

THE APPLICATION OF NEURAL NETWORK TO THE DEVELOPMENT OF SINGLE CRYSTAL SUPERALLOYS

Y.S. Yoo¹, I. S. Kim¹, D. H. Kim¹, C.Y. Jo¹, H. M. Kim², and C.N. Jones³

¹KIMM, 66 Sangnam-dong, Changwon, Kyungnam 641-010, Korea

²KISTEP, 275 Yangjae-dong, Seocho-gu, Seoul, 137-130, Korea

³Rolls Royce plc, PO Box 31, Derby, DB24 8BJ, UK

Keywords: alloy development, neural networks, single crystal superalloys

Abstract

The neural network (NN) method is applied to mechanical properties estimation and alloy development of single crystal superalloys. Databases have been constructed from previous publications and the Rolls-Royce materials database. The Bayesian neural network technique was used for the modeling of mechanical properties of single crystal superalloys in terms of alloy compositions and test conditions. Creep lives, yield strengths, and ultimate tensile strengths of various superalloys as a function of contents of alloying elements are estimated and analyzed. New alloys were designed by calculations of various properties as well as creep rupture lives for millions of compositions, followed by selection of optimized alloys. The developed alloys are made in single crystal form by directional solidification and tested. They exhibit excellent phase stability and creep rupture lives which are better than or equivalent to those of CMSX-4.

Introduction

The prediction of properties by statistical calculations has been utilised in alloy design of superalloys for several decades [1-3]. Ordinary Ni base superalloys contain more than 10 elements and it is hard to predict properties such as creep rupture life without relying on previous experimental data. Thus the most crucial step in computational alloy design is the accurate prediction of properties for unseen compositions based on the information extracted from experimental data.

Alloy composition is not the only variable affecting the properties of an alloy; microstructural features such as grain size, morphology and volume fraction of precipitates have a significant influence on various properties of materials. When interest is confined to single crystal superalloys, however, it is possible to neglect all the grain boundary related microstructural variables. Moreover most of the single crystal superalloys undergo heat treatment which gives a more or less uniform microstructure -- three dimensionally aligned cubic γ' in a γ matrix [4]. If it is assumed that there is an optimum heat treatment condition for each single crystal alloy composition, most of the microstructural features related to γ' (volume fraction, size, morphology etc.) can be regarded as a function of alloy composition. Thus it is possible to proceed with compositional modelling for single crystal superalloys with reasonable confidence, which is not the case for the polycrystalline superalloy.

Expanding the capability of the original PHACOMP, Harada introduced computational methods to predict phase equilibria and

properties of superalloy [1,2]. In their approach, most of the property prediction was based on statistical treatment of existing experimental data. In the case of creep rupture life prediction, for instance, they organised experimental data at certain creep condition (1040 °C/137MPa) and performed multiple linear regression in terms of alloy composition. They constructed an alloy design system to predict a number of properties as well as creep rupture life, which was utilised for successful superalloy developments [2,3].

By multiple linear regression it is possible to figure out easily the relationship between input and output variables when there is no apparent functional relationship. However, it is oversimplification to assume linearity between compositions and properties for the modelling of real data. If the content of certain elements is increased the creep rupture life might increase; but further increase of the elements might decrease the rupture life again. Moreover, there could be interaction between the actions of each element; variation of the contents of an element could change the functional relationship between other elements and the output.

In spite of its merits of convenience, multiple linear regression has limitations in that we cannot model non-linearity and that we cannot explain coupling between inputs. We could introduce a non-linear function to resolve these problems, but it is difficult to select a particular non-linear function *a priori* by only observing the experimental data, especially when there are dozens of input variables as in the case of a superalloy. Therefore we need more general and flexible regression methods. Artificial Neural Networks (ANN), or simply Neural Networks (NN), have great potential in modelling of complex experimental data.

NN have already been extensively applied in the materials science field for the purpose of experimental data modelling [5,6], and their usage is in ever increasing trend. There are many variants of the NN approach but an approach called Bayesian neural network, formulated by Mackay [7,8] and Neal [9,10], has great merits in terms of precision and confidence of prediction though it often involves complex algorithms and calculations. Bayesian NN was applied to property modelling in steel and superalloys [11-16]; most of the works are based on Gaussian approximation framework proposed by Mackay [7,8].

The purpose of the present work is to investigate the compositional modelling of creep rupture life of single crystal superalloys by Bayesian NN using the Markov chain Monte Carlo (MCMC) approach by Neal [9,10]. The basic philosophy of the modelling is similar to previous works, but better precision in

prediction at the expense of more calculation resources is aimed. New single crystal superalloys are designed on the base of this modeling. According to the results, newly designed alloys showed better or equivalent performance in creep tests compared with those of CMSX-4. This result is regarded as a positive sign that artificial neural network modeling can be a useful tool for real world alloy design work.

Neural Network

Like a human brain NN has a computational structure such that the prediction performance improves as it performs learning (or training) by data [17,18]. In this work only feed-forward back propagation neural networks are considered. The architecture of a feed-forward NN is composed of an input layer, a number of hidden layers, and an output layer (Fig. 1). Each layer contains a number of *nodes* (or neurons) which are fully connected with nodes in neighbouring layers by *weights*. Each node can have bias, and these weights and biases are the important parameters in NN carrying numeric information the network has learned to date.

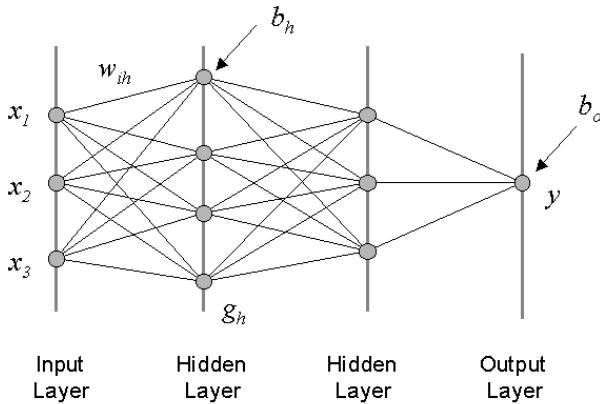


Figure 1. Schematics of neural network architecture.

Let us assume an architecture which contains one input layer with I nodes, one hidden layer with H nodes and one output layer with one node (will be designated as $I-H-1$ architecture). The extension to more general architecture is straightforward. The value of h 'th hidden node $g_h(x)$ is calculated as follows.

$$g_h(x) = \tanh\left(\sum_{i=1}^I x_i w_{ih} + b_h\right) \quad (1)$$

where x_i is the value of i 'th input, w_{ih} the weight connecting i 'th input node and h 'th hidden node, and b_h the bias applied to h 'th hidden node. The *tanh* function, called an *activation function*, acts as a signal function which process non-linear input/output where other types of activation functions are also possible. Once all the hidden nodes are calculated, they act as input for the calculations of nodes in the next layer. Finally output node y can be expressed as:

$$y = \sum_{h=1}^H g_h v_h + b_o \quad (2)$$

where v_h is weight connecting hidden node h to output node, b_o the bias applied to the output node.

