"Soft sensor" measuring for ball mills based on LSTM neural networks

Abstract. In the present work, considering the strong uncertainty(nonlinearities) in ball mills, a soft sensor is modeling to determine the amount of material processed in a ball mill, feed level, proposing a novel soft sensor based on Deep learning technique (a neuronal network, RNN-LSTM type), to improve the accuracy and reliability of fill level measurement.
This measurement methodology of ball mill fill level uses Two types of signals, the vibration and sound, these two signals combined avoid distortions in the environment, then it uses a neural encoder method for extract the principal features of the signals, and finally for the modeling uses a neural network based on LSTM (Long Sort Term Memory), which has the ability to remember samples over time being very efficient when working with sequences, as in this case, the filling sequence in the ball mill. The efficiency of this model based on LSTM is seen in the comparison with other regression techniques, Random forest regression, SVM regression, Nearest neighbor's regression, LS-SVM, MLP neural network, PLS Regression, Regression tree, adaBoost and Mean learner.

Keywords: Ball mill, Ball mill fill level, Soft sensor, LSTM regression.

1. Introduction

A ball mill is a slightly inclined rotating horizontal cylinder, partially filled with steel balls, which grind the material by friction and impact with the balls that until the degree of fineness necessary [1]. In order to achieve the desired particle size, the milling under industrial conditions is usually performed in "grinding circuits" [1]. In the "grinding circuit", ball mills have the lowest energy efficiency and also is difficult to measure the amount of material processed [2]. The amount of material processed is directly related to energy expenditure, so an efficient fill measurement of material contributes to energy savings from a 10%, perform the automation of the plant (ball mill), stability of the system, program the use of the mill in a flexible way, improve the stability of the system, prevent system overflows and jams [3].

The problem in ball mill is the difficulty to have a precise measurement for fill level, because material processed is measured indirectly. For solve this problem already exist some methods and measurements, including differential pressure sensors method [4]. The power consumption in the motor method [5], the ultrasonic method [6], the acoustic sensors method [7], [8], the vibration sensor method [9], [10], [11], but each of these methods have problems with measurement. The differential pressure method has problems with measurement delays and is heavily influence by the environment [12]. The power consumption in the motor method is widely used in the industry since it is not affected by the environment, but has a poor resolution and also is affected by the wear coals in the electric motor [12]. The ultrasonic method has problems with the processed material and steel balls, because when the fineness of the material changes the ultrasonic sensor signal also changes [12]. The acoustic sensors method is able to measure the fill level but is also affected for the environment noise [12]. The vibration sensor method, is commonly used in many equipment but also has the environment problem, but is not heavy affected as others methods [13] [12].

In this paper, the percent work is going to be combined two kinds of signals, the sound and vibration signal, it is to avoid the environment problems and improve the measurement process. In the last years some works about ball mill fill level measurements have being published [3], [13], [8], [9] [11], etc. All these works afford the problem with nonlinear methods like Back Propagation Neural Network (BPNN) [8], Cloud model [3], etc. These researches have made great progress for measurement of fill level in ball mills, but did not take the variable time. The time variable is important, because the fill level in ball mills is a sequence. For improve the accuracy, this paper introduces a LSTM neural network, it works with data sequence to make regressions, proving to be very efficient in other areas gives grates results in the speech generation, or predictions of written text [14].

The structure of the paper is as follows, in the second section is explain the concepts of LSTM networks, then the third section shows the soft sensor model, in the fourth section takes place the experiments and results, making comparisons with other methods using the same dataset, and final in the fifth section are the conclusions.

2. LSTM networks

2.1 LSTM concepts

The Recurrent Neural Network (RNN-LSTM) type neuronal networks are widely used for language modeling and speech recognition [15], for sequences, reprehending the state of the art in these areas. The RNN is a deep neural network (DNN) that is adapted to sequence data. RNNs maintain a vector of activations for each time step, which makes the RNN extremely deep. Their depth, in turn, makes them difficult to train due to the exploding and the vanishing gradient problems [15]. There have been a number of attempts to address the difficulty of training RNNs. Vanishing gradients were successfully addressed by Hochreiter & Schmidhuber in 1997, they developed the Long Short-Term Memory (LSTM) architecture which is resistant to the vanishing gradient problem.

