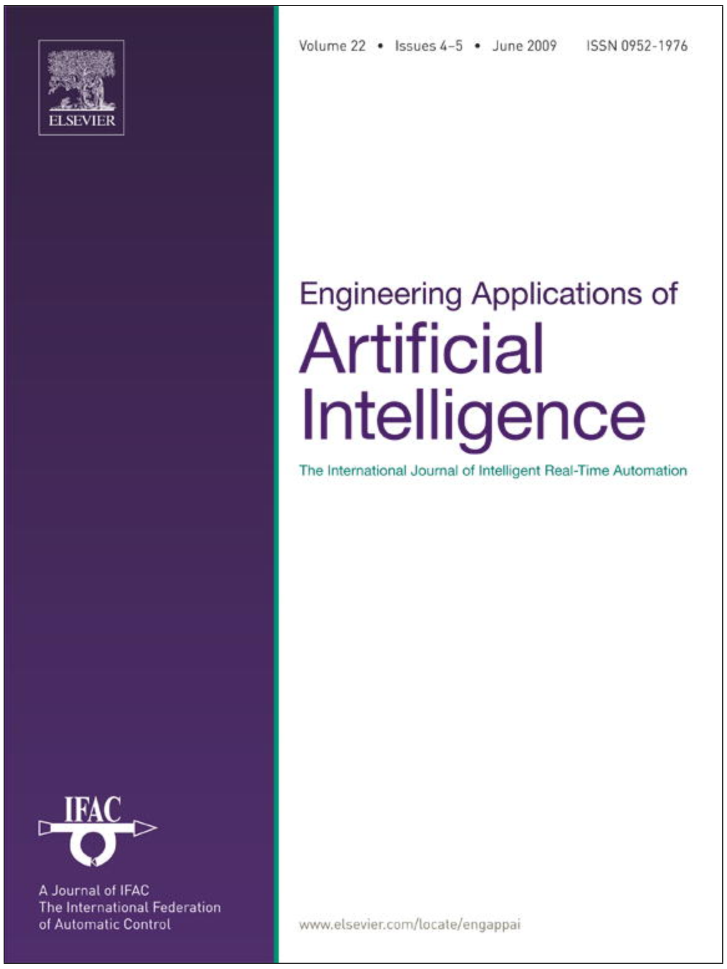
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.


Volume 22 • Issues 4–5 • June 2009 ISSN 0952-1976
ELSEVIER
Engineering Applications of Artificial Intelligence
The International Journal of Intelligent Real-Time Automation
IFAC
A Journal of IFAC The International Federation of Automatic Control
www.elsevier.com/locate/engappai

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Engineering Applications of Artificial Intelligence 22 (2009) 718–731

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Qualitative behavior rules for the cold rolling process extracted from trained ANN via the FCANN method

Luis E. Zárate *, Sérgio M. Dias [1]

Department of Computer Science, Applied Computational Intelligence Laboratory-LICAP, Pontifical Catholic University of Minas Gerais, Av. Dom José Gaspar 500, Coração Eucarístico, Belo Horizonte, 30535-610 MG, Brazil

A R T I C L E I N F O

Article history:
Received 24 January 2008
Received in revised form
13 October 2008
Accepted 30 November 2008
Available online 30 January 2009

Keywords:
Steel industry
Cold rolling process
Neural networks
Knowledge extraction
Formal concept analysis
Machine learning

A B S T R A C T

Nowadays, artificial neural networks (ANN) are being widely used in the representation of different systems and physics processes. In this paper, a neural representation of the cold rolling process will be considered. In general, once trained, the networks are capable of dealing with operational conditions not seen during the training process, keeping acceptable errors in their responses. However, humans cannot assimilate the knowledge kept by those networks, since such knowledge is implicit and difficult to be extracted. For this reason, the neural networks are considered a "black-box".

In this work, the FCANN method based on formal concept analysis (FCA) is being used in order to extract and represent knowledge from previously trained ANN. The new FCANN approach permits to obtain a non-redundant canonical base with minimum implications, which qualitatively describes the process. The approach can be used to understand the relationship among the process parameters through implication rules in different operational conditions on the load-curve of the cold rolling process. Metrics for evaluation of the rules extraction process are also proposed, which permit a better analysis of the results obtained.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

In the last years, the rolling process has been highly automated and reached a high level of sophistication. In this field, the objectives are always larger productivity and better quality of the final product. Considering the cold rolling process, the requirements to enhance the quality of products are typically thickness and shape. The extensive literature shows several strategies to reach these objectives, inclusive through non-conventional techniques as neural networks (Larkiola et al., 1996, 1998; Yang et al., 2004), fuzzy logic (Jung et al. 1995) and genetic programming (Son et al., 2004). The only condition for the new strategies is to represent correctly the behavior of the process in a quantitative and qualitative form for different operational conditions, so that these can be used in the design of on-line control and supervision systems.

The mathematical model of a process can be obtained through the study of variables involved in the process. In general, many physical models have non-linear characteristics and analytical complexities that avoid its application in on-line systems. In Montmitonnet (2006), a reviewing of models for the mechanical study of the cold strip rolling was presented. The author uses the classification of models in 1D (slab method), 2D/3D upper bound, finite difference and finite element methods. The 1D models, where velocity, strain and stress field depend only on the coordinate in the rolling direction, are based on the slab method (Bland and Ford, 1952; Alexander, 1972). Alexander's mathematical model (Alexander, 1972), which is considered within the slab method (1D model) as one of the most complete, will be considered in this work. As it is already known within the rolling theory and mentioned in Montmitonnet (2006), the 1D models are perfect for large L/h ratios (>3) (where L is the strip/work roll bite length and h is the output thickness). For $L/h<3$, the FEM models are more appropriate. In this work, the discussed case study has $L/h=5.61$, reason why the Alexander's model is considered. This model is non-linear and involves parameters through complex equations of difficult analytical solution. Moreover, that model is known by requesting numerical solution with significant computational effort. A more natural alternative to understand this kind of process is through symbolic representation, expressing the cause-effect relation among the parameters through qualitative rules (Cristea et al., 1997; Craven and Shavlik, 1999).

On the other hand, artificial intelligence (AI) techniques have been proposed as an alternative to represent knowledge of real world systems, without the necessity of a more detailed study

* Corresponding author. Tel.: +55 31 3319 4117; fax: +55 31 3319 4001.
E-mail addresses: zarate@pucminas.br (L.E. Zárate), sergiomariano@gmail.com (S.M. Dias).
URL: http://www.inf.pucminas.br/projetos/licap (L.E. Zárate).
[1] Tel.: +55 31 3319 4117; fax: +55 31 3319 4001.

0952-1976/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2008.11.009

Author's personal copy

Nomenclature

μ = friction coefficient
$\bar{Y}$ = average yield stress (N/mm^2)
g = gap (mm)
h_i = strip input thickness (mm)
h_o = strip output thickness (mm)
M = stiffness rolling mill modulus (N/mm)
P = rolling load per unit width (N/mm)
P_w = rolling load (N)
Tq = rolling torque (N mm/mm)
R = roll radius (mm)
t_f = front tension stress (N/mm^2)
t_b = back tension stress (N/mm^2)
W or = strip width (mm.)
E = Young modulus of the strip material
S_i = yield stress in plane-strain in the entry
S_o = yield stress in plane-strain in the exit

about the physical principles. AI proposes two main fields of representation: symbolism and connectionism. The symbolic field deals with symbolic or explicit-knowledge and the connectionist field, where artificial neural networks (ANN) are inserted, deals with implicit, numerical or sub-symbolic knowledge. Several researches have shown the capacity of the ANN to represent the most different physical systems. The application of ANN in several metallurgical processes can be seen in extensive literature (Andersen et al., 1992; Smartt, 1992; Aistleitner and Mattersdorfer, 1996; Zárate, 1998; Zárate et al., 1998; Gunasekera et al., 1998; Zárate and Bittencout, 2001, 2002; Shlang et al., 2001; Kim et al., 2002; Gálvez et al., 2003; Yang et al., 2004, Son et al., 2004, 2005, among others). The neural networks (NN) were proposed as an efficient model to predict mechanical properties (Myllykoski et al., 1996; Lenard and Zhang, 1997; Kim et al., 2008), rolling load (Larkiola et al., 1996; Gunasekera et al., 1998; Yang et al., 2004; Son et al., 2005 and Zárate and Bittencout, 2008) and to optimize the operation of rolling process.

In Petty (1996), an overview of the use of modeling and simulation based on computational techniques as finite element methods, finite difference applications, neural networks and expert systems were discussed. The author indicated the importance of the hybrid design of intelligent optimization systems based on experiments theory, physical models and neural networks. In Gams et al. (1997), a global integration structure, which allows the integration of an arbitrary number of different systems as input–output data acquisition, knowledge-acquisition, machine learning, statistical systems and artificial intelligence systems (neural networks and expert systems) into one integrated system for control of sendzimir rolling mill, was presented. The structure also enables the transformation of nontransparent implicit-knowledge (neural network) into a transparent explicit-knowledge based on decision tree. In Xiaoguang et al. (1999), a synergetic artificial intelligence system to scheduling in finishing train of hot strip mills was discussed. The authors establish that fuzzy theory and expert system can simulate logical inference ability in left-half-brain of human being, and neural networks can simulate thinking of image in right-half-brain of human being.

Different hybrid structures for rolling process control have been proposed in the literature. For example, in Jung et al. (1995), a combination of fuzzy logic and neural networks was used in controlling the strip shape in cold rolling. The authors developed a fuzzy control algorithm based on production data and operational knowledge. In Wang and Frayman (2002), a dynamically generated fuzzy neural network applied to torsion vibration control of tandem cold rolling mill spindles was discussed. The proposed structure does not need to specify initial network architecture and fuzzy rules, constantly combined and pruned, are used to minimize the size of the network. For this structure, irrelevant inputs are detected and deleted. In Son et al. (2004), genetic algorithm to select the optimal architecture of a neural network applied in the hot rolling process was presented. In Son et al. (2005), an on-line learning neural network was developed in order to improve the prediction of the rolling load in hot rolling mill. With the new structure, the thickness error of the strip was considerably reduced. In Thangavel et al. (2006, 2007), a hybrid controller for control of strip tension and looper height in tandem mill was proposed. In these works, genetic algorithms are used to search optimal fuzzy rules and membership functions of a neuro-fuzzy system. Recently, in Zárate and Bittencout (2008) and Peng et al. (2008), control systems for the cold rolling process, based on ANN were presented. In Zárate and Bittencout (2008), the authors determine the sensitivity factors from the differentiating of previously trained neural network and these factors are used as parameters in the proposed control system. As it can be seen, several are the applications of ANN in the steel industry, which motivate the research about the extraction and handling of the knowledge learned by them.