In conventional neural network the parameters are updated by so called back-propagation methods. Let's suppose we have a database consisting of n input vectors \mathbf{x}^α and corresponding targets t^α

$$D = \{\mathbf{x}^\alpha, t^\alpha \mid \alpha = 1 \dots n\} \quad (3)$$

The first stage in training is the expression of a cost function which is the measure of network error. One of the common cost function is the sum of squared error $E(\mathbf{w})$ expressed by

$$E(\mathbf{w}) = \frac{1}{2} \sum_{\alpha=1}^n (t^\alpha - y^\alpha)^2 \quad (4)$$

where t is output of real data and y is network output calculated by Eq. (2). E is a function of weight \mathbf{w} because y is function of \mathbf{w} . The NN training process, thus, turns into finding the value of \mathbf{w} which minimizes $E(\mathbf{w})$; that is, the optimisation problem. In statistical terms, it is equivalent to finding the maximum likelihood when a Gaussian distribution for weights is assumed. The gradient descent method is a numerically pertinent algorithm for this kind of high dimensional optimisation problem. The update of weight is therefore expressed as follows:

$$\Delta \mathbf{w} = -\eta \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} \quad (5)$$

where \mathbf{w} is a vector that represents all the network parameters (bias inclusive) and η is a positive constant called learning rate. Because instability in calculation is often caused by high η , we need to set an appropriate η value to get a stable solution. In a conventional NN, the overall training proceeds repeating feed-forward calculation and back-propagation update with all the available experimental data.

Generally the NN prediction becomes accurate as the number of layers and/or the number of nodes increases. If the architecture is too complex, however, the network models all the noise in experiment data and it may give poor generalisation. This phenomenon called over-fitting [17,18] should be avoided since what we want is not the detailed noise but the true generalisation underlying experimental data.

Another problem of a conventional Neural Network is that the model predicts just one single value for the output. Because every prediction involves uncertainty to some extent, we should be able to provide an uncertainty or error bar with each prediction. A potential solution to these problems which can be applied in NN data modelling is Bayesian Inference. Mackay [7,8] and Neal [9,10] laid the foundation of the Bayesian neural network by combining the concept of Bayesian inference and neural networks.

In neural network training the quantity we are to obtain is the conditional probability $p(\mathbf{w}|D)$, that is, probability distribution of weight \mathbf{w} given data D . This is represented by Bayes' rule,

$$p(\mathbf{w} | D) = \frac{p(D | \mathbf{w})p(\mathbf{w})}{p(D)} \quad (6)$$

The conditional distribution of the data $p(D|\mathbf{w})$ called likelihood is obtained from the NN calculation. $p(\mathbf{w})$ is the prior distribution of weight which can be regarded as the initial estimation on the final distribution. The term $p(\mathbf{w}|D)$ is called the posterior distribution and is the final goal of the calculations. $p(D)$ plays role of a normalising factor, ensuring the posterior distribution, Eq.(6), to be unity. Although one confronts an additional numerical burden with introduction of Bayesian inference, Eq.(6) gives the most natural reasoning on learning, that is, the probability parameter is updated on receiving new data.

The main interest in data modelling is to predict the output of unseen input variables. This prediction can be obtained by integration in high dimensional weight space,

$$p(t | x, D) = \int p(t | x, \mathbf{w})p(\mathbf{w} | D)d\mathbf{w} \quad (7)$$

A high dimensional integral like Eq. (7) is very difficult to obtain even with numerical integration. Mackay [7,8] and Buntine and Weigen [19] approached this problem assuming that the posterior distribution is a multivariate Gaussian with mode \mathbf{w}_{MP} (most probable weight). The Gaussian approximation approach cannot handle the presence of multiple modes in the posterior distribution that are likely to be encountered in complex real data modelling. Nevertheless there have been several successful applications of this approach, [8,11-16] taking advantage of the *committee* (average of several separate models) concept.

Another more robust method is direct integration of Eq. (7) by Monte Carlo methods [9,10]. The main disadvantage of this method is known to be the long computation time to obtain the desired distribution. To solve this problem Neal[9] used the Metropolis algorithm and hybrid Monte Carlo approach, and could attain a considerable speed-up in calculation. In this work the MCMC approach is used where the computation time was found to be no longer a serious problem.

The Markov chain Monte Carlo method makes no assumption concerning the form of distribution and performs a robust integration based on sampling. The main idea of Monte Carlo integration is to approximate the integral like Eq. (7) by the mean of the function $f(\mathbf{w})$ sampled from target density $p(\mathbf{w}|D)$,

$$\int f(\mathbf{w})p(\mathbf{w} | D)d\mathbf{w} \approx \frac{1}{n} \sum_{i=1}^n f(\mathbf{w}_i) \quad (8)$$

where $\{\mathbf{w}_i\}$ is a sample of weights drawn from the distribution $p(\mathbf{w}|D)$.

To obtain the correct expectation of Eq. (8), n samples of \mathbf{w}_i should be independent. But independent sampling from high dimensional space is a nearly impossible task. Instead a dependent variable called a Markov chain is generated that has $p(\mathbf{w}|D)$ as its stationary distribution, the estimate will still converge to the true value as n increases [10]. A well known *Metropolis algorithm* is one of the methods to generate such Markov chain.

Neal used the Hybrid Monte Carlo method, which combines the Metropolis algorithm with dynamic simulation, for efficient sampling of the posterior distribution [9]. It makes use of the gradient information provided by a back propagation network to find candidate directions in which changes have a high probability of being accepted. This approach allows the random walk behaviour of Markov chain to be avoided, which speeds up the sampling considerably [9,10].

In data modelling, the relative importance of different inputs can be determined using the Bayesian technique of *Automatic Relevance Determination*(ARD) [10,20,21], based on the use of a separate hyperparameters for each input. If the hyperparameter associated with an input specifies a small standard deviation for weights, these weights are likely to be all small, and the input will have little effect on the output; if the hyperparameter specifies a large standard deviation, the effect of the input is likely to be significant.

Neural network modeling for creep property

Data used in this work came from previous publications, patents and the Rolls-Royce materials database. Although a lot of design parameters are important in alloy design of superalloy interest was confined to creep rupture life which is a very important requisite for superalloy, especially in turbine blade applications. A total 1291 creep rupture life datapoints for 164 single crystal superalloys were collected; 633 data for 118 compositions from publications and patents---658 data for 46 compositions from Rolls-Royce database. It is assumed that every datapoint represents best creep properties for a given composition and test conditions. When a publication presents several datapoints according to different heat treatment condition or single crystal orientation for the same composition and test condition, only the datapoint with longest rupture life was chosen.