LSTM neural networks have the special ability to remember important data over time designed exclusively to solve the long-term memory dependency problem [15], so they have very good results in sequencing tasks. It is for this reason that this method was chosen to perform the modeling of soft sensor for the ball mill. All LSTM networks are in the form of a chain and their common structure is as show the follow picture [15].

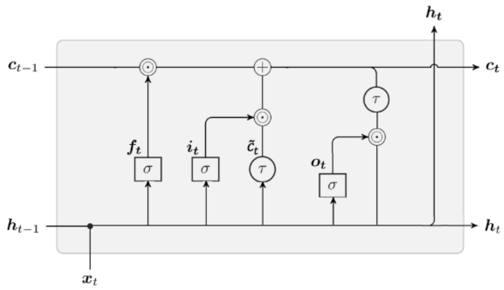

figure 1 classic structure for LSTM networks

The interaction of each line and interconnection have vectors, being a flow of tensors that is carried within each cell, each line loads a vector, from the output node to the next input node.

The most important part in the LSTM neural network structure is the "***state***", the state is represented by the horizontal line running through the whole diagram (from C_{t-1} *to* C_t figure 1), LSTM networks has the ability to adhere or suppress information to the state of the "Cell", through gates. Gates are composed of σ (sigmoid) functions and a multiplication operation at the output of each neural network. The first step in an LSTM type network is passes the information through the state line, this decision is made through the σ (sigmoid) function, this stage is called "forget gate layer". The following equation represents the forget gate.

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f)$$

Equation 1 forget gate layer

This gate takes h_{t-1} and x_t in the input and to the output delivery 0 and 1, for each state, in the weights of the neuron C_{t-1}. The next step is to save the new information in the weights of the neuron, this step is divided in two, the first stage is a σ (sigmoid) function called "input gate layer" which decides the values that will go to the next stage. The second stage is a τ (tanh) function, it creates the values that will be selected by the σ sigmoid function, $\tilde{C}_t$ [16].

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \bullet [h_{t-1}, x_t] + b_C)$$

Equation 2

The end result of this stage is:

$$C_t = f_t \bullet C_{t-1} + i_t \bullet \tilde{C}_t$$

Equation 3

Finally, the output of the neuronal state the output of the neural network is based on state filtering, the filtering is performed with a sigmoid function, which again according to training data, decides which values pass and which values do not, After the state values pass through the τ (tanh) function, this function throws values between -1 and 1 [16].

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \bullet \tanh(C_t)$$

Equation 4

2.2 The LSTM as regression model

Commonly the LSTM neural networks are used to generate data sequences and voice recognition, but in this case a structure is adapted to work as a regression model. The LSTM networks has different configuration, it depends on how many time steps or h_t we are going to take in the sequence-to-sequence task. We can set **n** inputs and **m** outputs, **one** input **m** outputs, **n** inputs and **one** output, in the present task we have the configuration of n inputs and one output, corresponding to n=6 and the output is h_7.

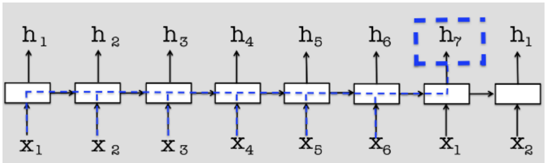

figure 2 configuration of the structure in the percent task, n=6 inputs an one output $h_7 h_t$

The LSTM neural networks are not designed to perform regressions, so they have to be adapted within a neural network, the neuronal networks that perform regression are formed by the input layer, several hidden layers, and an output layer, which contains a single neuron in the output, without activation function, which performs the regression. In the regression structure, the LSTM layer is replacing the hidden layers, as shows below.