In Mitra and Hayashi (2000), a survey about neuro-fuzzy rule generation was presented. The aim is the fuzzy rule generation from neural networks to make comprehensible, the knowledge embedded within the network. The authors showed that a learning process can be part of the knowledge-acquisition and, in the absence of an expert, sufficient time or data the reinforcement of learning can be resorted instead of supervised learning. If one has knowledge expressed through linguistic rules, a fuzzy system can be built. On the other hand, if one has data or can learn from a simulation or the real task, ANN is more appropriate. In reference to Hayashi and Buckley (1994), cited by Mitra and Hayashi (2000), it was proved that any rule-based fuzzy system may be approximated by a neural network and that any neural network feedforward and multilayer may be approximated by rule-based fuzzy system, which demonstrates the viability of the usage of both techniques. As it can be observed, the FCANN method considered in this work (Zárate et al., 2008) extracts the knowledge learned by the NN about the process through an implication rules minimal base, i.e., a minimum rules set. If a neural network hypothetically learns from representative data of the process, and presents a “zero” error in the training, all extracted knowledge will be an unconditional truth due to characteristics of this extracted rules set. In short, given a representative data set, a good training process and a satisfactory knowledge-acquisition of the network, the extracted rules do not need a forwarding refining as in the neuro-fuzzy rule generation, which demands the generation and refining of rules. This shows the relevance and contribution of the FCANN method.

Due to the amount of techniques for knowledge extraction from neural networks, a classification of them was necessary. In Andrews and Geva (2002), a new taxonomy to classify these techniques was introduced. From that taxonomy, it was possible to observe five primary classification criteria for different techniques: (a) the *expressive power* or *rule format* that corresponds to the rule extraction in various formats (propositional rules, fuzzy rules, scientific laws and others); (b) the *translucency* that permits to reveal the relationship between the extracted rules and the internal architecture of the trained neural network; (c) the *portability* of the rule extraction technique for different NN architectures; (d) the *quality* of the extracted rules; and (e) the

Author's personal copy

algorithmic complexity of techniques for rule refinement. The second proposed dimension, *translucency,* has two categories involving the *decompositional* and *pedagogical* techniques. The methods in the pedagogical category consider the trained neural network as a "black-box". The pedagogical techniques map inputs directly to output, generating examples that can be used for the symbolic learning algorithms. The FCANN method corresponds to the category of the pedagogical techniques.

Through the reviewing in the literature, it was possible to observe that the typical neural topology considered for different contributions corresponds to a totally connected feedforward multilayer perceptron. This is due to the great capacity that this type of neural network has to relate variables from data of the process and, due to their generalization capacity and low computational time that can be reached when in operation, what makes its usage attractive to the industrial sector. These networks can be used when the mathematical dominant model of a process is difficult to be obtained. NN can be trained through a data set measured directly from the real process, diminishing the time spent to obtain an analytical representation of the process from physics laws.

ANN have a mathematical function that relates the input and output parameters of a system from data. However, as mentioned before, that characteristic makes ANN a "black-box" because no explanation can be attributed to them for decision-making when in operation. ANN relates values from the training sets–provided to them–and informs their responses, but information about the generation principles is left aside. Thus, there is some implicit-knowledge hard to extract, which, if extracted, can reinforce an intelligent machine in its mechanism to explain its conclusions. The extracted knowledge can be expressed through symbolic rules, allowing the knowledge of the problem into its real context. In this work, the FCANN method to extract and represent knowledge from previously trained ANN based on formal concept analysis (FCA) will be applied to neural representation of the cold rolling process. In Zárate et al. (2008), the FCANN method was compared with conventional techniques of knowledge extraction such as TREPAN and C4.5 algorithms. Those comparisons showed the method relevance to extract qualitative knowledge of the process through implication rules of the type "*if*…*then*…", that are easy to understand. The FCANN method extracts the qualitative relations learned by the network, independently of its input–output structure, while conventional techniques aim to extract the knowledge working simply as a classifier. Moreover, the FCANN method revealed capable to represent different types of processes learned by ANN such as solar energy system, climatic behavior and evaluation of urban real estate, among others.

The experience shows that for industrial applications, any new approach, technique or method to extract knowledge from previously trained ANN must look for a *symbolic representation* of *easy understanding*, moreover being *independent of the structure* of the network considered. The FCANN approach gathers these characteristics and permits to change the detail level of the behavior rules "*if*…*then*…" generated, which can describe the process in study without requiring a new training process better.

On the other hand, considering the cold rolling process, the equation that governs a single stand rolling mill is a non-linear function on several parameters: input thickness, front and back tensions, average yield stress and friction coefficient. Any alterations on either of them will cause alterations on the rolling load and, consequently, on the outgoing thickness. In this work, the main objective is to represent through qualitative rules, extracted from previously trained neural network, the behavior of the cold rolling process via the FCANN method. This method permits to map the behavior of the process for different operational conditions on the load-curve. One motivation for this research is to capture knowledge about physical systems via qualitative rules to provide the understanding of the process by less-experienced operators. Moreover, it provides information about the analyzed process that can be used in the design of on-line control and supervision systems.

This work is about the neural representation of the cold rolling process and the extraction of knowledge from neural networks using the FCANN method. Metrics that evaluate the quality of the extracted knowledge are also presented. Thus, this paper discusses, in a generic form, the adopted procedures that can be extended for other industrial processes.

This paper is organized in seven sections. In the second one, a revision of the cold rolling process is presented. In the third one, neural representation of the process, based on Alexander's model, is discussed. In the fourth one, the FCANN theoretical fundaments are presented. In the fifth one, the extraction of knowledge applied for the cold rolling process is showed. Finally, the contributions and the conclusions of this work are presented.

2. Cold rolling process

Into classical theories (1D model), there are several models to calculate the rolling load necessary for the deformation process. These models, Eq. (1), are non-linear functions of several parameters, where it is possible to observe that any change in the input thickness (h_i), in the output thickness (h_o), in the back tension (t_b), in the front tension (t_f), in the yield stress (Y) and/or friction coefficient (μ), will cause alterations on the rolling load (P) and, consequently, on the outgoing thickness.

$$P = f(h_i, h_o, t_b, t_f, \mu, Y, E, R) \tag{1}$$

where (E) is the Young's modulus of the strip material and (R) is the roll radius.

During the rolling process, the cylinders are compressed against the strip by a force transmitted by the back rolls. Since the rolling mill is not perfectly rigid, the output thickness can be expressed by the elastic equation of the rolling mill, Eq. (2).

$$h_o = g + \frac{P \cdot W}{M} \tag{2}$$

where (g) being the roll-gap or "gap", (W) the width of the material being rolled, *(M)* the mill modulus and (P) rolling load per unit width.

In order to illustrate the behavior of a rolling mill, Fig. 1 (Dieter, 1976) shows variations in the nominal operational condition due to the variation in several parameters. Curve (I) corresponds to an increment in the back and/or front tensions of stress. Curve (II) represents a decrement in the input thickness; curve (III) corresponds to the operational condition; curve (IV) represents an increment in the output thickness and curve (V) shows possible increment in the friction coefficient and in the average yield stress or a decrement in the back and/or front tensions.

The characteristic behavior of the process just mentioned will be considered in the next sections in order to analyze quantitatively the neural representation and qualitatively, through symbolic rules, the cold rolling process.

3. Neural representation of the cold rolling process

As mentioned, in the last years, artificial neural networks are being proposed as powerful computational tools due to the low computational time of processing that can be reached when the network is in operation. These times can be of the order of some

Author's personal copy

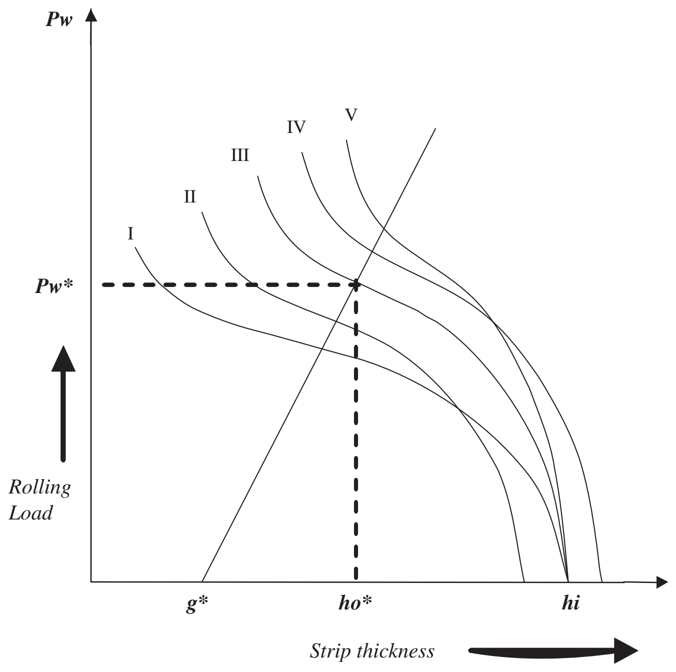


Fig. 1. Operational condition of a rolling mill.

hundreds μs. or minors and this makes the neural network attractive for design of on-line control and supervision systems.

In this section, a representation of the cold rolling process, Eq. (1), through ANN, will be considered. In this case, the Alexander's model (Alexander, 1972) was used to generate a database and it will be used for neural network training and validation. The representation will consider the cylinder-material deformation process.

3.1. *Neural representation of the cylinder-material deformation process*

The set of parameters to represent the rolling process via ANN is defined by Eq. (3).

$$(h_i, h_o, \mu, t_b, t_f, \bar{Y}) \xrightarrow{\text{Neural Network}} (P) \tag{3}$$

The type of network considered in this work corresponds to a multilayer perceptron, which approaches the cognitive models that try to describe the operation of the human brain. The type of learning applied is the supervised learning and the training algorithm is the "back-propagation". The neural network has tree layers: input layer (without neurons), layer of hidden neurons (which receives the external inputs) and layer of output (responsible by the generation of the output of the NN, Haykin, 1999). The definition of the net structure, such as the number of hidden layers and the number of neurons in those layers, is still a problem without definitive solution; although there are some approaches about it. In this work, the number of neurons for the hidden layer is suggested as 2N+1 neurons, where N is the number of inputs of the net, as suggested in Hecht-Nielsen (1990) and Kovács (1996). The practical experience shows that the rule (2N+1) is adequate for ANN with few inputs (around 20) as that the data set is well conditioned. For many inputs the rule is not true. The definition of the NN structure is an art that demands the designer's experience. When increasing the neurons quantity in the hidden layer, it increases the non-linear association capacity of the NN. However, increasing this quantity excessively can lead to an over-adjustment of the data, inclusive of noise, if this noise is present in the training sets (*overfitting*). On the other hand, ANN with few neurons in the hidden layer could not map the whole wished data-space (*underfitting*).

