

ANN Based Prediction of Ultimate Tensile Strength Factors

P Venkatesh Murthy¹, Gautam Suresh², S Pranav³, Anandan S M⁴, Rohit Kunnath⁵

Assistant Professor Gr. III, Dept. of Mechanical Engineering, NMAMIT, Nitte, India¹

Student, Dept. of Mechanical Engineering, NMAMIT, Nitte, India^{2,3,4,5}

ABSTRACT: Ultimate tensile strength (UTS) is the resistance a material offers until it ruptures. The ultimate way to examine this mechanical property is to break the specimen which incurs a subsequent cost increment. Modelling this process by an expert and properly-trained person against a machine-learning algorithm to check the value of UTS helps in cost decrement consequently. Previous researchers have presented outstanding results with a Bayesian-network-based process. We have used an ANN and K-nearest-neighbour algorithm for achieving the same objective. We compare the obtained results and we have come to a conclusion that artificial neural networks is the best suited for this purpose than the rest of counterparts for the prediction of UTS.

KEYWORDS: Ultimate tensile strength, machine-learning algorithm, K-nearest-neighbour algorithm, Artificial neural networks.

I. INTRODUCTION

Foundry is the creative activity that has evolved to become a strong controlling factor in industry that maintains the society as we know it. Foundry supplies important parts to automotive, naval, aeronautic and weapon industries. These parts often play an important role in more complex systems, like brake drums, wind castings or aerospace structures. Hence, these foundry processes have very strict safety controls in order to ensure the quality of the manufactured products. Therefore one may think, the tiniest error will cause fatal accidents. Presently the often used techniques for the assurance of defect-free foundry processes are rigorous production control and complex simulation techniques. They are highly expensive and only obtain excellent result at a very high cost increment in producing it. Therefore, in order to provide effective cost saving method which can help to increase the quality standards and to save resources in the process (i.e. saving money). Mechanical properties that are experimented on in this project are ability of the material of a part to take several forces and tensions. In this paper we focus on the so-called *ultimate tensile strength* that is the resistance offered to the force until it breaks. Hence, the castings that are manufactured have to obtain a certain optimum value or threshold of ultimate tensile strength in order to pass the strict quality tests. Further the, current standard procedures to determine that values are by the use of destructive inspections. Unfortunately, these procedures make the piece worthless after the inspection and this adds to the cost increment.

In the previous papers they have discussed about a mechanical properties prediction system based on a Bayesian network process. After a training period, the Bayesian network learned the behaviour of the model and, hence it was able to predict its outcome [2] (i.e. the value of ultimate tensile strength). This showed that algorithms can also help in enhancing foundry production techniques. So also, similar *machine-learning* classifiers have been applied in other domains with outstanding results, for example, neural networks in this way, successful applications of artificial neural networks include for instance spam filtering [5], intrusion detection [6], or industrial fault diagnosis.

II. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANN) are almost similar to working of a human brain. ANN simulates the behaviour of neurons in the human brain [3]. Previously, a neuronal network consists on interconnected neurons. Similar to the,

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

activation of a neuron it depends on its set of inputs, where y_i is the activation of the current neuron, f_i is the activation function, $W_{j,i}$ is the weightage of the neuron and a_j is the activation of the input neuron (shown in equation 3).

$$y_i = f_i \left(\sum_{j=0}^n W_{j,i} \cdot a_j \right) \tag{3}$$

More accurately, *Multilayer perceptron* (MLP) is a kind of artificial neural network model of simple neurons called perceptron's that are structured in complex layers. The complex layers can be classified as *input/first layer*, *hidden/middle layer* and *output/third layer*. We perform the training of the model using *back propagation algorithm* [27] that calculates the weights W_j of the activation function for each neuron.

ANNs are mathematical models constituted by several neurons, arranged in different layers (input, hidden and output), interconnected through a complex network. They solve a problem by means of learning rather than by specific programming based on well-defined rules. In a feed-forward ANN, each input node transmits a signal to each input neuron which after processing, passes on the results to all the neurons belonging to the hidden layer(s). The hidden layer neurons process such signals and then send their outputs to the output layer neurons which lastly, after processing these inputs, generate the output signals of the network. No connections exist among neurons belonging to the same layer. Although each neuron can have several inputs it gives only one output signal which depends on the input signals, the weights of connections, the threshold value and the activation functions.