The input variables of database are contents of Co, Cr, Mo, W, Re, Al, Ta, Ti, Nb, V, Hf, creep temperature T , and creep stress σ . The output variable is creep rupture life in hours. The constructed database forms a 14 by 1291 spreadsheet where the range of each variable were outlined at Table I. The range of Mo, W, Ta spans from 0.0 to 13.5, 18.6, 16.0 wt% respectively which is wide enough to consider most of the modern single crystal superalloys. The range of Re is from 0.0 to 6.1 wt%, which means the model can predict creep rupture life of 1st, 2nd and 3rd generation single crystal superalloys. About 5% of data out of 1291 datapoint were randomly selected and reserved separately without participating in NN training. Therefore these data act as new experiments for testing the performance of our model

An improvement of NN performance can be expected if the input and output have a relationship close to linear [22]. For this purpose data transformation for temperature and rupture life were tried as follows;

$$\begin{aligned} T' &= 1000/(T + 273) \\ t_r' &= \log t_r \end{aligned} \quad (9)$$

To see the effect of raw data transformation pre-experiments were tried varying the number of hidden nodes. Table II shows the results of neural network training without and with raw data transformation. In either case all fields are normalised to have

mean of zero and a standard deviation of one. We can see the benefits of transformation by comparing the two fields in Table II. In this study the original data were transformed according to Eq. (9) and each variable is normalised with the scale and shift parameters shown at Table I. After neural network training variables were converted into the original scale by corresponding inverse transformations.

Table I. Variable ranges and data preparation

Variable Name	Range	Transformation	Scale	Shift
Co(wt%)	0.0-15.0	None	0.219	-1.516
Cr(wt%)	0.0-16.0	None	0.272	-1.965
Mo(wt%)	0.0-13.5	None	0.685	-0.900
W(wt%)	0.0-18.6	None	0.310	-1.540
Re(wt%)	0.0-6.1	None	0.414	-0.882
Al(wt%)	3.5-7.1	None	2.149	-11.941
Ta(wt%)	0.0-16.0	None	0.283	-1.379
Ti(wt%)	0.0-7.3	None	0.681	-1.111
Nb(wt%)	0.0-3.9	None	2.306	-0.449
V(wt%)	0.0-4.0	None	1.902	-0.443
Hf(wt%)	0.0-0.95	None	13.493	-0.493
Temp(°C)	700-1150	1000/(T+273)	12.215	-9.944
Stress(MPa)	30-1000	None	0.005	-1.445
Life(hours)	0.2-10783	Log	0.674	-3.672

Table II. Effects of raw data transformation (Average Squared Error for training set)

Number of hidden node	Normalization only	Transformation & normalization
5	0.1365	0.1129
7	0.1028	0.0612
9	0.0667	0.0506
11	0.0671	0.0458
13	0.0478	0.0361
15	0.0488	0.0322
17	0.0466	0.0274

A publicly available code written by Radford Neal [23] was used in neural network training and prediction in this study. Pre module and post module such as data transformation and inverse transformation were coded in C language. These programs were run in Linux/X window environment installed in a personal computer. Average training time depends on the architecture of network, but roughly it took about 1 hour in 100 iterations.

The priors for the network parameters (the weights and biases) are defined hierarchically, using hyperparameters that control the standard deviations for network parameters in various groups. The parameters were organised in 4 groups: input-hidden weights w_h , hidden biases b_h , hidden-output weights w_o , and output bias b_o . The priors for the network weights are different for each group of weights.

For w_h , each parameter is picked from a Gaussian distribution with mean zero and standard deviation σ . At the next level the corresponding precision τ , which is expressed by $\tau = \sigma^{-2}$, out of input unit i is given a Gamma distribution with a mean μ_i and a shape parameter α_i . At the final level the mean μ_i is given a Gamma distribution with a mean μ_0 and a shape parameter α_0 . Values were assigned of $\alpha_i = 1$, $\alpha_0 = 0.5$, and $\mu_0 = 25I^2$, where I is the number of inputs. This three layer prior incorporates the idea of Automatic Relevance Determination (ARD)-the hyperparameters, τ_i , associated with individual inputs can adapt according to the relevance of the input; for an unimportant input, τ_i can grow large, thus forcing σ_i and the associated weights to decrease

For b_h and w_o , parameters were given a hierarchical prior consisting of two layers. At first each parameter was given a zero-mean Gaussian prior with standard deviation σ . Again the distribution of σ was specified in terms of a precision $\tau = \sigma^{-2}$ which was given a Gamma distribution with mean μ and shape parameter α . Values were set to $\mu=100$ and $\alpha=0.5$ for b_h and $\mu=100H$ and $\alpha=0.5$ for w_o , respectively, where H is the number of hidden units. The output bias b_o was simply given zero-mean Gaussian priors with a standard deviation one. In Neal's program, these prior specifications are given with the command something like "net-spec logfile 13 h 1 / - x0.2:0.5:1 0.1:0.5 - x0.05:0.5 - 1" where h is the number of hidden nodes.

To see the effect of the number of hidden nodes, NN training was used for a 13-x-1 architectures varying x from 3 to 46. Fig. 2 shows the Average Squared Error (ASE) guessing mean for the training cases and test cases with respect to the number of hidden nodes. Each point was obtained by complete 200 iteration NN training which took several hours. There is a clear tendency of ASE to decrease as the number of hidden node increases for training cases. Although the ASE for test cases were higher than those of training cases we can still observe the tendency of decrease with respect to the number of hidden nodes.

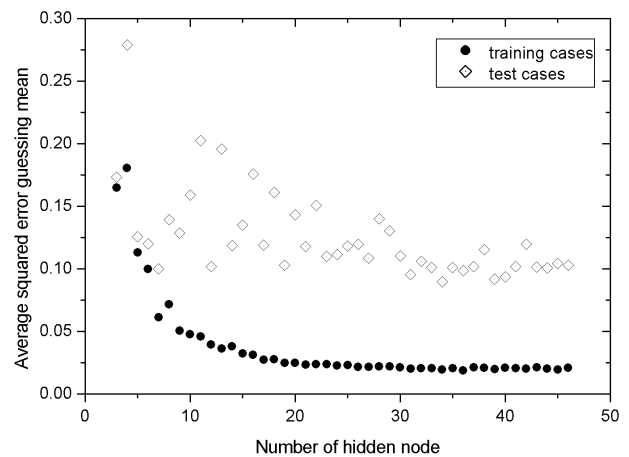


Figure 2. Average squared error guessing mean with respect to number of hidden node.

