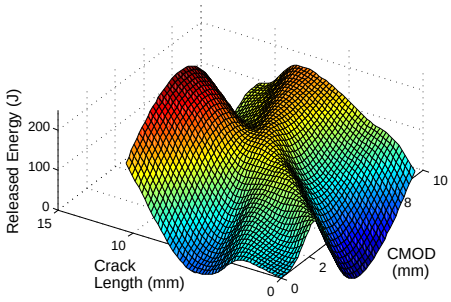


(c) Load v.s. Released Energy

Figure 9: Distribution of the input variables v.s. the measured and predicted released energy using the double loop neural networks procedure.



(a) CMOD & Load v.s. Released Energy



(b) CMOD & Crack Length v.s. Released Energy

Figure 10: Surface of the input variables against predicted released energy using the double loop neural networks model.

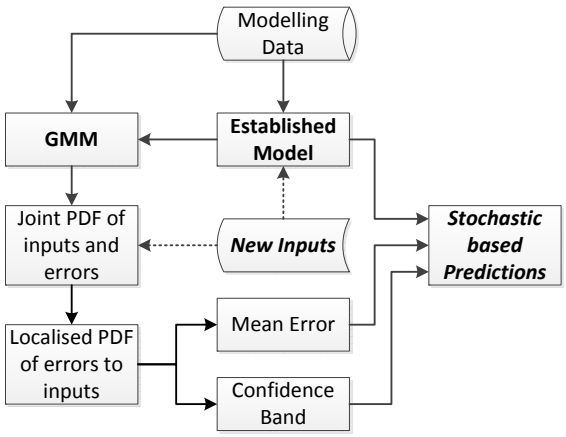


Figure 11: Error Compensation Block Diagram Using the GMM framework.

expensive. The error compensation strategy will save on computational costs since a new model needs not be developed if improvement in performance is desired after intial model design. Consequently, the error compensation strategy provides the field engineers and model designers an emergency tool to compensate the error in a speedy manner.. The error compensation strategy employs the Gaussian mixture modelling (GMM) paradigm. The GMM is a mature method of clustering and density estimation [19]. We use this GMM to monitor the distribution of the errors by applying the GMM process on the errors induced in the predictions. From the observed distribution error a compensation error is introduced into the model validation stage of the model elicitation. Fig. 11 shows the error compensation block diagram.

5.1. Construction of a GMM

The GMM data set consists of $X_e = (x_1^e, x_1^e, \cdots, x_1^n)$ which is a combination of the the inputs $X = (x_1, x_2, \cdots, x_n)$ and the errors on prediction on each data inputs $E = (e_1, e_2, \cdots, e_n)$. We note that the dataset used in the development of the GMM need not necessarily be that of the training data set. The testing data may also be used in the construction of the GMM model. However, for the dataset used in fitting the GMM, it is believed that the best choice should

be the actual field data which are totally new and different from the training and testing data used in model design stage. Since, we did not have any field data, it was assumed that the data we have fully reflect the 280 environment under investigation because there is a risk of overfitting from using just the training data, both the testing and training data have been used in fitting the GMM model. The construction of the *GMM* compensation scheme includes the following steps:

1. For randomly chosen parameters, initialize a *GMM*. The number of the Gaussian mixture components is calculated according to the Bayesian Information Criterion explained further in step 6. The initial parameters $(\omega_k, \mu_k, \sigma_k)$ are initialised using K-means clustering algorithm. Here ω_k represents the mixing coefficient (weight) of the kth cluster/component, μ_k and σ_k (covariance matrix) are the centre and width of the kth component respectively. The *GMM* is thus defined as follows:

$$P(x_n^e|\omega, \mu, \sigma) = \sum_{k}^{K} \omega_k g(x_n^e|\mu_k, \sigma_k) \tag{8}$$

   Where $P(x_n^e|\omega, \mu, \sigma)$ is the probability that x_n^e, $g(x_n^e|\mu_k, \sigma_k)$ is the probability of the data point x_n^e given that it belongs to the kth Gaussian component for total number of K components.