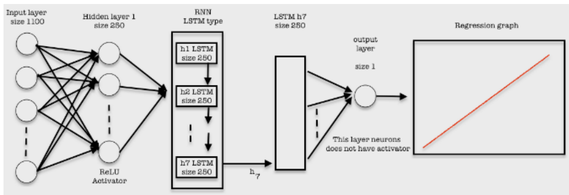

figure 3 The regression model structure

The neural network in the first layer has the input neurons, this layer is followed by a layer of hidden neurons (full connected layer), the activation functions is ReLU, also has biases and at the end of this layer is the dropout function, which avoids the over feeding, in the LSTM layer in the case of regression the structure has n inputs and one output, this output has n recurrent neurons, and to finish, the n neurons of the LSTM layer are full connected to the last layer, the regression layer this layer with also has bias but does not have activation function, allowing the neuronal values directly to pass through.

2.3 The lost function in the regression model

The loss function has two parts, the first is to calculate the value to be minimized and the second is the minimization function. The value to be minimized is calculated from the mini-batch, recalling that each and are vectors, according to the procedure the loss is equal to the RMSE function.

$$P_{target_i} = (y_i - y_i)^2$$

Equation 5 target to minimize

The minimization function, commonly is called the loss function, which we want to minimize the average negative log probability of the calculated target (fill level measured error).

$$loss = -\frac{1}{N}\sum_{t=1}^{N} Ln(P_{t\arg ert_i})$$
$$e^{-\frac{1}{N}\sum_{t=1}^{N} Ln(P_{t\arg ert_i})} = e^{loss}$$

Equation 6

In the training process its value is monitored throughout the process. The typical measure reported is average per-state perplexity.

3. Soft sensor model

The "soft sensor" are commonly used in control systems, for the measurement of material processed in ball mills [17]. Many of these "soft sensors" use sound or vibration signals, this signals come from a ball mill shells [3]. According to the literature, the procedures for processing data, in the "soft sensor" are: transformation, cleaning and reduction [18], once these procedures are performed, the system modeling is performed. In the present work we have three modules, these are: pre-processing module, feature extraction module and Build model module.

- The data pre-processing module: Extracts the stable signal from the acquired samples, and transforms these signals into the frequency domain with the Welch algorithm.
- Feature extraction module: Performs the reduction of the database, extracting the most important characteristics of the database and form of the training and test set.
- Build model module: Using the training data set, is trained the LSTM model, and with the test data set the regression is generated.

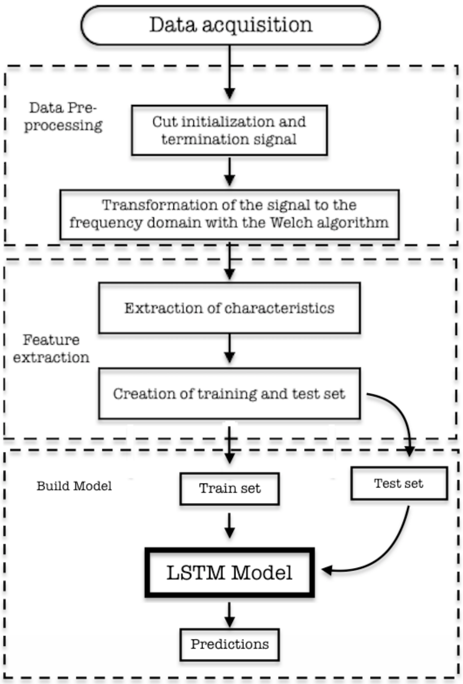

figure 4 proposed soft sensor model for the level

In the process of characteristics extraction is used a neuronal encoder to reduce the dataset. The already reduced signals serve to train the neural network, which is composed of a full connected layer, a LSTM layer and the output layer is full connected to one neuron witch perform the regression. The output neuron in the last layer does not have an activation function. In the test stage, the test data, never seen before by the neural network, is used to perform the regression, this stage simulates the online stage of the soft sensor.