Regarding model complexity Neal suggested that one should not restrict the number of hidden nodes based on the size of the training set, but should use as complex a model as one can handle

computationally, in order to extract the maximum information from the data [10]. Neal demonstrated this point by valid modelling in the extreme case of infinite networks [10]. In practice, we have to use networks with only a finite number of hidden units that our computational resources will allow us. After evaluation with training and test cases a 13-31-1 architecture was selected which showed a balance of performance and computation time. The real and predicted creep rupture life for this architecture is shown in Fig. 3. The predicted creep rupture lives show good agreement with the real experimental data. The error bar is not shown in Fig. 3 on purpose for clarity.

The prediction for reserved data was tried to see the performance of the model. Fig. 4 shows the real and prediction value for test cases where error bar of the prediction is shown in this case. Multiple correlation coefficient between the prediction and real data was 0.932 implying that this model is reasonably accurate. Note that the error bar length is different for each data point such that they represent true confidence of prediction that is one of the strong point of a Bayesian NN. For instance some data points showed quite long error bar which means the confidence of prediction for these data is not great. Since reserved data are equivalent to new experiments it may be possible to predict the creep life of a new single crystal superalloy with a similar precision as long as the inputs are in the range of Table I.

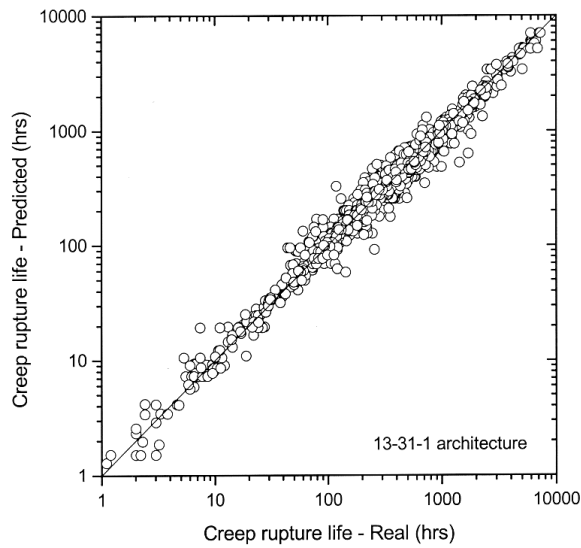


Figure 3. Real and predicted creep rupture life for training cases.

As described before the standard deviation σ_i for weight groups radiated from each input variable represent the influence of each input (ARD). ARD can be used for pruning; a lot of input variables are considered at first, and irrelevant inputs are faded out after training process. In the current case it is believed that all the 13 input variables are important to some extent. Therefore ARD was used to evaluate the relative importance of inputs. The standard deviation for each input weight group showed a little scatter according to the number of hidden nodes that were averaged for the architectures from 13-5-1 to 13-46-1. The results are shown in Fig. 5.

It can be seen that the effects of creep stress and temperature are

the most dominant of all input variables which is in good accordance with reality. Among the alloying elements the effect of Re was the highest showing the importance of this element in single crystal superalloy. The effect of Cr was also high ; which explains negative influence of this element on creep properties.

It is not surprising when it is considered that the level of Re have been increased through 1st, 2nd and 3rd generation single crystal superalloys in spite of its high price. It is known that Re contributes γ strengthening and slows down the γ' coarsening rate by retarding diffusion of other elements. Thus Re contributes to the creep rupture life of single crystal superalloy significantly.

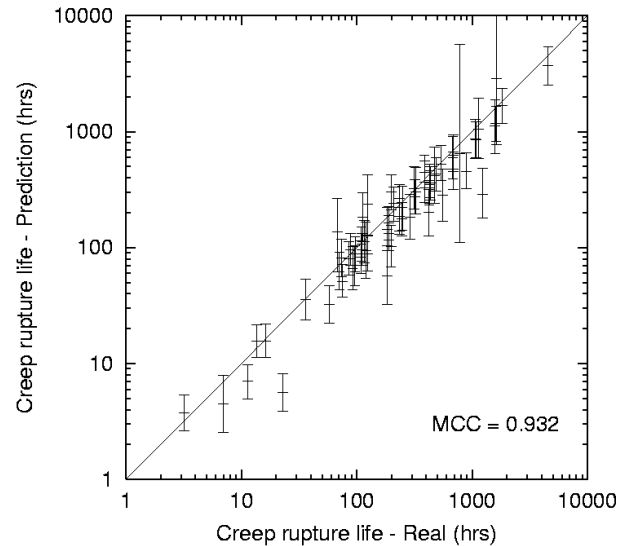


Figure 4. Real and predicted creep rupture life for reserved data.

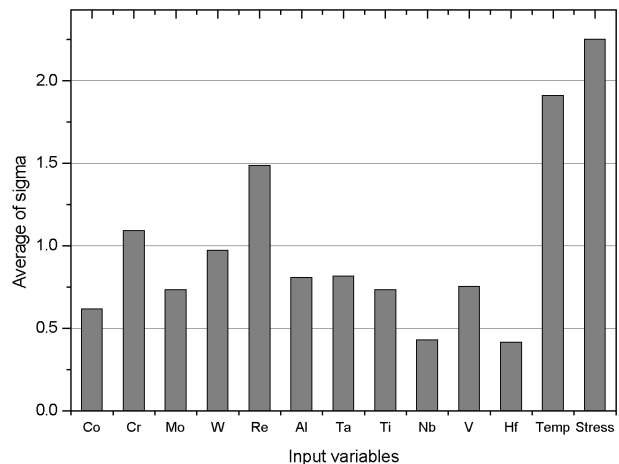


Figure 5. Effects of input variables obtained by Automatic Relevance Determination : Creep modeling.

The next dominant element was Cr. In general Cr negatively affects creep rupture life of superalloy. Nevertheless Cr is indispensable alloying element because it gives superalloy the resistance to oxidation and hot corrosion. We can also see the effects of solid solution hardening elements such as W and Ta and the γ' former such as Al and Ti are notable.

To see the effects of each elements keeping all other input variables constant the composition of CMSX-4, CMSX-10, Rene N5 were selected. The contents of each element were varied from 0 to 15 wt% and the creep stress and temperature were set to 210MPa and 950 °C, respectively. (In this variation Ni act as a buffer; so it is possible to make the composition of other elements to be constant.)