2. Let $Z_k(x_n^e)$ be the probability that the data point x_n^e is generated by the kth Gaussian component, then according to Bayes' Rule, $Z_k(x_n^e)$ may be calculated as follows:

$$Z_k(x_n^e) = \frac{\omega_k g(x_n^e|\mu_k, \sigma_k)}{\sum_{k=1}^{K} \omega_k g(x_n^e|\mu_k, \sigma_k)} \tag{9}$$

3. Let $\overline{\omega}_k$, $\overline{\mu}_k$ and $\overline{\sigma}_k$ be the estimated weight, mean and radius respectively.

These are computed as follows:

$$\begin{aligned} \overline{\omega}_k &= \frac{1}{N}\sum_{n=1}^{N} Z_k(x_n^e) \\ \overline{\mu}_k &= \frac{\sum_{n=1}^{N} Z_k(x_n^e)x_n^e}{\sum_{n=1}^{N} Z_k(x_n^e)} \\ \overline{\sigma}_k &= \frac{\sum_{n=1}^{N} Z_k(x_n^e)(x_n^e-\mu_k)(x_n^e-\mu_k)^T}{\sum_{n=1}^{N} Z_k(x_n^e)} \end{aligned} \tag{10}$$

N is the total number of data points

4. The next step is to compute the likelihood as follows:

$$P(X_e|\omega,\mu,\sigma) = \prod_n \sum_k Z_k(x_n^e) \tag{11}$$

5. Set the estimated parameters ($\overline{\omega}_k$, $\overline{\mu}_k$ and $\overline{\sigma}_k$) as the parameters of the next iteration and iterate steps 3 and 4 until the following condition is satisfied or the predefined maximum number of iterations is reached

$$\ln P(X_e|\overline{\omega},\overline{\mu},\overline{\sigma}) - \ln P(X_e|\omega,\mu,\sigma) < \epsilon \tag{12}$$

ϵ is a small number which was set to 10^{-4} in our case.

6. The number of Gaussian components used in the mixture modelling process was chosen according to the Bayesian Information Criterion (*BIC*) after fitting the *GMM* for different number of Gaussian components. The *BIC* is given as follows:

$$BIC = -2\log P(X^e|\omega,\mu,\sigma) + K\log N \tag{13}$$

Equation 13 shows that the *BIC* favours a relatively large number of Gaussian components and this equates to low values of the *BIC*. In this work, we have chosen the number of components that has a relatively low value of *BIC* but permits feasible computational expense.

The priori probability $P(e|x_i)$ gives the probability of the error for input data point x_i. This may be calculated according to the Bayes' rule as follows:

$$\begin{aligned} P(e|x_i) &= \frac{P(x_i, e)}{P(x_i)} \\ &= \frac{P(x_i, e)}{\int P(x_i, \xi) d\xi} \\ &= \frac{P(x_i, e)}{\int \sum_{k=1}^{K} \omega_k g(x_i, \xi | \mu_k, \sigma_k) d\xi} \\ &= \frac{P(x_i, e)}{\sum_{k=1}^{K} \omega_k \int g(x_i, \xi | \mu_k, \sigma_k) d\xi} \end{aligned} \tag{14}$$

The expected error can consequently be calculated as follows:

$$\overline{e}(x_i) = \int e \cdot P(e|x_i) de \tag{15}$$

It is this estimated error that is used in the compensating inference. This estimated error may also be used to give the confidence band in predictions of the of the model as calculated by the equation below:

$$Std(e(x_i)) = \sqrt{\int (e - \overline{e})^2 \cdot P(e|x_i) de} \tag{16}$$

It is easily seen that the error compensated output is given the following:

$$y_i^c = y_i - \overline{e}(x_i) \tag{17}$$

The $\overline{e}(x_i)$ can either be positive or negative. A negative $\overline{e}(x_i)$ means there is an under-estimation of the predicted output while a postive error means there is an over-estimation of the predicted output. By virtue of equation 17, in the case of under-estimation, the absolute value of the error must be added to the predicted output.