The neural network considered has 6 inputs, where the number of neurons in the hidden layer was chosen as 13 (2N+1). The number of neurons in the output layer was chosen as 1, corresponding to the number of outputs of the network. The log-sigmod function was chosen as the axon transfer function, since it is more consistent with the biophysics of the biological neuron.

Generally, the largest care to get a trained neural network lies on collecting and pre-processing the data. The pre-processing procedure consists in the data normalization in such a way that the inputs and outputs values will be within the range of 0–1.

The following procedure was adopted to normalize the input data before using it in the neural network structure:

(a) In order to improve the convergence of the network training process, the normalization interval [0, 1] was reduced to [0.2, 0.8], because in the log-sigmod function, the values [0, 1] are not reached: $f \to 0$ for $net \to -\infty$ and $f \to 1$ for $net \to +\infty$. Moreover, this can give the network an extrapolation capacity. As the logarithmic scale compacts large data values more than smaller values, when the neural network contains nodes with sigmoidal activation function, better results can be achieved if the data are normalized within the interval [0.2, 0.8] (Tarca and Cooke, 2005; Altincay and Demirekler, 2002).

(b) The data were normalized through the following formula:

$$f^a(L_o) = L_n = (L_o - L_{\min})/(L_{\max} - L_{\min}) \tag{4a}$$

$$f^b(L_n) = L_o = L_n^* L_{\max} + (1 - L_n)^* L_{\min} \tag{4b}$$

where L_n is the normalized value, L_o is the value to normalize, $L_{\min}$ and $L_{\max}$ are minimum and maximum values of the parameters of the process, respectively.

(c) $L_{\min}$ and $L_{\max}$ were computed as follows:

$$L_{\min} = (4 \times LimiteInf. - LimiteSup)/3 \tag{5a}$$

$$L_{\max} = (LimiteInf. - 0.8 \times L_{\min}/0.2) \tag{5b}$$

Eqs. (5a) and (5b) are obtained substituting in the Eq. (4a) $L_n = 0.2$ and $L_o = LimiteInf$; and $Ln = 0.8$ and $Lo = LimiteSup$. Where *LimiteInf* and *LimiteSup* are the minimum and maximum values of the original data sets, respectively.

4. Theoretical fundaments for the FCANN method

4.1. *Formal concept analysis—short review*

In FCA, formal context is a primordial definition and it can be represented through a cross table. In Table 1, an example of formal context for the "numerical domain" is represented. Such a table represents a structure that defines objects (numbers), attributes (characteristics) and a binary relation between them.

One of the most powerful aspects in FCA is its capacity to represent cross tables graphically. Fig. 2 is called the line diagram or Hasse diagram (Carpineto and Romano, 2004) and represents the example involving numbers, depicted in Table 1.

Such example has been considered here with the purpose of showing the potential of FCA. The usage of diagrams instead of tables is certainly more intuitive, especially for those who do not deeply comprehend the application domain. The mathematical concepts associated to FCA are presented as follows.

Author's personal copy

Table 1
Example of formal context.

Objects	Attributes		
	Even	Odd	Prime
1		x	
2	X		x
3		x	x
4	X		
5		x	x
6	X		
7		x	x
8	X		
9		x	

Objects (rows header) are numbers 1, 2, 3, 4, 5, 6, 7, 8 e 9. Attributes (columns header) are characteristics even, odd and prime.

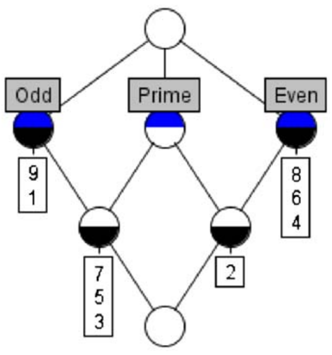


Fig. 2. Line diagram example.

4.1.1. *Formal context*

Formal contexts have the notation $K := (G, M, I)$, where G is a set of objects (rows headers), M is a set of attributes (columns headers) and I an incidence relation ($I \subseteq G \times M$). If an object $g \in G$ and an attribute $m \in M$ are in the relation I, this is represented by $(g, m) \in I$ or gIm and is read as "*the object g has the attribute m*".

Given a set of objects $A \subseteq G$ from a formal context $K := (G, M, I)$, it could be asked which attributes from M are common to all those objects. Similarly, it could be asked, for a set $B \subseteq M$, which the objects have the attributes from B. These questions define the derivation operators, which are formally defined as

$$A' := \{m \in M | gIm \, \forall g \in A\} \tag{6}$$

$$B' := \{g \in G | gIm \, \forall m \in B\} \tag{7}$$

A special case of derivate sets occurs when empty sets of objects or attributes are considered to be a derivate

$$A \subseteq G = \emptyset \square A' := M \quad B \subseteq M = \emptyset \square B' := G \tag{8}$$

4.1.2. *Formal concept*

Formal concepts are pairs (A, B), where $A \subseteq G$ (called extent) and $B \subseteq M$ (called intent). Each element of the extent (object) has all the elements of the intent (attributes) and, consequently, each element of the intent is an attribute of all objects of the extent. The set of all formal concepts in a formal context has the notation $B\ (G, M, I)$. Since a cross table representing a formal context is given, algorithms can be applied in order to determine its formal concepts and its line diagram (Ganter and Wille, 1996).

4.1.3. *Concept lattice*

When the set of all formal concepts of a formal context $K := (G, M, I)$ is hierarchically ordered according to the complete reticulate theory (Davey and Priestley, 1990; Gratzer, 1998), it is called conceptual reticulate with the notation $B\ (G, M, I)$. The formal concepts are related as $(A_1, B_1) \leqslant (A_2, B_2)$, when $A_1 \subseteq A_2$ and $B_2 \subseteq B_1$, being (A_1, B_1) called sub-concept and (A_2, B_2) called super-concept. For the reticulate shown in Fig. 2, the formal concept of extent {"2"} is, for example, sub-concept of the formal concept of extent {"4, 6, 8"}, with the attribute "Prime" distinguishing them.

4.1.4. *Many-valued context*

Even though this definition of formal contexts is valid for many situations, mainly to represent objects that have the presence or absence of some properties (attributes), it is not a good representation for the major part of situations, where objects have attributes that can take on several values. For this case, attributes are called many-valued attributes. So, the contexts, where the set M of attributes is composed by many-valued attributes, are called *many-valued contexts*. In this case, the notation of the formal context is given by the quadruple (G, M, V, I), where G is a set of objects; M is a set of many-valued attributes; V is the set of possible values by the attributes; and I is a ternary relation between G, M and V ($I \subseteq G \times M \times V$) (Ganter and Wille, 1996). In general, many-valued contexts can be transformed into single-valued contexts in order to obtain the formal concepts. A simple way to do such transformation is to replace each many-valued attribute by the corresponding attribute–value pair, as the example presented by Table 1. In Table 2, $T_j^{\,i}$ represents the pair attribute–value. Note that the number w_i of attribute–value pairs for each many-valued attribute is given by the cardinality of the set V (i.e. $|V|$). It is important to remember that attributes can have continuous values; for example, V may be equal to the set $\Re$ of real numbers. In this case, there will be as many attribute–value pairs as the cardinality of the set $\Re$, i.e. infinity. So, for this case, it may be interesting to choose intervals of values for attributes. This idea of discretization will be used by the FCANN method.

4.1.5. *Next closure algorithm*

The *Next Closure* algorithm was proposed in 1984 by Bernhard Ganter (Ganter, 2002) as an algorithm with a capacity to find *closure systems*, which is equivalent to the attainment of the formal concepts. The algorithm can be used to extract all formal concepts or to extract the minimum set of implication rules (*Steam-Base* or *Duquenne-Guigues base*) on a formal context (Guigues and Duquenne, 1986). The minimal implication base, which is non-redundant, provides a complete implication set, so that any valid implication on a formal context can be obtained through the combination of rules of the minimal base. The removal of any rule of the minimal base makes it an incomplete base. For more specific information about the *Next Closure* algorithm and the *Steam-Base* please refer to Guigues and Duquenne (1986), Ganter and Wille (1996) and Ganter (2002).

After the *Next Closure* algorithm is applied, it is possible to obtain an implication base £, with the following characteristics:

(1) Sound: each implication in £ is valid in $K := (G, M, I)$.
(2) Complete: each implication pertinent of $K := (G, M, I)$ is in £.

Table 2
Many-valued context.

Objects	Attribute # 1			…	Attribute # n		
	T_i^1	…	T_{w1}^1	…	T_i^n	…	$T_{wn}^{\ n}$
1	x			…			x
2			X	…	x		
…				…			

Author's personal copy

(3) Non-redundant: no implication in £ is originated from other implications of £.

And the base £ extracted by *Next Closure* algorithm has rules, among others, of the type

$$\text{if } x_i \in I_l^i \text{ and } x_j \in I_m^j \text{ and } x_k \in I_n^k \text{ then } y \in I_p \tag{9}$$

with:

X:	corresponds to a parameter or independent variable of the considered process.
I:	corresponds to a variation interval of the parameter.
Y:	corresponds to a parameter or dependent variable of the process. It normally corresponds to neural network output.

Note that the expression Eq. (9) corresponds to the rules without redundancy. Through these rules, it is possible to identify parameters and their intervals that determine the operational condition of the dependent variable. These rules can help the identification of the parameters that can change the operational condition, such as controllable variables, in a control system.

4.2. FCANN for knowledge extraction from ANN

In this section, the steps of the new approach to extract knowledge from neural networks, presented in Zárate et al. (2008), will be presented in a summary way. The approach allows the knowledge extraction and representation of physics processes in a stationary state, from previously trained ANN, which satisfies the requests of *symbolic representation* and *easy comprehension*. The structure of the neural network corresponds to a multilayer perceptron, feedforward, totally connected with N inputs and M outputs (in this work $M=1$). The output parameters (consequents) are variables to be analyzed from the input parameters (antecedents) through rules of the type *if*… *then*…. As follows, the steps of the FCANN method are presented.

Step 1. Select from the process a representative data set in order to train the ANN. It is defined as

$$X = [x_{ij}]m \times n \tag{10}$$

where: n is the number of parameters; X_{ij} to $i=1,\ldots,m$ and $j=1,\ldots,n-1$ are the input parameters and X_{in} to $i=1,\ldots,m$ is the output parameter. The output parameter should have a known probability distribution, such as a normal distribution $N_1(\bar{x}, S(x))$.

Step 2. Define the structure of the multilayer neural network (with N input parameters; H hidden layers and M output) and train it. In this work, $M=1$ for any situation.