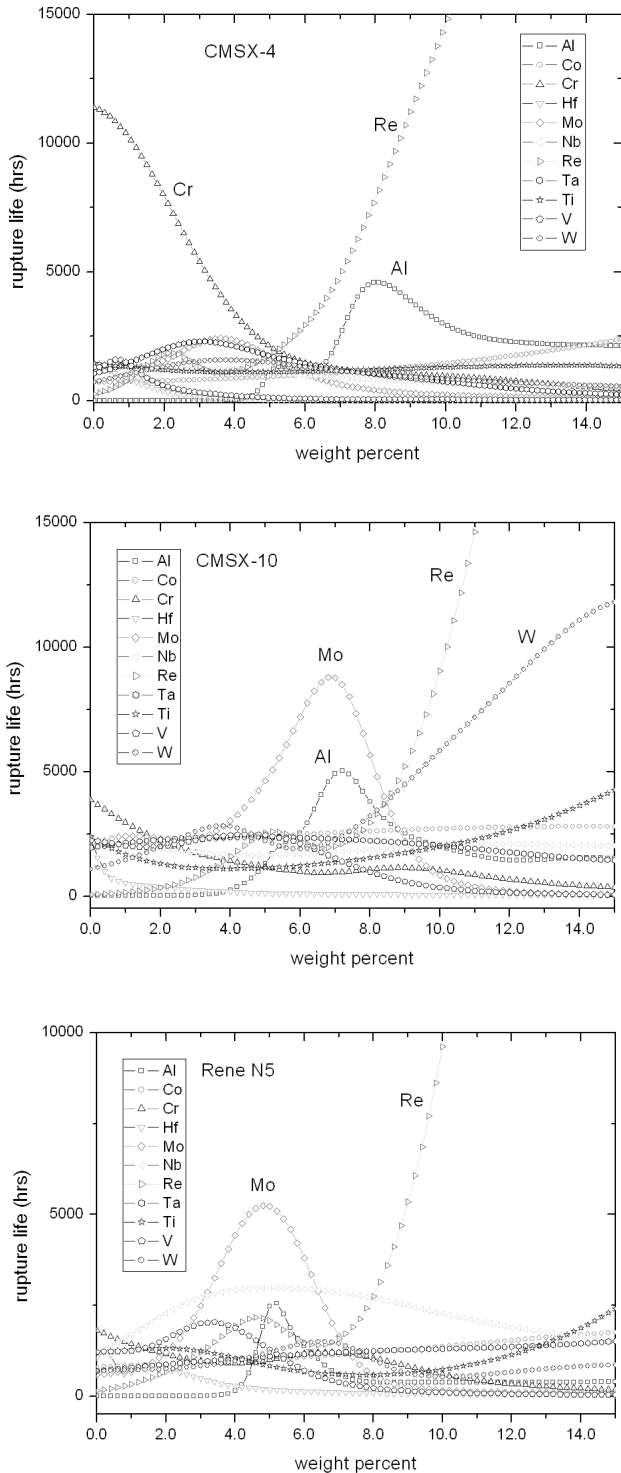


Figure 6. Effects of elements on creep rupture lives – based on CMSX-4, CMSX-10, Rene N5, respectively.

In Fig. 6 we can clearly see high positive effect of Re and high negative effect of Cr in CMSX-4 graph. The effect of Al was also high though it showed a maximum around 8.0 wt% of contents. In CMSX-10 graph we can see positive effect of W and maximum peak of Mo. Note that this kind of non-linear prediction is impossible in Multiple linear regression; MLR will only predict ever increasing or ever decreasing tendency in line. Also the effect of TCP formation for very high Re alloys is not considered in this model which is based on the range of Table I.

Neural network modeling for tensile properties

Similar procedures were applied to the modeling of yield strength (YS) and ultimate tensile strength (UTS) of single crystal superalloys. Total 229 YS datapoints and 231 UTS datapoints are collected from previous publications and Rolls-Royce database. Neural network trainings were performed for these databases varying the number of hidden nodes from 3 to 50. The optimum number of hidden nodes were determined to be 14 for YS modeling and 15 for UTS modeling. The real and predicted YS for 12-14-1 architecture is shown in Fig. 7 and that of UTS is shown in Fig. 8. Multiple correlation coefficients for YS and UTS were 0.988 and 0.977, respectively implying good agreements between real and experimental points.

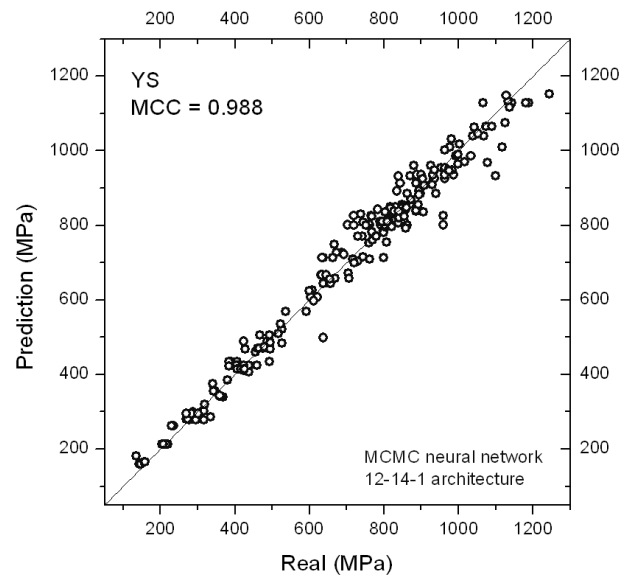


Figure 7. Experimental and predicted yield strength for 12-14-1 architecture

Fig. 9 and Fig. 10 shows ARD results for YS and UTS respectively. In each case, the effect of tensile test temperature was the highest as expected. Among the alloying elements the influence of Hf was the highest for YS and the influence of W was the highest for UTS. We can find the effect of W is higher than the effect of Mo in tensile properties.

Alloy Design

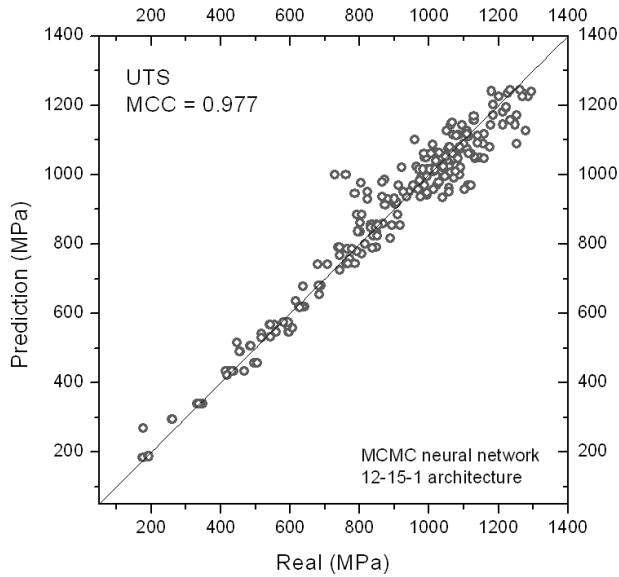


Figure 8. Experimental and predicted UTS for 12-15-1 architecture

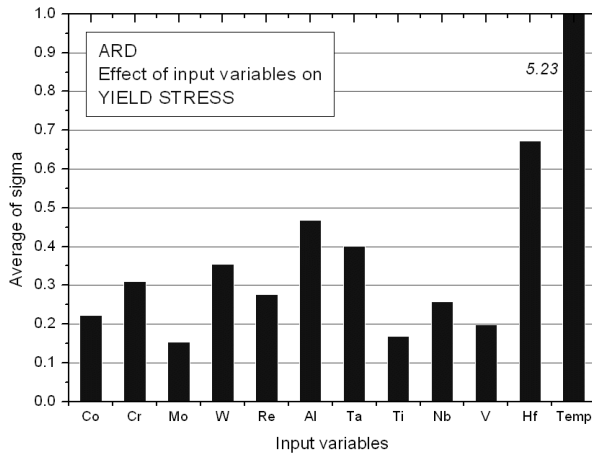


Figure 9. Effects of input variables obtained by Automatic Relevance Determination : YS modeling