For a *GMM* compensator, the time complex is $O(3k + kn + tn + 2ktn) = O(ktn)$ for random clustering, $O(ktn + kn + tn + 2ktn) = O(3ktn + kn + tn) =$

$O(ktn)$ for k-means clustering. Where k is the number of Gaussian components and t the number of iterations. Hence, after cancelling the coefficients, the time complex for the *GMM* compensator should be $O(n)$.

5.2. Compensated Fuzzy Model

The same data used in developing the fuzzy model were used in developing the GMM. Fig. 12 shows the error distribution for the output (released energy in Joules (J)). Two sample data points are taken so as to be able to visualise the distribution of the errors $P(e|x_{s1})$ and $P(e|x_{s2})$ for given inputs x_{s1} and x_{s2} respectively.

In deciding on the number of Gaussian components, the *BIC* criterion was used. Fig. 5.2 shows the *BIC* plot which favours higher number of Gaussian components which unfortunately increases the computational costs. The value of $k = 5$ was chosen which represents a trade-off between a good model fit and a reasonable computational burden. It is worth noting at this stage that only the crack length and *CMOD* input variables were used to train the *GMM* as the load variable was not used as it was found not to affect the distribution of the errors (independence). The two sample points (x_{s1} and x_{s2}) were chosen because x_{s1} leads to medium error (1.8622, $y_{s1} = 162.4522$) while x_{s2} leads to the error at the largest output value. The distribution of the data points are shown in Fig. 14. It is seen that error distribution reaches a maximum around point (1.36, 0.1721) and the mode is 1.36 joules compared to the actual value of 1.8622. Thus, using the error compensation formula of equation 17, the expected error is calculated as $\overline{e}(x_{s1}) = 1.3159$. The error variance at this point is also calculated to be $STD(e(x_{s1})) = 2.3180$. The compensated output is then calculated as $y^c_{s1} = 161.1463$ with a variance of 2.3180.

The second selected sample data point s_2 is seen to have a greater error of 4.3064. The output without error compensation is found to be $y = 186.4264$. As shown in Fig. 14 , the mode is at point (2.64,0.1723) which gives the most probable error as 2.64 compared to the error obtained as 4.3064. The compensated output was found to be y^c_{s2} with a variance of 2.3194 following exactly the

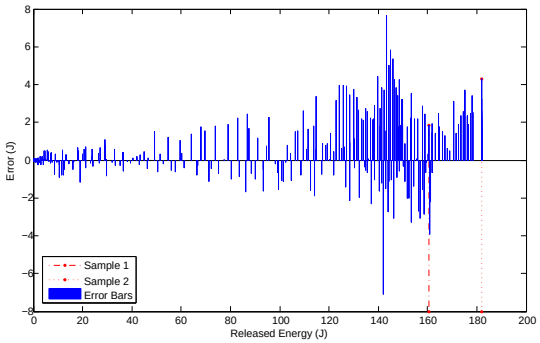


Figure 12: Distribution of error on GMM_{f1}.

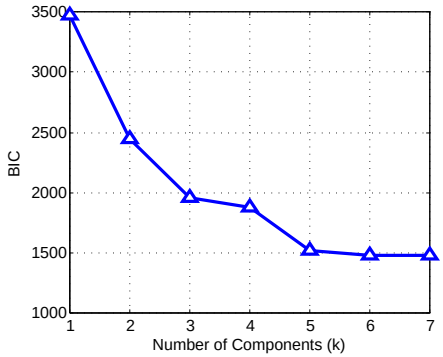


Figure 13: Bayesian Information Criterion Plot for GMM_{f1}. It is seen that the *BIC* criteria favour a more complex model (large number of Gaussian components). The number of Gaussian components (k) was set to 5 in this research. The value chosen satisfies the trade-off between feasible computational speed and good fitting.

same procedure as sample data point s_1.