Step 3. Build a synthetic database. To extract knowledge from the network, a synthetic database has been generated considering the domain ranges of the input parameters. This synthetic database will be used to operate and stimulate the trained neural network and to obtain the output parameter (explicit relation among parameters) trying to reveal the knowledge learnt by it. The database is defined as

$$Y = [y_{ij}]_{p\times n-1} \tag{11}$$

where: Y has only elements generated with the purpose of knowledge extraction; p represents the records number and n the parameters number. Thus, none of such elements has been collected from the process. Each input parameter has minimal and maximal values, which defines the domain range of each parameter. The minimum and maximum values are vectors that are, respectively, defined as

$$Inf = \{u_1, u_2, \ldots u_{n-1}\} \tag{12}$$

where $u_j = \min[x_{ij}]$, for $i=1,\ldots,m$

$$Sup = \{v_1, v_2, \ldots, v_{n-1}\} \tag{13}$$

where $v_j = \max[x_{ij}]$, for $i=1,\ldots,m$

The vector W defines the number of intervals that will be generated for each parameter, between the minimum and maximum values, Eqs. (12) and (13)

$$W = [w_j]; \quad j=1,\ldots,n-1 \tag{14}$$

Note that it is possible to consider $w_1 \neq w_2 \neq \ldots \neq w_{n-1}$

Hence, the variation interval of each parameter, which composes the synthetic data set, is represented by the following expression

$$Int = \{I_1, I_2, \ldots, I_{n-1}\} \tag{15}$$

where $I_j = (|v_j - u_j|/w_j)$, for $j=1,\ldots,n-1$

The values of each parameter used to generate the synthetic data set can be represented by:

$$[S] = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n-1} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{w_1,1} & s_{w_2,2} & \cdots & s_{w_{n-1},n-1} \end{bmatrix}$$

or implicitly expressed by:

$$s_{1j} = u_j + \frac{I_j}{2}, \quad \text{for } j=1,\ldots,n-1$$

$$s_{kj} = s_{k-1,j} + I_j, \quad \text{for } j=1,\ldots,n-1; \quad k=1,\ldots,w_j$$

It could be observed that the amount p of sets that will be generated depends on the number of intervals of each parameter. So p can be defined as:

$$p = w_1 \times w_2 \times \ldots \times w_{n-1} = \prod_{j=1}^{n-1} w_j \tag{16}$$

Step 4. Present the synthetic data set Y to the network in order to obtain the output parameter $Z=[z_{ij}]_{p\times 1}$, which is expected to have the same probability distribution $N_2(\bar{z}, S(z))$ as in the real parameter. To verify the network generalization, a comparison between the distributions $N_1(\bar{x}, S(x))$ and $N_2(\bar{z}, S(z))$ can be made. If $e_{x,z} = |\bar{x}_1 - \bar{z}_2|$ and $e_{S(x,z)} = |S(x)_1 - S(z)_2|$ are significant errors, this means that the training data are not representative for the process. Then, return to Step 1.

Step 5. Classify the parameters (columns) of the matrix $U=[Y, Z]_{p\times n}$ into intervals. As the data considered by this method have continuous values, the better context to represent these data is the many-valued context.

Step 6. Build a *formal context cross table*. The classification in intervals of the n variables of the objects in the domain (called of discretization), establishes a binary relationship between objects and attributes, named incidence, where an object has or not an attribute.

Step 7. Obtain the formal concept and build the line diagram. The formal concepts are the ordered pairs (*Object, Attribute*) obtained from the formal context.

Step 8. Apply the next closure algorithm in order to obtain the implication rules of the type: if…then…

4.2.1. Analysis of a parameters subset through the FCANN method

During the study of a physical process, it can be necessary to analyze the behavior of a parameters subset. Therefore, a constant value to some parameters can be assigned while others can vary,

Author's personal copy

allowing to analyze the qualitative behavior of the process. This study can be done through the FCANN method adapting Steps 3 and 6 that compose the method, as presented in Dias et al. (2008). These procedures are shown as follows:

Considering C as the set of all parameters indices that can be assumed as constant, Steps 3 and 6 may be redefined as:

Step 3. According to Step 3, a synthetic data set [S] should be built. To assign a constant value to a parameters subset of C, make:

If $j \in C$ then $S_{kj} = q$;   for $k = 1, \ldots, w_j$;

where $u_j \leqslant q \leqslant v_j$

Step 6. According to Step 6, the formal context $K{:} = (G, M, I)$ should be built. In this case, make $M = M'$, where:

$M' = \{m_i | m_i \in M;$ where $j \in C$ and $i \neq j\}$

Thus, the formal context built from a parameters subset is reduced. This way, the line diagram and the implication base from this new context can be more understandable for the process analysis.

4.3. *Metrics for evaluation of the rules extraction process*

Intrinsically, any method for knowledge extraction and representation from neural networks always presents some type of error, which may be introduced by the knowledge extraction method or by the neural network. There are two metrics for the FCANN method, the *representativeness* and the *fidelity*, which can be used to evaluate the process of rules extraction. These metrics are defined as:

Considering an implication base £ and a validation set T where:

$£ = \{f_1, f_2, \ldots, f_m\}$ and $T = \{t_1, t_2, \ldots, t_n\}$

where each rule f_i has the form "*if A then B*", and where A and B are subsets of the attributes set M.

Definition 1. : The *representativeness* is defined as the capacity of a rule $f_i \in £$ to represent a tuple $t_k \in T$. If a rule f_i exists, where A and B are valid to $t_{k,}$, then it is said that the rule f_i is able to represent this tuple.

Definition 2. : The *fidelity* is defined as the level of reliability of the implication base £. If a rule f_i exists, where A is valid and B is not valid to a tuple $t_{k,}$, then it is said that the rule f_i failed, reducing the fidelity.

The *representativeness* and the *fidelity* can be calculated from Eqs. (17) and (18), respectively:

$$R = \left(1 - \left(\frac{|NC|}{|T|}\right)\right) \times 100\% \tag{17}$$

where: R is the *representativeness*; NC is a non-represented data set; and T is the tested data set.

$$F = \left(1 - \frac{E}{|T|}\right) \times 100\% \tag{18}$$

where: F is the fidelity; E is the number of errors and T is the tested data set.

It is possible to adjust levels of *representativeness* and *fidelity* of the implication base. If the formal context $K{:} = (G, M, S)$ represents all knowledge of a neural network, so it is possible to assume that this knowledge is represented in the line diagram and, therefore, on canonical implication base £, where there is no *representativeness* flaws. However, it is not possible to construct a formal context that does not have flaws, because the context is built from a discretization process generating a many-valued context. This context causes a natural loss of its *representativeness* and, therefore, a decline in the *representativeness* of the implication base from this new context.

The many-valued context can represent a maximum number S of operational conditions of the process on analysis Eq. (19). However, even building a formal context with all combinations of the independent parameters (neural network inputs), it is not possible to ensure that all discretization intervals of the dependent parameter, generated by NN, are marked with an incidence in the formal context. Thus, there may be tuples t_k, which are not represented by the implication base £. This occurs due to the fact that the Next closure algorithm, used by the FCANN method, generates a minimal implication base, reducing the rate of *representativeness*.

$$S = (w)^{n-1} \tag{19}$$

where: S represents all combinations of the independent parameters of a formal context; w is the number of discretization intervals and n is the number of parameters of the neural network. Assuming that all independent parameters have the same discretization level, it is possible to consider: $w = w_1 = w_2 = \ldots = w_{n-1}$.

5. The FCANN method application for the cold rolling process

In this section, the application of the FCANN method for the cold rolling process will be presented. Initially, it is necessary to count on a database that supplies support for the neural representation of the process. For this reason, the Alexander's model (Alexander, 1972) was used as a generator of the database.

5.1. *Obtaining a database using Alexander's model*

To obtain the training and validation sets, the nominal operational condition for the rolling process was defined as:

- $h_i = 5.0\,(\text{mm})$
- $h_o = 3.6\,(\text{mm})$
- $\mu = 0.12$
- $t_f = 89.22\,(\text{N/mm}^2)$
- $t_b = 4.325\,(\text{N/mm}^2)$
- $Y = 256.325{+}468.187\varepsilon^{\,0.4275}$
- $\bar{Y} = 460.106\,(\text{N/mm}^2)$
- $P = 875310\,(\text{N/mm}^2)$

The data for the material and the rolling mill were defined as:

- $E = 200054\,(\text{N/mm}^2)$
- $v = 0.330$ Poisson's ratio of the strip
- $W = 500.0\,(\text{mm})$
- $R = 292.1\,(\text{mm})$
- $M = 4903.300\,(\text{N/mm})$
- $g = 1.846\,(\text{mm})$
- $S_i = 250.556\,(\text{N/mm}^2)$
- $S_i = S_o = 534.900\,(\text{N/mm}^2)$

Considering the nominal operational condition and the typical variations, on operational parameters suggested by Bryant et al. (1973) (see Table 3a and b), the population size was 10^6 records, where each parameter was varied in 10 intervals.

The rolling load value was obtained through the Alexander's model. Table 4 shows the inferior and superior limits of this parameter.

Author's personal copy

Table 3

h_i	h_o	μ	t_f	t_b	$\bar{y}$
a. Variation of the operational parameters					
$\pm 8\%$	$\pm 3\%$	$\pm 20\%$	$\pm 30\%$	$\pm 30\%$	$\pm 10\%$n

b. Variations in the rolling parameters						
Limit	h_i (mm)	h_o (mm)	t_b (N/mm^2)	t_f (N/mm^2)	μ	$\bar{y}$ (N/mm^2)
Minimum	4.60	3.492	3.030	62.458	0.096	383.869
Maximum	5.40	3.708	5.619	115.923	0.144	534.940
Increment	0.080	0.021	0.259	5.346	0.004	15.107

Table 4
Variation of rolling load.