TABLE 1

Details of the experimentation

N_m	T_m	N_c	\emptyset	t_m	U_{actual}
1	1365	1A	140	145	568
		1B	140	170	583
		1C	70	101	570
2	1327	2A	140	111	464
		2B	140	133	462
		2C	70	72	509
		2D	70	86	489
3	1338	3A	140	115	590
		3B	140	143	566
4	1345	4A	140	57	549
		4B	140	92	503
5	1342	5A	140	130	593
		5B	140	165	606
6	1292	6A	70	96	526
		7A	70	140	458
7	1280	7B	70	155	436
		8A	35	128	523
8	1327	8B	35	113	536
		9A	35	144	521
9	1352	9B	35	127	539
		10A	70	75	569
10	1373	10B	70	94	553

III. MECHANICAL PROPERTIES

Once we have understood this the phases shown in section 3. We can say that the resultant casting is subject to forces (loads). Engineers have to calculate the value of these forces and how the material deforms or breaks as a function of applied load, time or other conditions. Therefore, it is a very important theme knowing how mechanical properties affect to iron castings [35], since they directly affect the quality of the final piece. More accurately, the most important mechanical properties of foundry materials are the following ones [37]: 6 Igor Santos, Javier Nieves, Pablo G. Bringas and Yoseba K. Penya

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

- Strength: it is that property that resists deformation under load. There are many kinds of strength namely, ultimate strength and ultimate tensile strength.
- Hardness: it is that property that resists permanent indentation.
- Toughness: it is that property that allows the material to withstand shock and to be deformed without breaking. This property is a combination as strength and plasticity.
- Resilience: it is that property of a material to absorb energy as and when it is deformed elastically.
- Elasticity: it is that property of a material to return to its original shape after the force is removed.
- Plasticity: it is that property of a material to permanent deformation without breaking or rupturing. This property is the opposite of strength.
- Brittleness: it is the opposite of plasticity. A brittle metal breaks or shatters before it deforms. Generally, brittle metals have a high value in compressive strength but a low value in tensile strength.
- Ductility: it is the ability of a material that enables it to elongate, bend or twirl without cracking or breaking.
- Malleability: in comparison with ductility, it is that property that makes the material to deform by compressive forces without showing defects. A malleable material rolled into thin sheets.

An alternative method to predict mechanical properties in castings is based on the utilization of an artificial neural network (ANN). ANN has of many computational elements, operating in parallel, connected by links with variable weights which are suitably adapted during the learning process. The development of detailed mathematical models began in the 1960s; it is only in present days that improvements in the science of ANNs have allowed the development of manufacturing applications. In the utilization of ANNs, basic step consists of determining the input/output data necessary for the training stage. They have to be obtained either from a process model or through actual experimentation. Since analytical correlation between input and output variables in the previous study is that different set of ANN structures have been tested, and simple perception up to the multilayer perception with two hidden layers. For each ANN, different learning rates, iteration numbers, activation functions, and initial random weights have been considered. Correlating these results of this work, the next step in the research will be to extend the predictive capability of the ANN system to more complex shapes of industrial relevance.

ANNs mathematical models comprise of several neurons, arranged in different layers (input, hidden and output), and interconnected through a complex network. They solve a problem by means of learning rather than by specific programming based on well-defined rules. In a feed-forward ANN, each input node transmits a signal to each input neuron which after processing, passes on the results to all the neurons belonging to the hidden layer(s). The hidden layer neurons process such signals and then send their outputs to the output layer neurons which lastly, after processing these inputs, generate the output signals of the network. No connections exist among neurons belonging to the same layer. Although each neuron can have several inputs it gives only one output signal which depends on the input signals, the weights of connections, the threshold value and the activation functions.

The casting ladle was preheated to avoid temperature losses. Inoculation was performed by means of 0.35% of FeSi75. After inoculation the temperature was measured and part of the melt was poured into the chill mould for spectrographic analysis of the SG cast iron produced. The results of the analysis are summarized in Table 3 for the 10 melts produced

TABLE 2

Details of the charge materials

Charge materials	Percentage
Pig iron	25.8
Mild steel	21.5
Ductile iron scrap	38.7
Coke	9.0
CaCO ₃	2.5
Desulphurizing agents	2.4
Graphite	0.1