4. Experiments and results

4.1 Data acquisition

The experiments were carried out in a laboratory, in a ball mill with 60 cm in diameter and 80 cm long, having a capacity of 200 liters. The ball mill is controlled by a 4 kW tri-phase motor, it has a rotational speeds of 1 r/1.4s or 42.86 r/min, to perform data extraction at different filling levels, were filled loads of material. A data acquisition system is used to pick up sound, vibration and the motor control frequency signals from a microphone, accelerometer and the motor. The signals were acquired at a sampling frequency of 51200Hz. The installation of the accelerometer and microphone is in the center of the shell of the ball mill, to capture the vibrations coming from all directions. In the data extraction stage, a constant speed of 42.86 r/min is defined for the rotation, and a constant ball Wight of 292L is load.

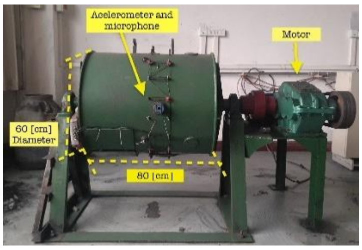

figure 5 The ball mil machine

The neural network needs data entry and associated labels, where each sound and vibration signal are the data entry and the associated labels are the filing level in the ball mill. For generate the dataset, the ball mill was filing with a range of 0L to 139L of material, with a deferens of 1L in every step, generating 140 audio and vibration samples, with 140 labels. Subsequently, the sampling of the signal is a time interval between 60 and 70 seconds of each fill level, the length of the data is approximately $51.2KHz \times 65s \cong 3{,}3 \times 10^6$ this procedure is similar to the procedure performed in [13].

4.2 Cut initialization and termination signal

For extract the stable signal in the samples first is subtracted the mean of each signal, it is performed to avoid the zero drift, $s = x(t) - \frac{1}{N}\sum_{t=0}^{N} x(t)$ and then to cutoff the beginning and ending of each recording, for only take the stable part, is used the control frequency of the motor. For find the cutting points, (deleting the initialization and ending parts in an automatic way) were searched all the position where values are above 48 Hz (the frequency control has a variation between 0 and 50Hz), in a vector $position = [s_n, s_{n+1}, s_{n+2}, \ldots, s_{n+p}]$ where $s_n, s_{n+1}, s_{n+2}, \ldots, s_{n+p}$ represents positions, where $n \approx 3 \cdot 10^5$ and $n + p \approx 3.28 \cdot 10^6 \rightarrow p \approx 3 \cdot 10^6$ then the first and last positions number is taken from the "*position*" vector, s_n and s_{n+p}, then are taken all the points between thus tow positions in the sound and vibration signal, this results in signal with length between $3 \cdot 10^6$ and $3.28 \cdot 10^6$.

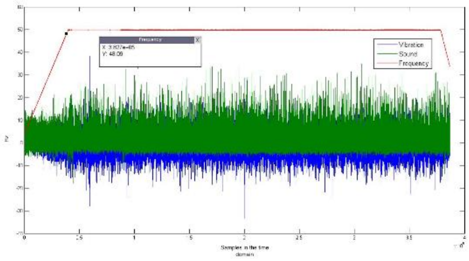

figure 6 This image shows, the motor frequency signal control in [Hz], vibration and audio signals.
(signals of audio and vibration are not real-scale)

The total of the samples made in laboratory are 140 audio and 140 vibration signals and taking into account that each sample is part of a steady state in the grinding, without transition between level filling, or initialization-stopping states, a sweep of each signal is performed, with a window size of $L = 3 \cdot 10^6 / 22 = 136363$ samples ($3 \cdot 10^6$ is the smallest sample size), without overlap, at each step a sub-sample is extracted and 22 steps are taken in each sample (components not enter in the sweep are discarded), with what is formed $140 \times 22 = 3080$ samples in total, Each sample retains its original label it is coming from , which corresponds to level filling.

4.3 Transforming to the frequency domain

In order to extract the main characteristics of the audio and vibration signals, the transformation to the frequency domain is done with Welch method [19], because it has low computational cost and had a good performance in previous ball mill soft sensors, [3], [13], is choose. Welch method has a spectral density $p(f) \mid f \mid \leq \frac{1}{2}$, according to the window size L were chosen [19]. The choose window, in this case were a Hamming = W window, with a L = Fs / 2, where Fs = 51200 Hz, resulting in a L = 25600 length, and a power spectral density vector dimension of L / 2 = 12800, Following is showing the transformation of one sample.