	P (N/mm^2)
Minimum *(LimitInf)*	10202.1845
Maximum *(LimitSup)*	28737.3626
Average	17745.23
Standard deviation	2942.34

5.2. Obtaining the training and validation sets

To obtain representative training and validation sets for the process in Zárate and Bittencout (2002) and Zárate et al. (2006), a technique to build better-defined training sets was suggested. The technique is based in the expression Eq. (20), which has been borrowed from the statistics area and has been used in this work to calculate a suitable size for the training set:

$$n=\left(\frac{z}{e}\right)^{2*}(f^{*}(1-f)) \tag{20}$$

where n is the size of the sample, z is the reliance level, e is the error around the average and f is the population proportion. Fixing z in 90% ($z=1.645$), f in 0.5 and e in 3%, so $n\approx 752$. It is important to emphasize that the value e is not the maximum error expected for the network but for the sample. The following procedure has been adopted to build the training set:

1. For each variable of the network, the records containing the maximum and the minimum values have been selected.
2. The operational condition of the process (i.e. the most representative parameters in the data set) has been chosen and added to the training set.
3. The remaining elements have been randomly selected, with the purpose of reaching the size n of the training set.

the best final weights for the hidden and output layers with its bias weights are shown as follows:

$$W^h\begin{bmatrix}1.0875 & -0.6364 & -1.6868 & 0.0596 & 0.5301 & -2.4555\\ 1.1067 & 0.9291 & 1.7450 & 0.2780 & 1.4508 & -1.0157\\ 4.9758 & -1.3205 & -0.4317 & 0.0381 & -0.3250 & 0.5436\\ 3.0246 & -1.8265 & 4.9737 & -0.2657 & -1.0357 & 6.5506\\ 0.1056 & 1.2439 & -1.1441 & -0.0925 & 0.3360 & -0.4493\\ -0.3700 & 0.5878 & -0.4980 & 0.3943 & 0.4255 & -0.4846\\ 1.2936 & -0.0404 & -0.3537 & 0.4911 & -0.2658 & 0.9934\\ 2.2988 & -0.1578 & 1.7001 & 0.2870 & 0.1085 & 1.7665\\ -0.5041 & 0.4494 & 0.1645 & 0.1006 & 0.2580 & -3.7008\\ 0.0451 & 0.5815 & 0.1215 & 0.0283 & 0.6825 & -0.3352\\ 1.2870 & 0.2444 & -0.5138 & 0.6144 & -0.0496 & 1.1370\\ -0.3141 & 0.3506 & -0.0332 & 0.7756 & 0.0829 & -0.8123\\ 0.3310 & -0.0094 & 0.4496 & 0.1482 & 0.0057 & 0.8093\end{bmatrix}$$

$$\times W^h_{bias}\begin{bmatrix}0.1085\\ 0.0971\\ 2.0488\\ -13.9984\\ 1.1422\\ 0.4308\\ 0.9892\\ -7.8942\\ -1.0852\\ 0.2660\\ 0.5366\\ 0.1827\\ -0.2991\end{bmatrix} W^o\begin{bmatrix}-2.9475\\ 1.4831\\ 4.2028\\ 7.8188\\ -1.9307\\ -1.4457\\ 0.5915\\ 5.5544\\ -2.9404\\ -0.7128\\ 0.6960\\ -1.1435\\ 0.7147\end{bmatrix} W^o_{bias}[-2.5307]$$

5.3. Training and validating processes

For the training process, an error of 98 N/mm^2 (applied to each itemset), with $n=752$ data, $N=6$ inputs (h_i, h_o, μ, t_b, t_f, $\bar{y}$), and $M=1$ output, a neural network containing 13, $2n$+1, neurons in the hidden layer has been trained. For the training process, the *10-fold* cross validation technique was used (Kohavi, 1995), random values between -1.0 to +1.0 were considered for net weights, and the training was carried out with the back-propagation algorithm. After the training process, the average error $Error_{ave}=13.306$ and standard deviation $SD=15.324$ were obtained. The trained network produced satisfactory results and, therefore, it was considered representative of the process. In Table 5, results of the application of the cross validation technique are summarized and

5.4. Building synthetic data set and applying FCANN

The next step is to build the synthetic data set. In order to verify the influence of the *'number of values per parameter'* w_j, for $j=1\ldots5$ to be applied, the set $W=\{2, 3, 4, 5, 6\}$ was chosen (see Step 3 in Section 4.2). With W and $N=6$ input parameters, different synthetic data sets with p_j number of values were generated $P=\{64, 729, 4096, 15625, 46656\}$, as specified by Eq. (11).

After a synthetic data set has been generated, these data are presented to the network to obtain the output parameter (i.e. *P rolling load*). Thus, in order to verify the generalization capacity of the network, the probability distributions were compared (Step 4, Section 4.2). Table 6 shows reference values of the statistical

Author's personal copy

Table 5
Results of applied *k-fold* cross validation.

Error	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$	Global
Minimum	0.114	0.156	0.118	0.21	0.128	0.034	0.748	0.154	0.061	1.048	0.034
Maximum	58.802	39.630	102.955	32.48	27.579	64.623	39.916	323.999	86.177	49.973	323.999
Average	13.897	10.498	15.664	12.163	11.42	10.874	13.961	12.9603	14.586	17.07	13.306
SD	8.84	7.712	15.31	6.292	7.118	10.588	9.151	37.847	13.412	8.326	15.324

Table 6
Comparative analysis of the synthetic database.

Database	Number of values	Average	Standard Deviation
Original	1,000,000	1819.09	300.53
2 data per parameter	64	1692.46	192.34
3 data per parameter	729	1692.59	208.49
4 data per parameter	4096	1692.63	214.20
5 data per parameter	15625	1692.65	216.83
6 data per parameter	46656	1692.67	218.25

Table 7
Attributes created according to values ranges.

Parameter	1st Interval	2nd Interval
Input thickness	[4.6–5.0〉	[5.0–5.4]
Output thickness	[3.492–3.6〉	[3.6–3.708]
Friction coefficient	[0.096–0.12〉	[0.12–0.144]
Back tension	[3.027–4.325〉	[4.325–5.62]
Front tension	[62.454–89.22〉	[89.22–115.9]
Yield stress	[400.88–467.91〉	[467.9–534.9]
Rolling load	[10202.2–19469.7〉	[19469.8–28737.4]

distribution of both original and synthetic sets. It is possible to observe that approximately 95% of the data for the original data set is within the range $\bar{z}\pm 2S(z)$ of the synthetic set. Then, it is expected that the output given by the network, for the synthetic data, will be accurate.

As proposed by the Steps 5 and 6 in Section 4.2, the next stage is to convert data set into a cross table similar to the one presented in Table 2 (formal context). For such conversion, the values of the data set have been divided from 2 to 6 ranges. Table 7 shows an example for 2 intervals. It will be observed that the number of intervals determines the number of extracted rules. Each entry of data is now considered an object and the ranges represent attributes. After the conversion of the data into a formal context, the formal concept can be generated (see Step 7 in Section 4.2). Finally, the *Next Closure* algorithm is applied in order to extract the rules from neural network (Step 8, Section 4.2).

5.5. Rules extraction

After the application of the *Next Closure* algorithm, for two intervals of discretization and 4 data per parameter, 17 rules were obtained. It is necessary to observe that the algorithm finds the minimum number of rules (as discussed in Section 4.1.5) which describes the behavior of the process considered. Those rules are presented below:

1. if $\mu=1$ and $\bar{Y}=1$ then $P=1$
2. if $h_i=1$ and $\bar{Y}=1$ then $P=1$
3. if $h_i=1$ and $\mu=1$ and $t_f=2$ then $P=1$
4. if $h_i=1$ and $h_o=2$ and $\mu=1$ then $P=1$
5. if $h_i=2$ and $h_o=1$ and $\mu=2$ and $\bar{Y}=2$ then $P=2$
6. if $h_i=1$ and $P=2$ then $\bar{Y}=2$
7. if $h_i=1$ and $P=2$ then $\bar{Y}=2$
8. if $h_i=2$ and $h_o=1$ and $\mu=2$ and $P=1$ then $\bar{Y}=1$
9. if $\bar{Y}=1$ and $P=2$ then $h_i=2$ and $\mu=2$
10. if $h_o=1$ and $\mu=2$ and $\bar{Y}=2$ and $P=1$ then $h_i=1$
11. if $h_i=2$ and $h_o=1$ and $\bar{Y}=2$ and $P=1$ then $\mu=1$
12. if $\mu=1$ and $t_f=2$ and $\bar{Y}=2$ and $P=2$ then $h_i=2$
13. if $h_o=2$ and $\mu=1$ and $\bar{Y}=2$ and $P=2$ then $h_i=2$
14. if $h_i=1$ and $t_f=2$ and $\bar{Y}=2$ and $P=2$ then $\mu=2$
15. if $h_i=1$ and $h_o=2$ and $\bar{Y}=2$ and $P=2$ then $\mu=2$
16. if $h_i=2$ and $\mu=2$ and $\bar{Y}=2$ and $P=1$ then $h_o=2$
17. if $h_i=1$ and $\mu=1$ and $\bar{Y}=2$ and $P=2$ then $h_o=1$ and $t_f=1$

In Zárate (1998), it was verified that the back tension parameter (t_b) has little influence on the cold rolling process. It is possible to observe that this parameter is not present in any of the 17 rules gotten for 2 intervals, confirming the previous observation.

For automation, control and supervisory systems design in industrial processes, it is interesting to consider rules that have the dependent parameter as consequent of the type expressed by Eq. (21). Five rules (1–5) satisfy that condition, and these can help to identify which are the parameters that can change the operational condition, like the controllable variables for a control system.

$$\text{if } \{\text{independent parameters}\} \text{ then } \{\text{dependent parameter}\} \tag{21}$$

For 4 intervals of discretization and 4 data per parameter, 658 rules were obtained, where only 155 have the back tension parameter. An example of those rules is shown below:

1. if $h_o=3$ and $\mu=1$ and $t_b=3$ and $t_f=4$ and $\bar{Y}=4$ then $P=2$
2. if $h_o=2$ and $\mu=2$ and $t_b=3$ and $t_f=2$ and $\bar{Y}=2$ then $P=2$
3. if $h_o=2$ and $\mu=2$ and $t_b=2$ and $t_f=2$ and $\bar{Y}=2$ then $P=2$
4. if $h_o=1$ and $\mu=2$ and $t_b=1$ and $t_f=2$ and $\bar{Y}=4$ and $P=2$ then $h_i=1$
5. if $h_i=4$ and $\mu=2$ and $t_b=2$ and $t_f=3$ and $\bar{Y}=3$ and $P=2$ then $h_o=4$

Table 8 shows the number of rules obtained for different intervals of discretization and the number of values per parameter. Note that bigger the level of discretization, the more details about the process are obtained. Therefore, rules containing the back tension appeared. Table 9 shows the number of rules that can be obtained containing the back tension parameter. It is important to observe that the cold rolling process is a non-linear process. Thus, the influence of the back tension depends on the combined effect of the other parameters.

As the rules obtained describe the cold rolling process, some observations about the process can be taken, in order to deeply understand it. For example, it could be observed that when μ and $\bar{Y}$ belong to interval 1, certainly P (rolling load) belongs to interval 1 independently from the other parameters. A similar analysis can be applied for every rule.

Author's personal copy

Table 8
Number of rules obtained for different conditions.

Interval of discretization	Number of values per parameter				
	2	3	4	5	6
2	42	24	17	6	17
3	57	153	120	153	125
4	69	220	658	607	580
5	39	250	814	2301	2194

Table 9
Number of rules with the back tension parameter.

	Interval of discretization/number of values per parameter				
	2	3	4	5	6
2	18 (42%)	12 (50%)	0 (0%)	0 (0%)	0 (0%)
3	0 (0%)	9 (6%)	19 (16%)	14 (9%)	27 (22%)
4	31 (45%)	47 (21%)	155 (24%)	120 (20%)	164 (28%)
5	0 (0%)	49 (20%)	193 (24%)	727 (32%)	735 (33%)

5.6. Evaluation of the rule extraction process

In this section, the *representativeness* and *fidelity* will be used as metrics for the evaluation of results obtained through the FCANN method for the cold rolling mill process.