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

Table 3

1	Item No:	Diameter	Area	Breaking load	UTS	C	Si	Mn	P	S	Cr	Mo	Ni	Cu	Ti	V	Sn	Mg	Ceq:
2	13265	20	314	82	261	3.25	1.84	0.74	0.108	0.065	0.229	0.003	0.018	0.101	0.043	0.011	0.01	0.002	3.9
3	13321	20	314	72.8	232	3.24	1.72	0.76	0.097	0.062	0.305	0.003	0.024	0.102	0.039	0.011	0.012	0.002	3.85
4	19672	20	314	82	261	3.37	1.89	0.72	0.093	0.062	0.173	0.004	0.022	0.06	0.061	0.015	0.01	0.002	4.04
5	12582	20	314	84.8	270	3.29	1.88	0.83	0.118	0.057	0.284	0.004	0.02	0.1	0.051	0.013	0.013	0.002	3.96
6	13381	20	314	86	274	3.23	1.78	0.77	0.101	0.067	0.267	0.009	0.017	0.109	0.037	0.011	0.011	0.001	3.86
7	13293	20	314	69.2	220.38	3.28	1.89	0.84	0.105	0.06	0.261	0.004	0.019	0.108	0.043	0.011	0.013	0.002	3.95
8	13399	20	314	84.8	270.06	3.3	1.9	0.82	0.094	0.058	0.265	0.005	0.013	0.052	0.06	0.016	0.013	0.002	3.97
9	13124	20	314	84	267.51	3.69	2.63	0.216	0.072	0.012	0.033	0.002	0.006	0.011	0.047	0.008	0.01	0.048	4.59
10	63612	20	314	73.2	233.12	3.16	1.95	0.72	0.093	0.06	0.258	0.013	0.031	0.12	0.039	0.013	0.012	0	3.84
11	13533	20	314	85.6	272.61	3.23	1.61	0.6	0.119	0.064	0.29	0.021	0.038	0.183	0.038	0.013	0.015	0.002	3.8
12	13297	20	314	81.6	259.87	3.23	1.83	0.89	0.101	0.073	0.29	0.002	0.018	0.088	0.043	0.015	0.011	0.003	3.88
13	13551	20	314	80	273.83	3.25	1.9	0.69	0.113	0.06	0.324	0.022	0.04	0.197	0.031	0.012	0.016	0.003	3.92
14	13411	20	314	85.2	271.33	3.47	1.9	0.54	0.097	0.059	0.081	0.01	0.015	0.097	0.053	0.011	0.011	0.001	4.14
15	13409	20	314	82	261	3.48	1.92	0.58	0.101	0.059	0.111	0.006	0.018	0.073	0.06	0.012	0.012	0.001	4.15

The output X_i produced by the neuron i in the layer l is given by the following relationship:

$$X_i = f \left(W_{i,0} + \sum_{j=1}^n W_{i,j} X_j \right) \quad (1)$$

where f is the activation function, n the number of elements in the layer l , and $W_{i,j}$ the weight associated with the connection between the neuron i in the layer l and the neuron j in the layer $l-1$ whose output is X_j . Usually, the threshold value X_0 is constant and equal to 1 so that the corresponding weight $W_{i,0}$ (offset or bias) shifts the activation function along the abscissa axis. In supervised learning, a data set containing the input patterns and the corresponding output patterns is used to train the network. An iterative algorithm adjusts the weights of connections so that the responses y to the input patterns generated at output neurons, according to Eq. (1), are as close as possible to their respective desired responses d . This is achieved by minimizing the learning error, defined by the mean square error (MSE) $MSE = \frac{1}{QN_0} \sum_{m=1}^Q \sum_{n=1}^{N_0} [d_n(m) - y_n(m)]^2$

$$MSE = \frac{1}{QN_0} \sum_{m=1}^Q \sum_{n=1}^{N_0} [d_n(m) - y_n(m)]^2 \quad (2)$$

Where N_0 is the number of outputs and Q the number of training sets. Since the desired responses are known at the output level, the local error δy can be easily calculated for output neurons; conversely, the desired responses at the hidden layer level are unknown so that the local error for hidden layer neurons cannot be determined. This problem was overcome by the back-propagation algorithm [8]. It works by back-propagating the error signals from the output layer neurons to those of the hidden layer, with one pattern presentation at a time. At each presentation cycle K , a forward phase to determine the output errors is followed by a backward one to propagate the error signals among the hidden layer neurons. The weights of connections are adjusted using the following equation:

$$\Delta W_{i,j} = W_{i,j}(K + 1) - W_{i,j}(K) = -\eta D_j X_i \quad (3)$$

Where η is the learning rate, is the parameter controlling the stability and the rate of convergence, and D_j the derivative of the MSE. This, when j is an output neuron, is given

$$D_j = (d_n[m] - y_n[m]) f'(P_j) \quad (4)$$

Where $f'(P_j)$ is the derivative of the activation function, and when j is a hidden layer neuron and k an output neuron, it is

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

$$D_j = f'(P_j) \sum_{k=1}^{N_0} D_k W_{k,j} \tag{5}$$

The weights of connections are repeatedly adjusted until the least MSE is obtained. Back-propagation iteration is completed when Eq. (3) is applied to all of the neurons in the network; then the process starts again with a new input± output pattern presentation. Once the weights are adjusted, the performance of the trained network can be tested by applying input patterns not included in the training set. To this purpose, the generalization error, defined as the MSE between the outputs generated in response to an input not presented during training and the desired one, is used to quantify the predictive performance of neural networks.