The procedure described above was followed for all the data points and the compensated output found (called GMM_{f1}).

Two new data sets X_2 and X_4 are plotted for with and without error compensation as shown in Figs. 15 and 16. The *RMSE* for Y_2 and Y_4 without error compensation are 2.7832 and 4.1747 respectively.

It is observed that there are larger errors in the high energy regions than in the lower energy regions. There was an improve in modelling performance as shown in the figures with *RMSE* for Y_2^c and Y_4^c being 2.7348 and 3.4944 respectively.

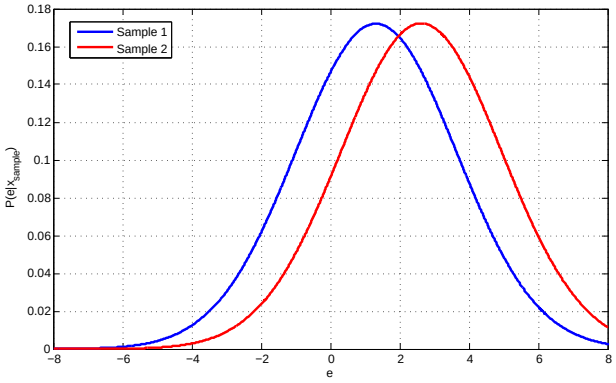
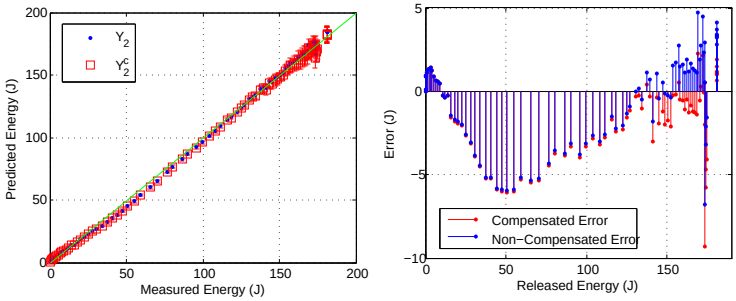
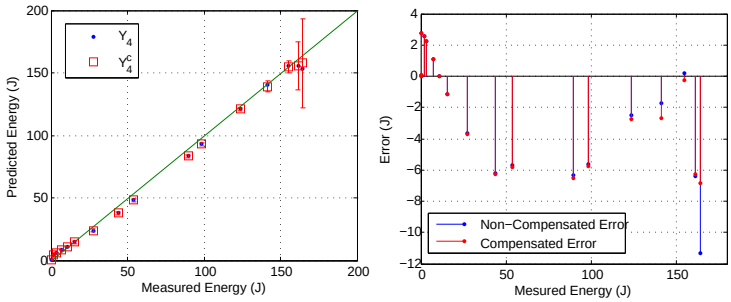


Figure 14: Distribution of two errors of the fuzzy model.