In Table 10, the *representativeness* of some implication bases, built from different discretization intervals, is presented. The discretization intervals number applied on the independent parameters (inputs of the net) is called m and the discretization interval number used on the dependent parameter (output of the net) is called n. If $n=1$, $m>1$ and $p=m$ (being p the number of sets that will be generated by the synthetic database, depending on the number of intervals of each parameter) then, all the operational conditions of the process on analysis will be represented in the formal context, and the dependent parameter corresponding to the output parameter supplied by the network will be always mapped in the same interval; thus, the implication base will not present *representativeness* flaws.

In Table 10, it can be observed that, when $m \geqslant n$ the *representativeness* value is bigger, for $m=5$ and $n=2$ the *representativeness* was of 95.343%. When $m<n$ the output value of the network can be attributed to an incorrect neighbor discretization interval. This is due to neural network error, which is considered acceptable. An extreme case can occur when m is smaller than n. For example, for $m=1$, there can be a set of objects (formal context lines) with the same antecedent, but several consequents eliminating the possibility of an existing representative rule. In short, m and n are parameters to adjust the quality and the representativity of the implication base and should be chosen preferably as $m>n$. In Fig. 3, these behaviors can be visually observed.

In order to generate all the formal context combinations, the number of values by parameter (considering this number equal for all input parameters $w=w_1=w_2=\ldots=w_{n-1}$ of the synthetic database, Section 4) must be equal to m. When $w<m$, the number of objects generated will be insufficient for the production of all possible combinations. In the other hand, if $w>m$, there may be repeated objects (operational conditions), which do not add information to the context, or there may exist objects with equal independent parameters leading to different dependent parameters. This was caused by the acceptable error given by the neural network.

Table 10
Representativeness (%) for different combinations of m and n.

m	N			
	2	3	4	5
2	90.96	74.734	34.193	36.237
3	92.338	91.124	60.422	68.928
4	94.471	93.405	90.255	67.786
5	95.343	93.927	91.68	89.752

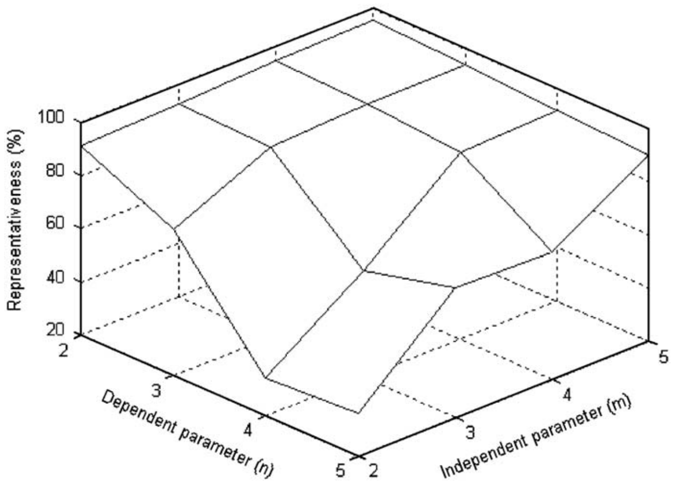


Fig. 3. *Representativeness* for different combinations of m and n.

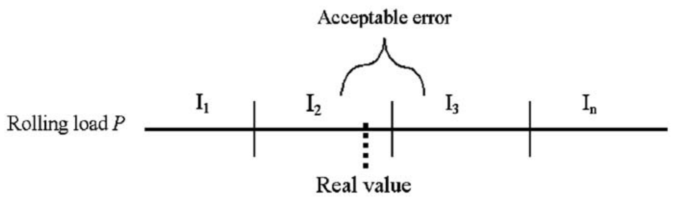


Fig. 4. Error produced by the neural network

While the low *representativeness* occurs by the losing of information in the formal context, the failure on the *fidelity* will occur when the output of the neural network presents a value within the acceptable error. This error can produce an incorrect classification of the value, making a rule to become failed (see Fig. 4). Still in Fig. 4, it is possible to observe that the acceptable error of the neural network oscillates between two adjacent intervals produced by the discretization process. This way, the value supplied by the network can be represented in the interval I_2 or I_3.

As in the *representativeness*, the *fidelity* can be controlled adjusting the discretization level of the independent and dependent parameters. Again, if $m>1$ and $n=1$, then all the rules will not present fidelity failures, because in this case there will be only one possible element in the $\langle$consequent$\rangle$ set.

In Table 11, the *fidelity* of some implication bases, built from different discretization number intervals, are presented. Note that the optimum values are found when m is bigger than n. Therefore, it is important to observe in Table 11 that the best *fidelity* value was obtained to $m=2$ and $n=4$. This occurred because, for this same combination, the *representativeness* presented a much lower value (Table 10). So a smaller amount of rules was used and, consequently, the *fidelity* of the implication base was better. However, as in the *representativeness*, where m and n are

Author's personal copy

Table 11
Fidelity (%) for different combinations of *m* and *n*.

m	N			
	2	3	4	5
2	85.773	87.965	98.713	97.537
3	91.224	86.734	92.648	93.152
4	93.674	90.800	87.125	87.985
5	94.975	92.912	89.953	87.731

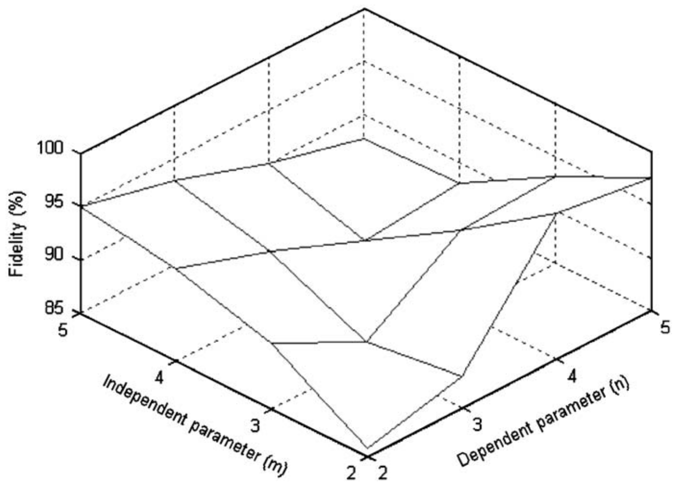


Fig. 5. Fidelity for each combination of *m* and *n*.

adjustment parameters of the *fidelity* of the implication base, these values should be preferably chosen as $m > n$. In Fig. 5, this behavior is observed and it is possible to note which regions have the best *fidelity* indices.

It is necessary to highlight that, although the *representativeness* and the *fidelity* are apparently similar metrics, there are intrinsic differences between them. In summary, when the *representativeness* fails, it means that for a specific tuple of the test base, there does not exist any rule in the implication base with a valid antecedent that attends it. On the other hand, the failure in the fidelity occurs if, for the tuple in analysis, there is a rule that attends it; that is, the rule antecedent is valid, although its consequent is not.

A second important point refers to the usage of a synthetic database. When using this base, the goal is to explore the generalization characteristic of the neural network, trying to extract learned relations and information during its training process. The usage of a synthetic database permits the FCANN method to introduce a metric of *amplitude* (Zárate et al., 2008), which refers to the capacity of incrementing the synthetic data set size, cases or operational conditions, which can be considered for rule extraction. However, this characteristic should be explored together with the discretization level used in the independent and dependent parameters, avoiding the appearing of repeated objects, i.e. of redundant operational conditions.

Other important points to be highlighted are that the number of values per parameter to generate the synthetic database, and the number of discretization intervals determine the cardinality of the implication base £ and the number of formal concepts of the line diagram. An increment in the number of parameters and in the discretization level can drastically increase the number of rules and the line diagram density, what would make difficult the comprehension of the extracted knowledge.

5.7. Mapping of the qualitative rules on load-curve

The FCANN method is able to extract and represent the learned physical behavior by neural networks. In Section 2, the behavior of the cold rolling process was discussed and, in this section, this behavior will be confirmed through the FCANN method. In this analysis, the rolling load behavior *P*, regarding the output thickness variation h_o, will be verified. The other parameters will be fixed in a constant value (Table 12), as proposed in Section 4.2.

Applying average values in the input thickness h_i, back tension t_b, front tension t_f, friction coefficient μ and in the average yield stress $\bar{Y}$ (Table 12), and using the same discretization level in the independent and dependent parameters, $m = n = 15$, it is possible to observe the relation of the rolling load *P* regarding the output thickness h_o through the line diagram (Fig. 6). In the diagram, two groups of formal concepts are observed (representing different operational conditions of the process): the left group, which presents a superior rolling load (*P*) value, is associated with the smaller output thickness (h_o) values; the right group, which presents an inferior rolling load (*P*) value, is associated with the biggest output thickness (h_o) values. This behavior is expected for this type of process and it permits to clearly observe the qualitative behavior of variables analyzed.

In addition to the line diagram analysis, another possibility to obtain the behavior of the cold rolling process is through the implication base £. Again, fixing the same discretization level in the independent and dependent parameters ($m = n = 15$) and analyzing the rolling load *P* and the output thickness h_o behaviors, for different friction coefficient values μ (Table 12), it is possible to build the characteristic load-curve (Fig. 1).

In Fig. 7, the behavior of the rolling load *P* for variations of $\pm 20\%$ over the nominal value of the friction coefficient μ is presented. The ordinate axis represents the 15 possible discretization intervals of rolling load and the abscissa axis represents the output thickness values h_o, for the same input thickness value h_i. For a minimum friction coefficient value μ, the load-curve stays under the nominal load-curve, as discussed in Section 2 (Fig. 1). For this case, all the rolling load values were mapped into the intervals 5 and 6. For an average friction coefficient value μ, the curve was mapped into the intervals 6 and 7. For the increase in the friction coefficient value, the load-curve stays above the nominal curve as expected, and all the rules were mapped into the intervals 7–9.

The friction coefficient variation produces an alteration in the rolling load characteristic values and this behavior was mapped by the FCANN method through the implication base. It is necessary to highlight that for the values of $m = n < 10$, this behavior cannot be noticed, since the discretization intervals cannot attend the rolling load variation.

Again *m* and *n* are adjustment parameters that permit to better explore the qualitative behavior of the process, besides controlling the *representativeness* and the *fidelity*.

Table 12
Constant values attributed during the analysis.