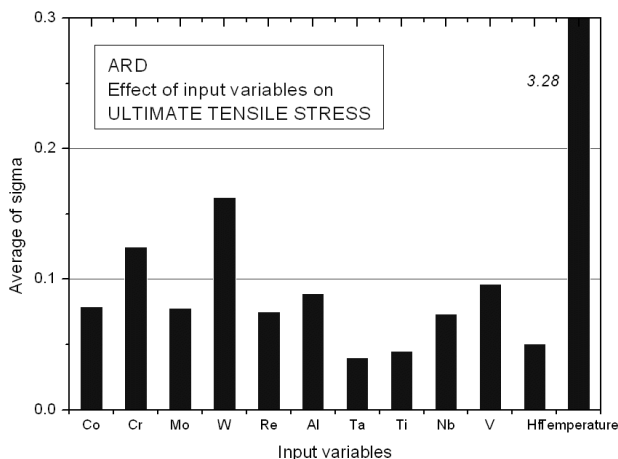


Figure 10. Effects of input variables obtained by Automatic Relevance Determination : UTS modeling

With a neural network prediction module for the creep rupture life, an alloy design system is constructed which can calculate volume fraction of γ' , Barrow's (σ phase formation) criteria, heat treatment window, etc. As a 1st stage, 112,896 compositions were calculated with combinations of rough steps. Calculated results were filtered with criteria: (i) volume fraction of γ' is between 0.62 and 0.68, (ii) Barrow's criteria is higher than 0.0 to ensure phase stability, (iii) heat treatment window is wider than 40 °C, (iv) $(W+Re)/Ta$ is lower than 1.25 to avoid freckle formation, (v) total refractory elements are more than 15 weight percent for enough solid solution strengthening. From the filtered results compositions with high creep rupture life at 950 °C were selected, and the 2nd stage of calculations were performed with 11,027,016 compositions having finer steps.

The final compositions selected are shown at Table III. They are the 2nd generation single crystal superalloys with 3 wt % of Re like CMSX-4 which are designated as K2S series alloys in this study. They contain high Co and low Cr that follow the trend of the 3rd generation single crystal alloys (especially CMSX-10). If K2S-1 is regarded as the base, K2S-2, K2S-3, and K2S-4 are designed to see the effects of W, Hf, and Ta, respectively. Mother ingots are produced by Ross & Catherall Ltd., UK. According to composition analysis results real alloys showed very close composition to those of target and amount of trace elements are well controlled.

Table III. Compositions of newly designed K2S alloys

	Co	Cr	Mo	W	Re	Al	Ta	Ti	Hf
K2S1	12.0	4.0	2.0	4.2	3.0	6.2	6.8	0.8	0.0
K2S2	12.0	4.0	2.0	4.6	3.0	6.2	6.8	0.8	0.0
K2S3	12.0	4.0	2.0	4.2	3.0	6.2	6.8	0.8	0.1
K2S4	12.0	4.0	2.0	4.2	3.0	6.2	7.4	0.8	0.0

Rod type single crystal specimens with 15mm diameter and 180mm length are prepared by directional solidification in an ALD DS/SX furnace with withdrawal rate of 3.8 mm/min. Homogenization heat treatments were performed as follows.

- o Heat to 1260 °C, hold 10 minutes
- o Heat to 1288 °C at 30 °C/hour
- o Heat to 1320 °C at 6 °C/hour
- o Hold for 6 hours followed by gas fan quench
- o Aged at 1140 °C for 2 hours followed by gas fan quench
- o Aged at 870 °C for 16 hours followed by air cooling

Fig. 11 is the microstructure after homogenization showing no coarse γ' with negligible interdendritic porosities. Trace of solidification dendrites can be seen which could not be fully removed due to local melting. Fig. 12 shows the γ/γ' microstructure after 1st ageing. Cuboidal γ' is aligned 3 dimensionally separated by thin \square channel which is typical and ideal single crystal γ/γ' microstructure.

Creep Tests

Fig. 13. shows creep curves for CMSX-4 and K2S alloys crept at 950 °C/355MPa and 982 °C/248MPa. For 950 °C tests creep lives of K2S alloys were about twice (K2S-1, K2S-3) or three times

(K2S-4) longer than that of CMSX-4 at this test condition. If we regard the composition of K2S-1 as the base, Hafnium addition (K2S-3) resulted in negligible effect on creep life whereas extra W addition (K2S-2) and extra Ta addition (K2S-4) resulted in significant effects which is in good accordance with common expectation. That is, additional refractory elements contributed to enhanced creep life via solid solution strengthening; grain boundary strengthener Hf contributed little in single crystal creep life. For 982 °C notable difference in creep life was not observed among the K2S alloys and CMSX-4 because our NN calculations is performed only at the 950 °C/355MPa condition.

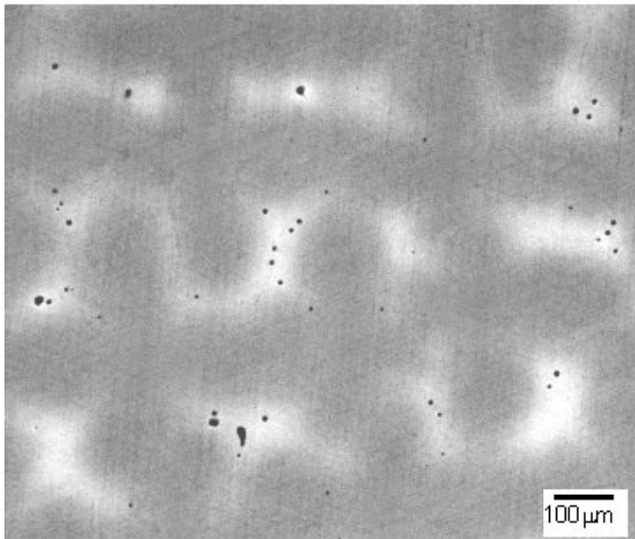


Figure 11. Solution heat treated microstructure of K2S-3 single crystal alloy.