(a) Measured v.s. Predicted (b) Energy v.s. Error

Figure 15: Error distribution for Y_2 and Y_2^c for GMM_{f1}



(a) Measured v.s. Predicted (b) Energy v.s. Error

Figure 16: Error distribution for Y_4 and Y_4^C for GMM_{f1}

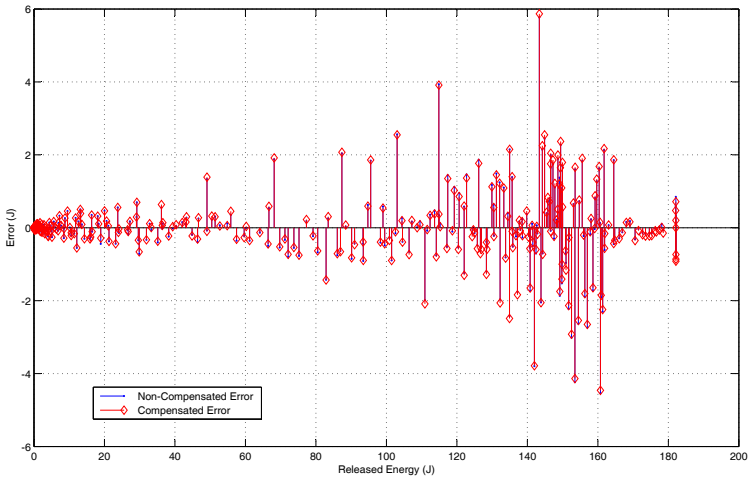


Figure 17: NN GMM_{n1} error distribution before and after applying GMM_{n1} on Training Set

5.3. Compensated Neural Networks Model

The neural network model trained was used in developing the *GMM* and the training data was used in generation of the error distribution. It was discovered that after fitting the error distribution and the outputs compensated for these errors the *RMSE* after compensation (0.9914) was larger than before compensation was applied (0.9904). We have called this the GMM_{n1}. Fig. 17 shows the distribution of these errors before and after compensation.

It is evident that GMM_{n1} cannot accurately compensate for the errors in modelling process. The bad fitting of the GMM_{n1} is due to the fact that the *RMSE* was initially very low without compensation especially for the training data set which means the fitted *GMM* cannot significantly generate meaningful error compensations.

To remedy this problem, dataset $X2$ was combined with dataset $X1$ (part of the training data) to construct a new *GMM* which we refer to here as GMM_{n2}. The error distributions before and after error compensation are shown in Fig. 18. The *RMSE* of 1.8473 and 1.4003 were obtained without and with compensation respectively which shows that GMM_{n2} provides a better modelling accuracy and better error compensation than GMM_{n1}. It can also be seen that at low

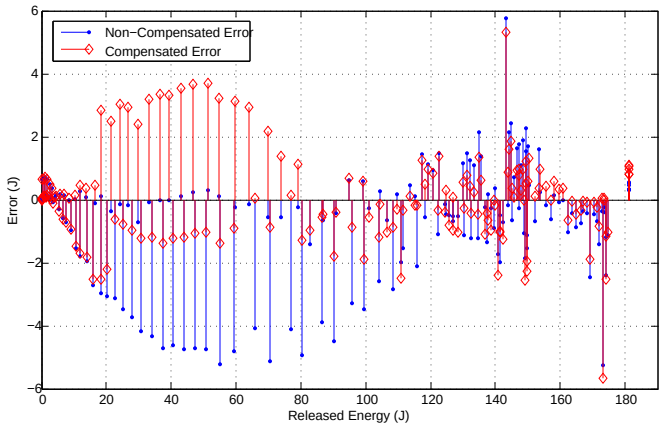
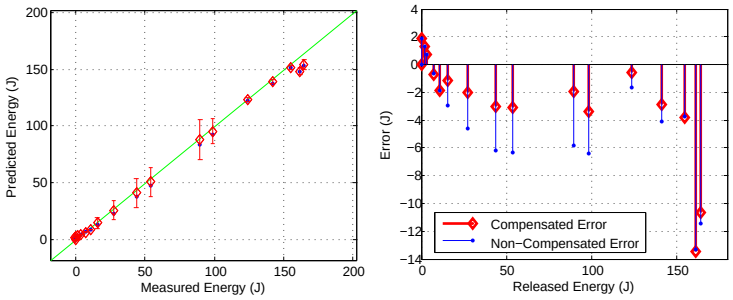


Figure 18: NN GMM_{n2} error distribution before and after applying GMM_{n2} on Training Set



(a) Measured v.s. Predicted (b) Energy v.s. Error

Figure 19: Error distribution for Y_4 and Y_{C4} for GMM_{n2}.

energy regions, better error compensations were observed when the errors are negative.