	Average		
Input thickness h_i(mm)	5		
Back tension t_b(N/mm^2)	4.3245		
Front tension t_f (N/mm^2)	89.1905		
Yield stress $\bar{Y}$ (N/mm^2)	459.4045		
	Average	Minimum	Maximum
Friction coefficient μ	0.12	0.096	0.144

Author's personal copy

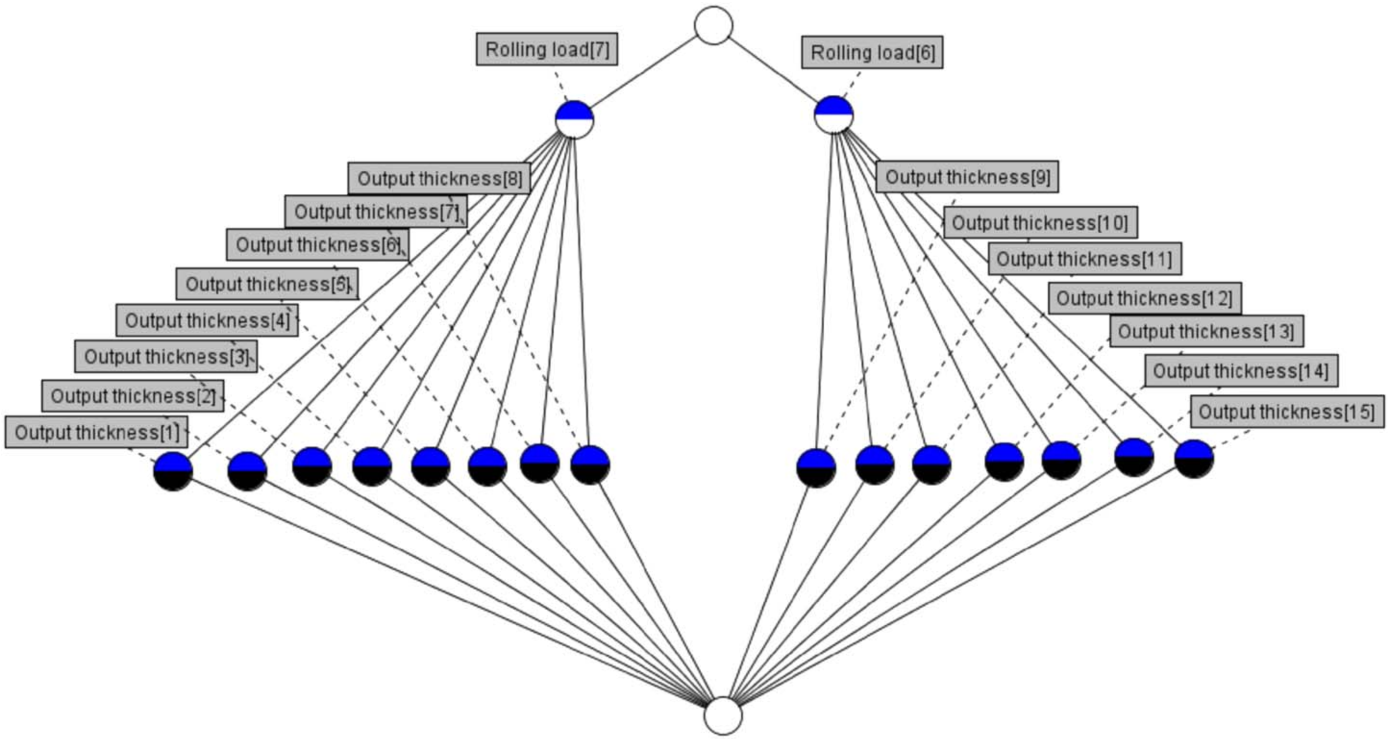


Fig. 6. Line diagram for analysis of the rolling load P and output thickness h_o.

The capacity of the FCANN method to represent the qualitative behavior of a process is very relevant. This characteristic can be used to understand processes without the necessity of complex mathematical models. The data can be collected directly from the process in the study to model the neural network and, afterwards, apply the FCANN method. A qualitative representation from the implication base and the graphic representation through the line diagram, what describes in a comprehensible form the process in study, are obtained.

6. Conclusions

In this work, the new approach denominated FCANN for knowledge extraction and representation from previously trained neural networks has been applied in the cold rolling process. The new approach uses the fundaments of the formal concept analysis to represent the behavior of the process being studied in a symbolic and qualitative form.

The results show that the neural representation of the rolling process is satisfactory and that the FCANN method is able to represent the qualitative behavior of the process. With the FCANN method, it was possible to analyze the rolling process through the line diagram and the implication base. Analyzing the line diagram for this case study, two groups of rolling load and their respective output thickness values were observed, showing the process behavior. Through the implication base, it was possible to build the characteristic load-curve for different friction coefficient values.

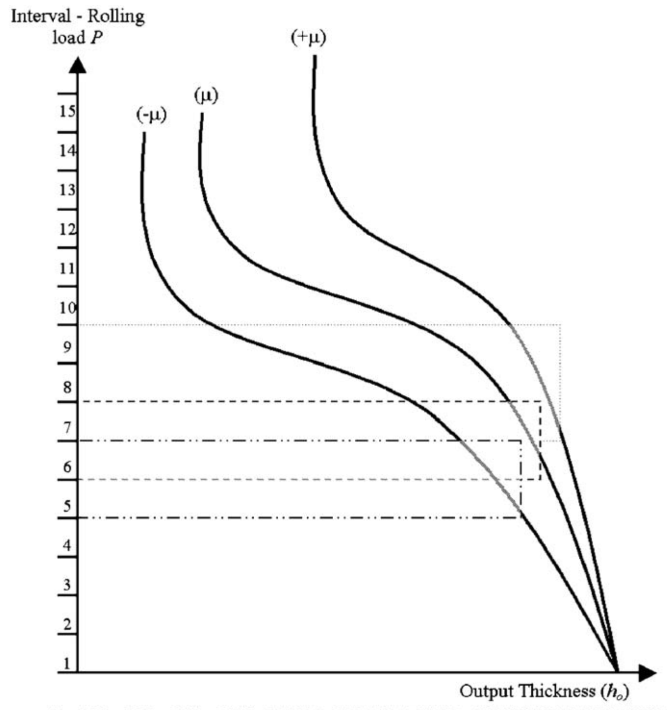
The reconstruction of the load-curve was possible due to the alterations proposed in some stages of the FCANN method, initially proposed in Dias et al. (2008). These alterations permit to analyze the behavior of specific parameters of a process, fixing the others. This way, it is possible to describe the physical behavior of the process whose analysis result can be useful for the learning process of the phenomenon in study. The FCANN method can be used to understand processes without the necessity of complex mathematical models. From a set of collected data, it is possible to train a neural network, apply the FCANN method and use the results to bear an intelligent system for on-line control and supervision.

Two metrics, the *representativeness* and the *fidelity,* were presented with the objective to evaluate the quality of the extracted rules. The best values of *fidelity* and *representativeness* were obtained when the dependent parameter (neural network output) has a smaller quantity of discretization intervals than the independent parameters (neural network inputs). However, it is important to notice that the quality of the rules obtained through the FCANN method also depends on the synthetic data set generated and on the neural network, which is responsible for the representation of the real process. The FCANN method is not capable of presenting good results when the neural network does not represent properly the process in analysis.

The synthetic database, as well as the neural network, should be representative of the process in analysis. Moreover, this database, together with the discretization level of the independent and dependent parameters, can be used as a precision control mechanism and the quantity of extracted rules.

It is important to note that the FCANN approach extracts relations among the parameters involved in the neural representation, independently of considering uniquely as antecedent the network inputs and as consequent its output. This means that FCANN tries to show the qualitative relations learned by the network, independently of its input–output structure. Notice that FCANN looks for a minimum implications base from which other implication rules can be deduced. FCANN produces qualitative behavior rules where the dependent parameters of the process can or cannot be the consequent variables of the rules extracted. This can be useful in industrial processes for automation, control and design of supervisory systems, where it is important to

Author's personal copy



Minimum (-μ)	**Average (μ)**	**Maximum (+μ)**
if $h_o = 1$ **then** $P = 6$	**if** $h_o = 1$ **then** $P = 7$	**if** $h_o = 1$ **then** $P = 7$
if $h_o = 2$ **then** $P = 6$	**if** $h_o = 2$ **then** $P = 7$	**if** $h_o = 2$ **then** $P = 7$
if $h_o = 3$ **then** $P = 5$	**if** $h_o = 3$ **then** $P = 7$	**if** $h_o = 3$ **then** $P = 8$
if $h_o = 4$ **then** $P = 5$	**if** $h_o = 4$ **then** $P = 7$	**if** $h_o = 4$ **then** $P = 8$
if $h_o = 5$ **then** $P = 5$	**if** $h_o = 5$ **then** $P = 7$	**if** $h_o = 5$ **then** $P = 8$
if $h_o = 6$ **then** $P = 5$	**if** $h_o = 6$ **then** $P = 7$	**if** $h_o = 6$ **then** $P = 8$
if $h_o = 7$ **then** $P = 5$	**if** $h_o = 7$ **then** $P = 7$	**if** $h_o = 7$ **then** $P = 8$
if $h_o = 8$ **then** $P = 5$	**if** $h_o = 8$ **then** $P = 7$	**if** $h_o = 8$ **then** $P = 8$
if $h_o = 9$ **then** $P = 5$	**if** $h_o = 9$ **then** $P = 6$	**if** $h_o = 9$ **then** $P = 8$
if $h_o = 10$ **then** $P = 5$	**if** $h_o = 10$ **then** $P = 6$	**if** $h_o = 10$ **then** $P = 8$
if $h_o = 11$ **then** $P = 5$	**if** $h_o = 11$ **then** $P = 6$	**if** $h_o = 11$ **then** $P = 8$
if $h_o = 12$ **then** $P = 5$	**if** $h_o = 12$ **then** $P = 6$	**if** $h_o = 12$ **then** $P = 9$
if $h_o = 13$ **then** $P = 5$	**if** $h_o = 13$ **then** $P = 6$	**if** $h_o = 13$ **then** $P = 9$
if $h_o = 14$ **then** $P = 5$	**if** $h_o = 14$ **then** $P = 6$	**if** $h_o = 14$ **then** $P = 9$
if $h_o = 15$ **then** $P = 5$	**if** $h_o = 15$ **then** $P = 6$	**if** $h_o = 15$ **then** $P = 9$

Fig. 7. Behavior of the rolling load *P* for variations of $\pm 20\%$ friction coefficient nominal value.

consider rules that have the dependent parameters as consequent. This can help to identify which parameters can change the operational condition, such as the controllable variables for a control system.

Acknowledgments

The authors would like to thank the financial support of the National Research Council CNPq-Brazil. Project CT-INFO/MCT/CNPq No. 031/2004, Process: 506512/2004–5.

References

Aistleitner, K., Mattersdorfer, G., 1996. Neural Network for identification of roll eccentricity in rolling mills. Journal of Materials Processing Technology 60, 387–392.