IV. TRAINING OF NEURAL NETWORKS

Before training, the network architecture must be defined. As a general rule, the number of neurons must be large enough to form a map region that is as complex as necessary for a given problem. However, it must not be so large that many of the necessary weights of connections cannot be accurately estimated from the available training data.

Furthermore, a trained ANN is very effective only if high generalization performance is achieved. In the problem considered, 14 input neurons were used in order to predict the value of ultimate strength in the single output neuron. The meanings of the input neurons are given in Table 4. Normalization of the values was obtained by dividing each value of the training set by the maximum value a of the specific variable considered. Several feed-forward fully connected ANNs were investigated; activation functions, learning rates and initial random weights. Each network was trained to different number of iterations. Validation of the neural network the capability of ANNs to correctly generalize was checked using some input/output data not included in the training set. They have been chosen among

Table 4
Input neurons to the ANN

Neuron number	Process variable
1	Carbon (%)
2	Silicon (%)
3	Manganese (%)
4	Sulphur (%)
5	Phosphorus (%)
6	Copper (%)
7	Tin (%)
8	Nickel (%)
9	Molybdenum (%)
10	Magnesium (%)
11	Chromium (%)
12	Ladle temperature (°C)
13	Time to mould (s)
14	Module (volume/surface) of the casting (mm)

The entire experimental set on the basis of the following considerations:

1. Each input neuron variable of the validation set must be within the range defined by the entire training set for that variable.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

2. In the training set there is no sample of the same casting number and diameter, which thus differs only in the time to mould. The latter condition has been introduced for the purpose of avoiding undesirable good results induced by the presence of "similar" samples in the training and validation sets. As a result, the samples used for validation of all the investigated topologies are those indicated in Table 6. The validation set is thus mainly suited to verify the ability of the ANN to predict mechanical properties in castings of different melt composition. Nevertheless, it is still possible to verify the predictive capability of the network on the time to mould parameter, by considering the results of the samples which belong to the same melt namely, $4A \pm 4B$ and $9A \pm 9B$.

It is worth noting that the sign of strength variation with the time to mould parameter is not consistent throughout the experiments, but depends on the melt and on the diameter of the casting; thus, it is of particular relevance in the context of strength prediction. The possibility of directly verifying the network capability on castings of various diameters (or cooling rates) is not so easy because it is impossible to pour two moulds at the same time with the same ladle. Nevertheless, the importance of this parameter is such that good results will implicitly validate it.

V. CONCLUSION

It is worthwhile to note that this paper gives a basic insight to our research for finding ANN based prediction of ultimate tensile factor value for future use. We find that the cost increment can be reduced to a great extent.

REFERENCES

- [1] Igor Santos, Javier Nieves, Yoseba K. Peña and Pablo G. Bringas Faculty of Engineering (ESIDE) University of Deusto Bilbao "Towards Noise and Error Reduction on Foundry Data Gathering Processes"
- [2] Igor Santos, Javier Nieves, Yoseba K. Peña and Pablo G. Bringas S3Lab, Deusto Technology Foundation, Bilbao, Basque Country "Machine-learning-based Mechanical Properties Prediction in Foundry Production"
- [3] S. Calcaterra, G. Campanab, L. Tomesanib,* aSABIEM Foundries, Bologna, Italy bDepartment of Mechanical Construction Engineering, University of Bologna, Viale Risorgimento 2, Bologna 40136, Italy" Prediction of mechanical properties in spheroidal cast iron by neural networks"
- [4] V.V.Mane¹, Amit Sata² and M. Y. Khire³ "Data mining driven DMAIC framework for improving foundry quality – a case study New Approach to Casting Defects Classification and Analysis Supported by Simulation"
- [5] An-hui Cai a,*, Yong Zhoua, Jing-ying Tan a, Yun Luoa, Tie-lin Li a, Min Chenb, Wei-ke An a "Optimization of composition of heat-treated chromium white cast iron casting by phosphate graphite mold"
- [6] igor santos, javier nieves, pablo g. bringas and yoseba k. penya*" Machine-learning-based defect prediction in high-precision foundry production"
- [7]Jan Voracek "Prediction of mechanical properties of cast irons"
- [8]Igor Santos, Javier Nieves and Pablo Garcia Bringas" Collective Prediction of Ultimate Tensile Strength in High-precision Foundries"
- [9] Igor Santos, Javier Nieves, Carlos Laorden, Borja Sanz and Pablo Garcia Bringas "Collective Classification for the Prediction of Microshrinkages in Foundry Production"
- [10] Carlos Laorden, Xabier Ugarte-Pedrero, Igor Santos, Borja Sanz, Javier Nieves, Pablo G. Bringas" On the Study of Anomaly-based Spam Filtering Using Spam as Representation of Normality"