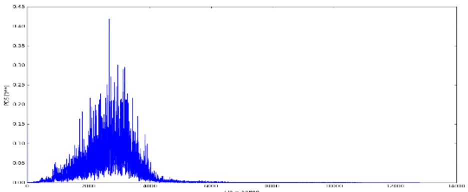

figure 7 This image shows the first sample of sound signal in the frequency domain.

Once all the samples were transformed to the frequency domain, was formed a matrix with all of them, this matrix has a dimension of 3080 number of samples, per 12800 number of frequencies (it is for sound and vibration signal respectively), following is showing part of these matrices in picture format, where the brightness colors represent larger values and dark colors represent lower values.

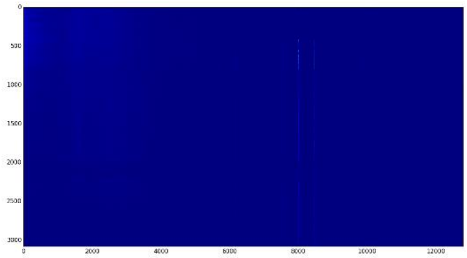

figure 8 This image shows the sound matrix with 3080 by 12800 components.

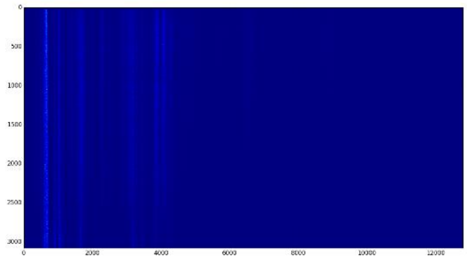

figure 9 This image shows the vibration matrix with 3080 by12800 components.

In accordance with the procedure for data processing in a soft sensor [18] the data transformation has been performed, now the database has to be cleaned. After analyzing the samples obtained, it is concluded that the database has a frequency spectrum of 0 Hz to 11 kHz, so that frequencies above 11 kHz are eliminated. The sound and vibration matrices are left with 3080 x 11000 components. To conclude, a mean pooling is performed on the matrices, the mean pooling is done with a size of 20 components, without overlapping, leaving the matrices of sound and vibration with 3080 by 550 components.

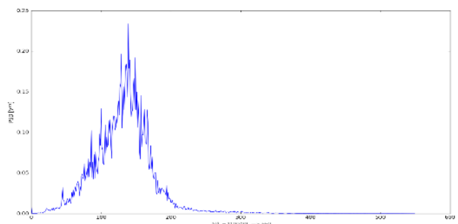

figure 10 This image shows a sample of the audio signal, after the process of pooling mean, this signal is the same as is showed in the figure 7.

4.4 Extraction of characteristics

The extraction of characteristics is made to reduce the dataset dimension, it reduces the computational load in the process of regression or classification offers improvements, eliminating the noise or zeros from the characteristics [18] this data reduction is the last step in data procesin prosedure. In the present work, to perform the extraction of characteristic, we choose a neuronal encoder.

The neural encoding, is a neuronal encoder, this is based on a MLP (Multilayer perceptron) which performs a regression, when it reaches a deceased training level, the regression stops and all the database is introduced through the neural network and an intermediate layer acts as an encoding output, reducing the data base and extracting the main characteristics, the coding layer has 250 neurons, (e.i) the resulting database has a dimension of 3080 by 250, for audio and vibration signals.

4.5 The train-test data separation

The neural networks need a database to perform their training and test, to form the train and test dataset, was performing a simple algorithm. The samples obtained are 3080, after the reduction procedure, each sample is a vector of 250 components and all the samples forms a matrix of 3080 by 250, for sound and vibration signal, then these tow matrices are appending, resulting in a matrix of 3080 by 500, to form the train and test datasets this matrix is dived in two, by the samples numbers (3080), the even samples are part of the train set (1540 x 500) and the odd samples are part of the test set (1540 x 500), in this way we have a continuous group of samples for the training and testing of the neural network.