Data set $X4$ was also used to ascertain the performance of GMM_{n2} as done for GMM_{f1}. The results are shown in Fig. 19. This figure shows that the$RMSE$ before compensation was 5.0736 and after compensation was 4.2013, indicating a 17% increase in modelling accuracy on the holdout data set (X4). The fitted parameters for GMM_{n1} is shown in Table 4. The BIC analysis as shown in Fig. 20 An optimal value of 4 was chosen for k because repeated simulation runs indicated no significant increase in modelling performance for $BIC < 1200$.

5.4. Comparison with Benchmark Models (ANFIS)

k		4			
ω		2.123E-1	3.277E-1	1.587E-1	1.300E-2
μ	CMOD(mm)	2.252E-1	7.067	1.615	3.646
	Crack Length (mm)	3.213E-9	7.768	7.939E-1	3.874
	e (mm)	9.840E-2	1.712E-1	-3.725	-3.725E-1
σ	CMOD (mm)	4.060E-2	1.862	7.565E-1	3.3291
	Crack Length (mm)	1.000E-3	8.143E-1	7.738E-1	7.1025
	e (mm)	1.214E-1	2.353	1.119	6.054E-1

Table 4: Fitted Parameters for GMM_{n2}

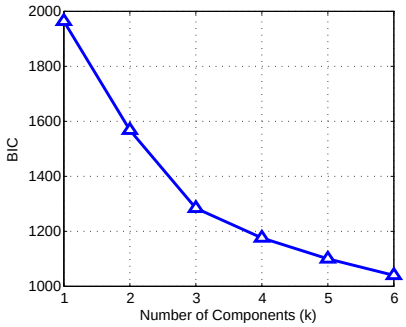


Figure 20: BIC analsysis for GMM_{n2}.

	Before		After	
	Training	**Testing**	**Training**	**Testing**
Fuzzy	1.7291	4.1747	1.5926	3.7455
ANN	1.8473	5.0736	1.4003	4.2013
ANFIS	2.4742	5.4848	1.9420	5.0134

Table 5: Comparison of Results (RMSE) of elicited models with ANFIS before and after error compensation.

The proposed modelling schemes were compared with results from using the adaptive neuro-fuzzy inference system (ANFIS) for with and without GMM compensation. The ANFIS model was elicited for different number of *MFs* in each input space. 10 fold cross-validation on the training data was used select the best parameters of the ANFIS model. The optimisation process was also performed 10 times for different numbers of MFs in the input space. The best performance was consistently found to be an ANFIS model with 3 MFs. The average performance of the ANFIS model with 3 MFs (27 rules) is as shown in Table 5. Data set X_4 was used as the testing data which represents the 15% part of the whole data used in testing the elicited fuzzy and neural network models.

It can be seen that the proposed modelling frameworks were found to have better generalisation performance than ANFIS. However, in all models, the *GMM*-based error compensation strategy was able to improve the modelling performances for both training and testing data sets.

6. Analysis and Conclusion

The fitted *GMM* models clearly reflect the distribution of the errors of the elicited models which can be fed-back into the modelling process for error compensation and confidence bands without the need to train the model all over again. This can significantly save computational costs. However, one must be careful in implementation as it was observed that unlike the *GMM* model fitted using the error from the fuzzy model, a more accurate model such as the one

driven by neural networks may depreciate the performance of the final elicited model. This is because the models are already very accurate and further attempts at error compensation may lead to performance degradation. As already shown, a better approach is to train the *GMM* model on an entirely new data set as was done on the neural networks model using the training data combined with data set $X4$. It is worth noting that the highly non-linear surface of the neural networks models may cause degradation in interpretability of the process. The fuzzy model, however provides a smooth surface plot of input/output mapping which may enhance interpretability.

Finally, it was further observed that the error bars are significantly lower in the lower energy regions than in the higher energy regions which corroborated our findings of increased uncertainty in the final stages of the fracture propagation process.