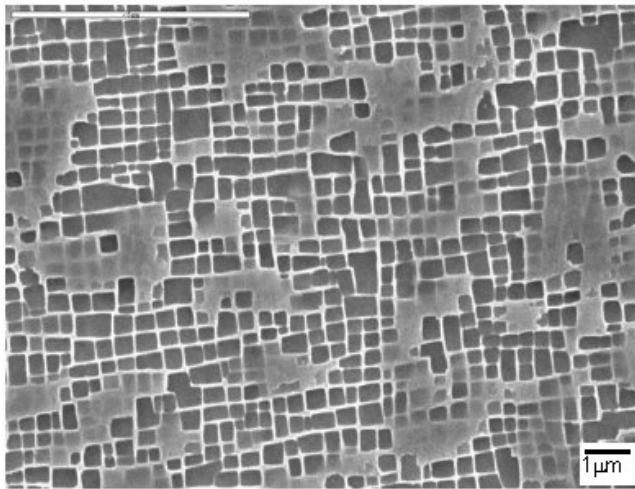


Figure 12. γ/γ' microstructure after solution and ageing treatment

Fracture surfaces are observed with SEM as shown in Fig. 7. Square areas containing center pores are aligned with identical orientations which is typical single crystal crept fracture surface.

Rafting microstructures are observed at the cross sections of crept K2S alloys as shown in Fig 8. The γ' (shown as white phase at the SEM photograph) elongated as plates perpendicular to the applied

stress axis (vertical direction in fig. 8) which is the favourable single crystal microstructure for creep properties. No topologically close packed (TCP) phases are observed at the specimens crept at 950 °C. Considering that single crystal superalloys are very sensitive in TCP phase formation around this temperature, K2S alloys seem to have good phase stability.

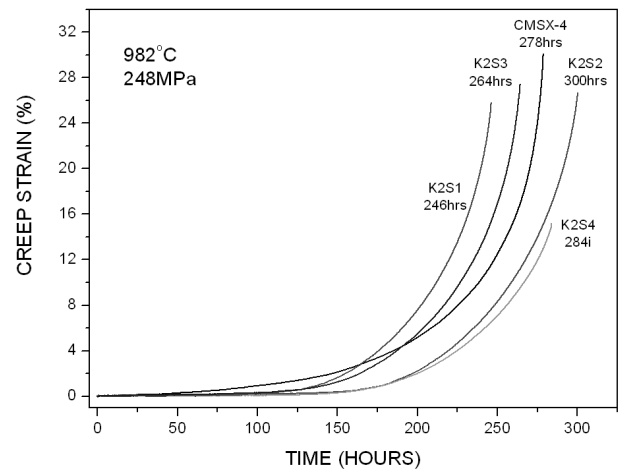
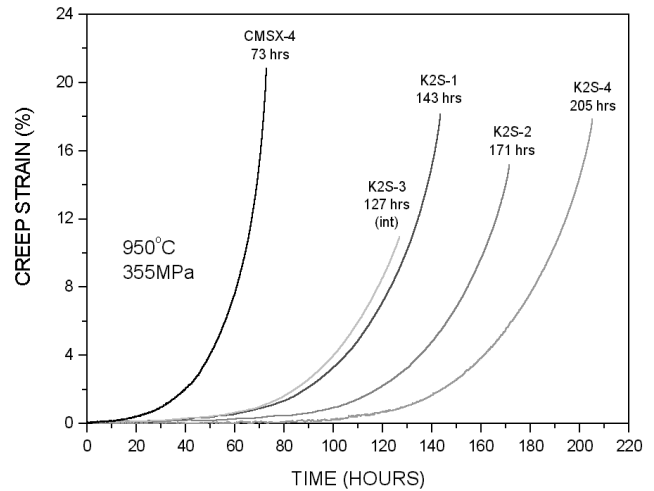


Figure 13. Creep curve for CMSX-4 and K2S alloys crept at the condition of 950 °C/355MPa and 982 °C/248MPa

Fig. 16. shows TEM microstructure of CMSX-4 crept at the condition of 950 °C/355MPa. In this rafted microstructure well developed dislocation network can be observed in γ' as well as in γ matrix. On the contrary this kind of network is not observed in K2S-4 crept at the same condition as can be seen in Fig. 17. We can observe some part of γ' is sheared by dislocations but network structure is observed only in γ matrix in Fig. 17. Difference in creep life between two alloys at 950 °C/355MPa might be explained by the difference of dislocation microstructure.

Conclusions

1. Mechanical properties of single crystal superalloys was modelled by Bayesian neural network with Markov chain Monte Carlo methods that showed good prediction performance. A test

of the model was performed with the reserved data and the multiple correlation coefficient between prediction and real creep life was 0.932.

2. The number of inputs was relatively small which was possible because of simple microstructure of single crystal superalloy.

3. As well as alloy composition, the model contains test conditions such as creep stress and temperature as input. It can be more versatile model compared with a model which can predict creep properties for only one condition.

4. The automatic relevance determination (ARD) technique revealed the influence of alloying elements; the greatest influence to creep life was by Re and Cr which is in accordance with reality.

5. Using neural network model new single crystal superalloys are developed which exhibited excellent phase stability and creep rupture lives better than or equivalent to those of CMSX-4

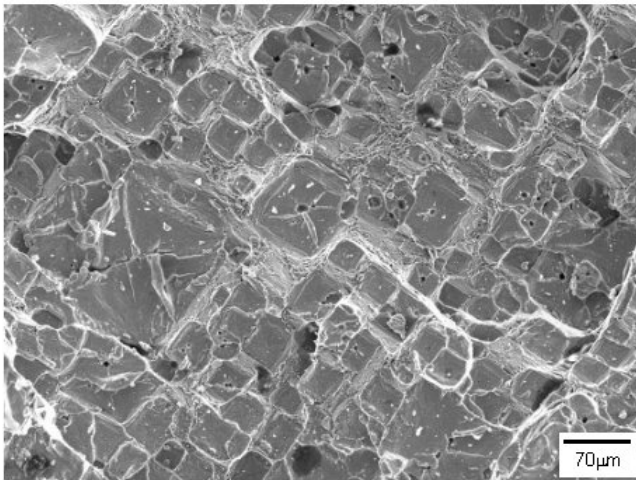


Figure 14. Fracture surface of K2S-1 alloy crept at the condition of 950 °C/355Mpa