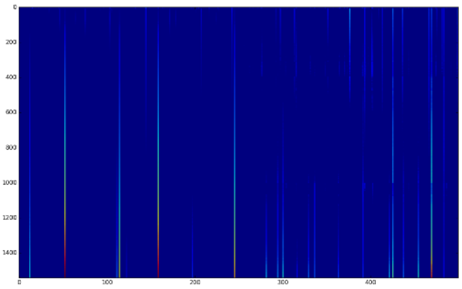

figure 11 This image shows the training array, which is 1540 by 500.

4.6 Model regression

4.6.1 The training stage

The core of the model consists of an LSTM cell that processes sample to computes probabilities of the possible fill level. The memory state of the network is initialized with a vector of zeros and gets updated after reading each simple. The training set is duplicated in by 4, it is because the neural networks as more data has more accurate are the results, resulting in a training set of 6160 samples with 500 characteristics each one. For computational reasons, the data is processed in mini-batches of size 250*11. The mini-batch is dived in 250 sub matrices, each one with 11 samples, these sub matrices are swept with a window of length 6, with a sweep step equal to 1, overlapped, until to reach the 11th sample, each sub-matrix is a sample time, and each sub-matrix, retains its respective label it is from.

In order to make the learning process tractable, it is a common practice to truncate the gradients for backpropagation to a fixed number in this case the number is 22500 epochs reaching a training loss-RMSE value of 0.05, and the learning rate was set to 0.0015 using Adam optimizer.

4.6.2 The test stage

The test part is done with a group of data never previously "seen" by the neural network, simulating an online measuring procedure in ball mill fill level. The data test has the same configuration as the training data, this test set has 1540 samples and each one has 500 components (as the train set), but the mini-batches has a change, being 560*11=6160 he entire test set, then as in the training stage, the batch is divided in 560 to form sub matrices of 11 samples these sub matrices are swept with a window of length 6, with a sweep step equal to 1, overlapped, until to reach the 11th sample, and for training one-time sample is send at ones.

In the testing stage the backpropagation function and the loss function are suppressed, just is taken the output (the seventh state) of the neural network, to form the regression.

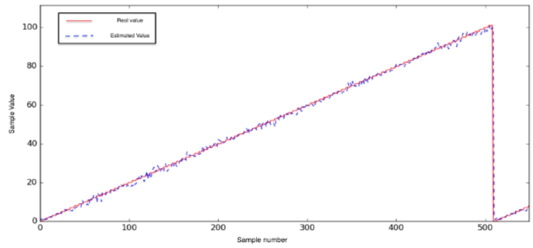

figure 12 Representation of the complete regression with LSTM method, blue line represents the estimated value, red line represents the real value.

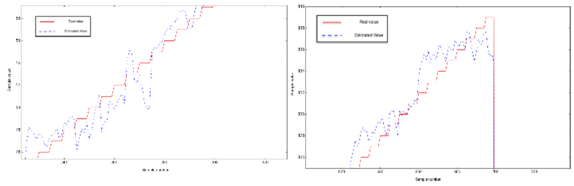

figure 13 Zoom of the figure 12, the initial state.

4.6.3 Tensorflow

All the code and the work was made with python and Tensorflow, the deep learning platform from Google, all the code for LSTM regression is upload in github https://github.com/RodrigoGantier/ball-mill-LSTM, the dataset is not proportionated.

4.7. Results

The neuronal encoding method with the signals combination of audio and vibration is use to perform the comparison with other regression methods, in the next table we can see the results.

Regression Method	RMSE sound and vibration
Random forest regression	8.170
SVM regression	16.141
LSTM network regression	**1.50**
Nearest neighbors regression	3.026
LS-SVM	2.661
MLP neural network	8.501
PLS Regression	79.776
Regression tree	3.409
adaBoost	1.870
Mean learner	40.638

Table 1 Comparison between regression models.