References

[1] G. Buzzichelli, L. Scopesi, Fracture propagation control in very high strength gas pipelines, Revue de Métallurgie 97 (11) (2000) 1409–1416.

[2] S. Acharyya, S. Dhar, A complete gtn model for prediction of ductile failure of pipe, Journal of Materials Science 43 (6) (2008) 1897–1909.

[3] M. Karabin, F. Barlat, R. Shuey, Finite element modeling of plane strain toughness for 7085 aluminum alloy, Metallurgical and Materials Transactions A 40 (2) (2009) 354–364.

[4] J.-B. Leblond, G. Perrin, J. Devaux, An improved gurson-type model for hardenable ductile metals, European journal of mechanics. A. Solids 14 (4) (1995) 499–527.

[5] P. Reed, M. Starink, S. Gunn, I. Sinclair, Invited review: Adaptive numerical modelling and hybrid physically based anm approaches in materials engineering–a survey, Materials Science and Technology 25 (4) (2009) 488–503.

[6] G. Zhang, M. Mahfouf, G. Panoutsos, S. Wang, A multi-objective particle swarm optimization algorithm with a dynamic hypercube archive, mutation and population competition, in: Evolutionary Computation (CEC), 2012 IEEE Congress on, IEEE, 2012, pp. 1–7.

[7] G. Zhang, M. Mahfouf, Q. Zhang, S.-A. Gaffour, J. Yates, S. A. Soberanis, C. Pinna, G. Panoutsos, Systems-modelling of compact tension energy in high strength pipeline steel, in: World Congress, Vol. 18, 2011, pp. 12126–12131.

[8] M. Çöl, H. Ertunc, M. Yılmaz, An artificial neural network model for toughness properties in microalloyed steel in consideration of industrial production conditions, Materials & design 28 (2) (2007) 488–495.

[9] O. Ozbulut, C. Mir, M. Moroni, M. Sarrazin, P. Roschke, A fuzzy model of superelastic shape memory alloys for vibration control in civil engineering applications, Smart materials and structures 16 (3) (2007) 818.

[10] S. A. Soberanis, 3d cafe modelling of ductile fracture in gas pipeline steel, Ph.D. thesis, University of Sheffield (2007).

[11] M.-Y. Chen, D. Linkens, D. Howarth, J. Beynon, Fuzzy model-based charpy impact toughness assessment for ship steels, ISIJ international 44 (6) (2004) 1108–1113.

[12] J. Abonyi, Fuzzy model identification, Springer, 2003.

[13] G. Feng, A survey on analysis and design of model-based fuzzy control systems, Fuzzy systems, IEEE Transactions on 14 (5) (2006) 676–697.

[14] Q. Zhang, M. Mahfouf, Mamdani-type fuzzy modelling via hierarchical clustering and multi-objective particle swarm optimisation (fm-hcpso), International journal of computational intelligence research 4 (4) (2008) 314–328.

[15] Q. Zhang, M. Mahfouf, A hierarchical mamdani-type fuzzy modelling approach with new training data selection and multi-objective optimisation mechanisms: A special application for the prediction of mechanical properties of alloy steels, Applied soft computing 11 (2) (2011) 2419–2443.

[16] O. Obajemu, M. Mahfouf, L. A. Torres-Salomao, A new interval type-2 fuzzy clustering algorithm for interval type-2 fuzzy modelling with application to heat treatment of steel, in: World Congress, Vol. 19, 2014, pp. 10658–10663.

[17] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM computing surveys (CSUR) 31 (3) (1999) 264–323.

[18] Y. Yang, D. Linkens, M. Mahfouf, A. Rose, Grain growth modelling for continuous reheating processa neural network-based approach, ISIJ international 43 (7) (2003) 1040–1049.

[19] G. McLachlan, D. Peel, Finite mixture models, John Wiley & Sons, 2004.