Alexander, J.M., 1972. On the theory of rolling. Proceedings of the Royal Society of London A 326, 535–563.

Altincay, H., Demirekler, M., 2002. Why does output Normalization create problems in multiple classifier systems? In: Proceedings of the ICPR, pp. 775–778.

Andersen, K., Cook, G.E., Barnett, R.J., 1992. Gas tungsten arc welding process control using artificial neural networks. International Trends in Welding, Science and Technology, ASM: 877–1030.

Andrews, R., Geva, S., 2002. Rule extraction from cluster neural nets. Neurocomputing 47, 1–20.

Bland, D.R., Ford, H., 1952. An approximate treatment of the elastic compression of the strip in cold rolling. Journal of the Iron and Steel Institute 171, 245–249.

Bryant, G.F., Edwards, W.J., McClure, C.H., 1973. Automation of Tandem mills, London: The Iron and Steel Institute. Cap. 1, pp. 1–29.

Carpineto, C., Romano, G., 2004. Concept Data Analysis: Theory and Applications. Wiley, New York.

Craven, M., Shavlik, J., 1999. Rule Extraction: Where Do We Go from Here?, Technical Report, University of Wisconsin Machine Learning Research Group Working Paper.

Cristea, A., Cristea, P., Okamoto, T., 1997. Neural network knowledge extraction. Revue Roumaine des Sciences Techniques, Serie Electrotechnique et Energetique 42 (4), 477–491.

Davey, B., Priestley, H., 1990. Introduction to Lattices and Order. Cambridge University Press, Cambridge, England.

Dias, S.M., Nogueira, B.M., Zárate, L.E., 2008. Adaptation of FCANN Method to Extract and Represent Comprehensible Knowledge from Neural Networks. New Challenges in Applied Intelligence Technologies-Series: Studies in Computational Intelligence, vol. 134. Springer, Berlin/Heidelberg, pp. 163–172.

Dieter, G.E., 1976. Mechanical Metallurgy, second ed. McGraw-Hill, Inc.

Gálvez, J.M., Zárate, L.E., Helman, H., 2003. A model-based predictive control scheme for steal rolling mills using neural networks. Journal of the Brazilian Society of Mechanical Sciences and Engineering 25 (1), 85–89.

Gams, M., Karba, N., Drobnic, M., 1997. Integration of multiple reasoning systems for process control. Engineering Applications of Artificial Intelligence 10 (1), 41–46.

Ganter, B., 2002. Formal concept analysis: algorithmic aspects. TU Dresden, Germany, Technical Report

Ganter, B., Wille, R., 1996. Formal Concept Analysis: Mathematical foundations. Springer, Berlin.

Gratzer, G., 1998. General Lattice Theory, Applied Lattice Theory: Formal Concept Analysis, second ed. Birkhäuser Verlag, pp. 591–605.

Guigues, J.L., Duquenne, V., 1986. Familles minimales d'implications Informatives résultant d'um Tableau de denne'es binaire. Mathematiques et Sciences Humaines 95, 5–18.

Gunasekera, J.S., Zhengjie, J., Malas, J.C., Rabelo, L., 1998. Development of a neural network model for a cold rolling process. Engineering Applications of Artificial Intelligence 11, 597–603.

Haykin, S., 1999. Neural Networks—A comprehensive Foundation, second ed. Pearson Education, India.

Hecht-Nielsen, R., 1990. Neurocomputing. Addison-Wesley Publishing Co., New York.

Hayashi, Y., Buckley, J.J., 1994. Approximation between fuzzy expert systems and neural networks. International Journal of Approximate Reasoning 10, 63–73.

Jung, J., Im, Y., Lee-Kwang, H., 1995. Development of fuzzy control algorithm for shape control in cold rolling. Journal of Materials Processing Technology 48, 187–195.

Kim, D.H., Lee, Y., Kim, B.M., 2002. Applications of ANN for the dimensional accuracy of workpiece in hot rod rolling process. Journal of Materials Processing Technology 130–131, 214–218.

Kim, H.J., Mahfouf, M., Yang, Y.Y., 2008. Modelling of hot strip rolling process using a hybrid neural network approach. Journal of Materials Processing Technology 201, 101–105.

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pp. 1137–1143.

Kovács, Z.L., 1996. Redes Neurais Artificiais, Edição Acadêmica São Paulo, Cap 5, pp. 75–76, São Paulo, Brasil.

Larkiola, J., Myllykoski, P., Nylander, J., Korhonen, A.S., Cser, L., 1996. Prediction of rolling force in cold rolling by using physical models and neural computing. Journal of Materials Processing Technology 60, 381–386.

Larkiola, J., Myllykoski, P., Korhonen, A.S., Cser, L., 1998. The role of neural networks in the optimization of rolling process. Journal of Materials Processing Technology 80–81, 16–23.

Lenard, J.G., Zhang, S., 1997. A study of friction during the lubricated cold rolling of an aluminum alloy. Journal of Materials Processing Technology 72, 293–301.

Mitra, S., Hayashi, Y., 2000. Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Transactions on Neural Networks 11 (3), 748–768.

Montmitonnet, P., 2006. Hot and cold strip rolling process. Computer Methods in Applied Mechanics and Engineering 195, 6604–6625.

Myllykoski, P., Larkiola, J., Nylander, J., 1996. Development of prediction model for mechanical properties of batch annealed thin steel strip by using artificial neural network modeling. Journal of Materials Processing Technology 60, 399–404.

Peng, Y., Liu, H., Du, R., 2008. A neural network-based shape control system for cold rolling operations. Journal of Materials Processing Technology 202, 54–60.

Petty, D.M., 1996. Application of process modelling—an industrial view. Journal of Materials Processing Technology 60 (1), 421–426.

Shlang, M., Lang, B., Poppe, T., Runkler, T., Weinzier, K., 2001. Current and future development in neural computation in steel processing. Control Engineering Practice 9, 975–986.

Smartt, H.B., 1992. Intelligent Sensing and Control of Arc Welding. International Trends in Welding, Science and Technology, ASM, 843–851.

Author's personal copy

Son, J.S., Lee, D.M., Kim, I.S., Choi, S.K., 2004. A study on genetic algorithm to select architecture of an optimal neural network in the hot rolling process. Journal of Materials Processing Technology 153–154, 643–648.

Son, J.S., Lee, D.M., Kim, I.S., Choi, S.K., 2005. A study on on-line neural network for prediction for rolling force in hot-rolling mill. Journal of Materials Processing Technology 164–165, 1612–1617.

Tarca, A.L., Cooke, J.E.K., 2005. A robust neural network approach for spatial and intensity-dependent normalization of cDNA microarray data 21(11), 2674–2683.

Thangavel, S., Palanisamy, V., Duraiswamy, K., Chenthur Pandian, S., 2006. A hybrid intelligent controller for control of strip tension and looper height in tandem finishing mill. IEEE International Conference on Industrial Technology 15–17, 434–439.

Thangavel, S., Palanisamy, V., Duraiswamy, K., Chenthur Pandian, S., 2007. A genetic-based hybrid intelligent controller for looper tension in steel rolling mills. International Journal of Modelling, Identification and Control 2 (3), 219–228.

Wang, L., Frayman, Y., 2002. A dynamically generated fuzzy neural network and its application to torsional vibration control of tandem cold rolling mill spindles. Engineering Applications of Artificial Intelligence 15, 541–550.

Xiaoguang, S., Ning, H., Wei, W., Zhenbang, S., Guodong, W., 1999. Application of synergetic artificial intelligence to the scheduling in the finishing train of hot strip mills. Journal of Materials Processing Technology 60 (1–4), 405–408.

Yang, Y.Y., Linkens, D.A., Talamantes-Silva, J., 2004. Roll lod prediction—data collection, analysis and neural network modeling. Journal of Materials Processing Technology 152, 304–315.

Zárate, L.E., 1998, Doctoral Thesis. Um Método para Análise da Laminação Tandem a Frio. Universidade Federal de Minas Gerais, Escola de Engenharia Metalúrgica e de Minas, Belo Horizonte, MG, Brasil.

Zárate, L.E., Bittencout, F.R., 2001. Analise Qualitativa de Processos Através dos Fatores de sensibilidade via Redes Neurais e sua Aplicação na Laminação em Tandem. V Congresso Brasileiro de Redes Neurais, Rio de Janeiro, Rj, Brazil, vol. 1, pp. 421–426.

Zárate, L.E., Bittencout, F.R., 2002. Controle da Laminação a frio Baseado em Redes Neurais com Capacidade de Generalização e Lógica Nebulosa via Fatores de Sensibilidade. XIV Brazilian Automatic Control Conference, Natal, RN, Brazil, vol. I, pp. 979–984.

Zárate, L.E., Bittencout, F.R., 2008. Representation and control of the cold rolling process through artificial neural networks via sensitivity factors. Journal of Materials Processing Technology 197, 344–362.

Zárate, L.E., Pereira, E.M.D., Oliveira, L.A.R., Gil, V.P., Santos, T.R.A., Nogueira, B.M., 2006. Techniques for Training Sets Selection in the Representation of a Thermosiphon System Via ANN, In: Proceedings of the IJCNN, Vancouver, Canadá, pp. 2736–2741.

Zárate, L.E., Gálvez, J.M., Helman, H., 1998. A neural networks-based controller for steel rolling mills by using sensitivity functions. XII Brazilian Automatic Control Conference I, 29–34.

Zárate, L.E., Dias, S.M., Song, M.A.J., 2008. FCANN: A new approach for extraction and representation of knowledge from ANN trained via Formal Concept Analysis. Neurocomputing 71, 2670–2684 (doi:10.1016/j.neucom.2007.09.025).

Luis E. Zárate was born in Lima, Perú. He received the B.S. degree in Electronics Engineering in 1981 from URP, Perú. He received the M.S and Dr. degrees from the Federal University of Minas Gerais, Brazil in 1992 and 1998, respectively. Since 1992 he hold research and teaching position at Pontifical Catholic University of Minas Gerais, Brazil. Professor Zárate has research interest in Computational Intelligence, Neural Networks and Data Mining and their applications in industrial processes. He is the coordinator of the Applied Computational Intelligence Laboratory and is responsible for several research projects supported by governmental and particular organizations in Brazil.

Sérgio M. Dias was born in Minas Gerais, Brazil. He received B.S. degree in Computer Science in 2007 from Pontifical Catholic University of Minas Gerais. Sérgio Dias has research interest in Formal Concept Analysis, Computational Intelligence and Neural Networks. He studies Computer Science at the post-graduated program at Federal University of Minas Gerais, where he studies the algorithms to constructing concept lattices. He also collaborates with the Applied Computational Intelligence Laboratory, where he studies the Applicability of the Formal Concept Analysis for Representation of Knowledge of Trained Neural Networks.