It is also worth noting that the regression made with the combination of the audio and vibration signals has better results than the separately performed regressions, confirming the combined signals decrease the distortions of the environment, following can see a table with these results.

Regression	RMSE sound	RMSE vibration	RMSE sound and vibration
LSTM network	**2.07**	**1.70**	**1.50**

Table 2 Comparison between single and combined regression models.

5. Conclusions

LSTM neural networks are commonly used in classification and recognition of sequences like natural language, which over time (samples) need to be remembered. The classical systems of neural networks or regression methods does not have the ability to remember patterns over time, for this reason the neural networks type LSTM have good results with data sequences as in this case, which is a system of continuous measurement through time, also be seen that the combination of the audio and vibration signals have a better regression result as compared to the separate regression of these two signals, being verified the theory that these two signals are complemented, reducing the environment distortions.

Related Links

[1] V. Monov, B. Sokolov and S. Stoenchev, "Grinding in Ball Mills: Modeling and Process Control.," *CYBERNETICS AND INFORMATION TECHNOLOGIES*, 2012.

[2] T. Ming, B. Jing and F. Yushun, "Hybrid Intelligent Modeling Approach for the Ball Mill Grinding Process".

[3] G. YAN, X. GONG, X. XU and X. HAN, "Conceptual Representation and Measurement Model0f BaU Mm Fill Level Based on Cloud Model," *Chin.Soc.for Elec.Eng*, 2014.

[4] R. Farnham and B.-M. I, "Enhanced control system for non-standard pulverized fuel mills using individual mill capability indices," *Control '96, UKACC International Conference on (Conf. Publ. No. 427)*, 1996.

[5] M. G. Melero, J. M. Cano, J. Norniella, F. Pedrayes, M. F. Cabanas, C. H. Rojas, G. Alonso, J. M. Aguado and P. Ardura, "Electric motors monitoring: an alternative to increase the efficiency of ball mills," *International Conference on Renewable Energies and Power Quality (ICREPQ'14),* 2014.

[6] Z. F and F. X.L, "Fuzzy-PID control system for level of ball mill based on level ultrasonic measurement," *Mechanical and Electrical Engineering Magazine,* 2008.

[7] B. ARUP, B. SUMAN and S. JAYA, "Designing of Intelligent Expert Control System Using Petri Net For Grinding Mill Operation," *WSEAS Transactions On Applications*, 2005.

[8] Y. Sha, Y. Chao and G. Y.G., "Analysis of acoustic signal and BP neural network-based recognition of level of coal in ball mill," *Journal of Northeastern University (Natural Science),* 2006.

[9] B. Behera, B. Mishra and C. Murty, "Experimental analysis of charge dynamics in tumbling mills by vibration signature technique," *Minerals Engineering 20,* 2007.

[10] J. Tang, L.-j. Zhao, J.-w. Zhou, H. Yue and T.-y. Chai, "Corrigendum to ''Experimental analysis of wet mill load based on vibration signals of laboratory scale ball mill shell'' [Miner. Eng. 23 (2010) 720–730]," *Minerals Engineering.*

[11] M. J. B. Z. P. Huang, "Investigation on measuring the fill level of an industrial ball mill based on the vibration characteristics of the mill shell," *Minerals Engineering.*

[12] T. C. J. C. Y.H. Sha, "Measure methods of ball mill's load," *Modern Electric Power ,* 2006.

[13] G. Yan, "Soft Sensor for Ball Mill Fill Level Based on Uncertainty Reasoning of Cloud Model".

[14] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville and Y. Bengio, "A Recurrent Latent Variable Model for Sequential Data," *arXiv:1506.02216v3 [cs.LG],* 2015.

[15] Rafal Jozefowicz, Wojciech Zaremba and Ilya Sutskever, "An Empirical Exploration of Recurrent Network Architectures," *Journal of Machine Learning Research,* 2015.

[16] W. Z. S. Vinyals, "RECURRENT NEURAL NETWORK REGULARIZATION," 2015.