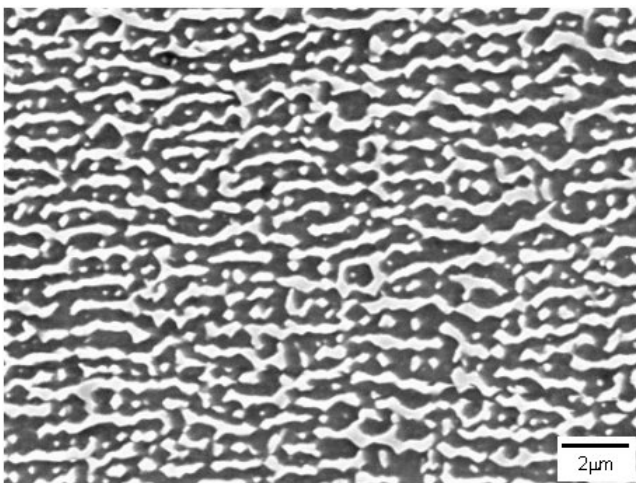


Figure 15. Cross section microstructure of K2S-1 alloy crept at the condition of 950 °C/355Mpa

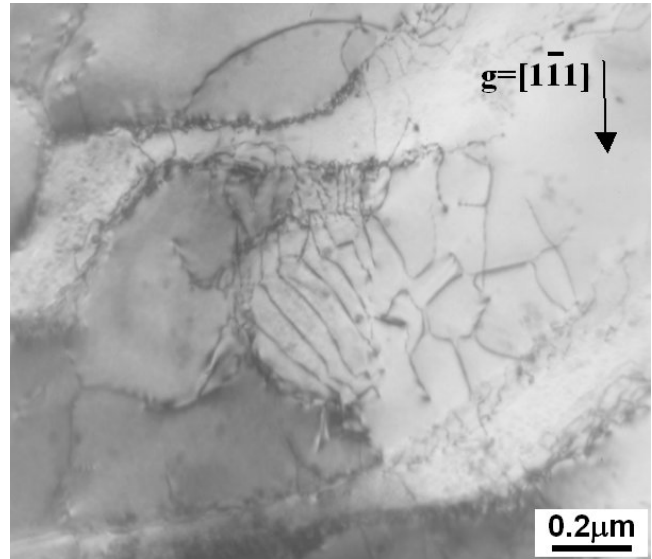


Figure 16. TEM microstructure of rafted CMSX-4 crept at the condition of 950 °C/355Mpa

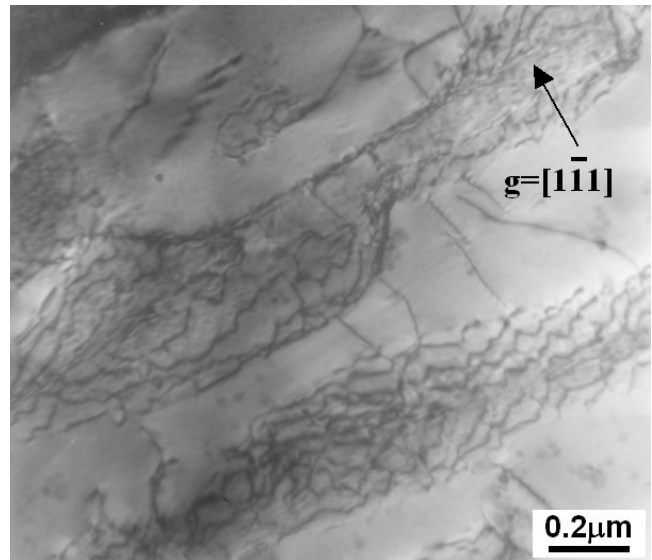


Figure 17. TEM microstructure of rafted K2S-4 crept at the condition of 950 °C/355Mpa

Acknowledgements

The authors would like to thank Professor Radford Neal at the University of Toronto for many helpful comments on this manuscript. Part of this work was supported by the Korea UK Research Centre.

References

1. H. Harada and M. Yamazaki, Tetsu to Hagane, 7 (1979) 337.

2. H. Harada, K. Ohno, T. Yamagata, T. Yokokawa, and M. Yamazaki, *Superalloys 1988*, Proceedings of the Fourth International Symposium on Superalloys, Seven Springs (1988) 733.
3. T. Hino, T. Kobayashi, Y. Koizumi, H. Harada and T. Yamagata, *Superalloys 2000*, Ed. by T.M. Pollock et al., TMS (2000) 729.
4. D.N. Duhl, *Superalloys, supercomposites and superceramics*, ed. by J.K. Tien and T. Caulfield, Academic Press (1989) 161
5. A. Mukherjee, S. Schmauder, M. Ruhle, *Acta Metall. Mater.*, 43 (1995) 4083.
6. J. Cai, R.A. Cottis, S.B. Lyon, *Corrosion Science* 41 (1999) 2001.
7. D.J.C. Mackay, *Neural Computation* 4 (1992) 415.
8. D.J.C. Mackay, *Neural Computation* 4 (1992) 448.
9. R.M. Neal, Bayesian training of backpropagation networks by the hybrid Monte Carlo method, Technical Report CRG-TR-91-1, Dept. Computer Sci., University of Toronto, Canada, (1992)
10. R.M. Neal, *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics No. 118, New York : Springer (1996)
11. J.M. Schooling and P.A.S. Reed, *Superalloy 1996*, ed. by R.D. Kissinger, et al., TMS (1996) 409.
12. J. Jones and D.J.C. MacKay, *Superalloy 1996*, ed. by R.D. Kissinger, et al., TMS (1996) 417.
13. H.K.D.H. Bhadeshia, D.J.C. MacKay, and L.E. Svensson, *Materials Science and Technology* 11 (1995) 1046.
14. H. Fujii, D.J.C. MacKay and H.K.D.H. Bhadeshia, *ISIJ International*, 36 (1996) 1373.
15. D.J.C. MacKay, *Mathematical Modelling of Weld Phenomena III*, eds H. Cerjak and H.K.D.H. Bhadeshia, The Institute of Materials, London (1997)
16. H. Fujii, D.J.C. MacKay, H.K.D.H. Bhadeshia, H. Harada and K. Nogi, *Journal of the Japan Institute of Metals*, 63 (1999) 905.
17. D.E. Rumelhart, G.E. Hinton, and R.J. Williams, *Nature*, 323 (1986) 533.
18. R.P.Lippmann, *IEEE Acoustics, Speech and Signal Processing Magazine* 4(1987) 4.
19. W.L. Buntine and A.S.Weigend, *Complex systems*, 5 (1991) 603.
20. D.J.C. MacKay, *ASHRAE Transactions*, 100 (1994) 1053.
21. R.M. Neal, *Neural Networks and Machine Learning*, ed. by C.M. Bishop, Springer-Verlag (1998) 97.
22. Personal communication with R. M. Neal
23. <http://www.cs.utoronto.ca/~radford/>