A Deep Learning Approach for Condition-Based Monitoring and FaultDiagnosis of Rod Pump System

Article · June 2017

DOI: 10.29268/stsc.2017.0003

3 authors, including:

Hangqi Zhao
Rice University
16 PUBLICATIONS **617** CITATIONS
SEE PROFILE

Jian Wang
IBM
17 PUBLICATIONS **60** CITATIONS
SEE PROFILE

A DEEP LEARNING APPROACH FOR CONDITION-BASED MONITORING AND FAULT DIAGNOSIS OF ROD PUMP SYSTEM

Hangqi Zhao[1], Jian Wang[2], Peng Gao[2]
[1]Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA
[2]IBM Research – China, Beijing, China
[1]Hangqi.Zhao@rice.edu, [2]{wangwj, bjgaop}@cn.ibm.com

Abstract

Petrochemical industry is one of the key industry areas where Internet of Things (IoTs) and big data analytics could be widely applied to support smarter production and maintenance. In oil and gas exploitation, sucker-rod pumping systems are used in approximately 90 percent of artificially lifted wells. An automatic pipeline is crucial for real-time condition monitoring and fault detection of the system to save costs. Here we used convolutional neural network (CNN), a deep learning framework, to identify the working conditions of pump wells based on dynamometer cards and the corresponding sensor data. Two schemes, namely, data-based CNN and image-based CNN are proposed and compared with traditional machine learning algorithms such as k-Nearest Neighbors and Random Forests. Through experiments on a real dataset from oil fields, we show that CNN based approach could significantly outperform traditional methods without any need of manual feature engineering that requires domain expertise. Besides, we proposed a semi-automatic method for labeling big datasets of dynamometer cards, which could significantly reduce the labor work by manual labeling. Our work provides a feasible and efficient method for fault detection in oil pump systems and paves the way to applying deep learning techniques in IoTs related industries.

Keywords: Fault Diagnosis; Condition Based Monitoring; Sucker-rod Pumping Systems; Convolutional Neural Network; Deep learning; Internet of Things

1. INTRODUCTION

Oil is one of the most important natural resources in the world and needs to be artificially elevated from underground for extraction. The sucker-rod pumping system is the most widely used lift method in industrial oil productions. The real-time monitoring of the working conditions of this system is essential to the maintenance, diagnosis and optimization of the system, and currently this work is done manually by identifying the "dynamometer cards" generated from the sensors when the system is operating.

A dynamometer card is the relation curve of the load and the displacement of the sucker-rod pump recorded in a complete elevation cycle. The shape of the curve, together with the values of the curve data, is the most straightforward and effective indicator of the operation condition of pump wells. Different operation faults will be indicated by their unique features on the dynamometer cards. The identification and diagnosis of the operation condition is thus a visual interpretation process that not only demands large amount of labor and time cost but requires deep domain expertise. Moreover, massive data from sensors is generated from the daily operations of oil fields, making it more difficult for human experts to handle and analyze. Thus, an automatic and reliable approach for the pattern recognition of the dynamometer cards is of great significance for improving the diagnosing and maintenance efficiency of the system as well as preventing larger faults by early prediction.

Due to its high industrial values, various studies have been conducted on the intelligent recognition of dynamometer cards for decades. Two mostly used machine learning techniques are Artificial Neural Network (ANN) (Rogers, 1990; Nazi, 1994; Xu, 2007; de Souza, 2009; Wu, 2011) and Support Vector Machines (SVM) (Tian, 2007a, b; Li, 2013, Yu, 2013). Other methods include expert system (Martinez, 2013), clustering (Li, 2015), Extreme Learning Machine (Gao, 2015), or numerical methods based on descriptors of the dynamometer cards (Lima, 2012). Some of these studies suffered from low accuracies, efficiencies

and limited dataset size, or focused on only a few fault types. More importantly, most of these methods required hand-engineered features derived from the dynamometer cards for model training or analysis. This again requires deep domain insights and usually involves complicated algorithms or numerical computations (Yu, 2013). Moreover, the optimal choices of features or descriptors are still under research. Thus, a more general, efficient and accurate approach for automatic recognition of dynamometer cards will be a great improvement for the intelligent processing and analysis of massive sensor data as well as predictive maintenance in current oil and gas industry.

On the other hand, there have been significant and transformational developments in the Internet of Things (IoTs) (Zhang, 2012; Wu, 2014) and big data technology (Siriweera, 2015; Nural, 2015) with the massive data generated by IoT sensors and ever growing technologies in artificial intelligence. Cognitive IoT and big data analytics is taking on a critical role in driving the intelligent processing, learning and inference from big IoT datasets, making automated monitoring and control of physical systems possible through predictive modeling.

Inspired by this, in this patented work we adopted a deep learning technique, convolutional neural network (CNN), in solving this problem for the first time in the petrochemical community. We developed two types of CNN, one based on images (I-CNN) and one based on sensor data (D-CNN). For comparisons, k-Nearest Neighbors (k-NN) as a baseline method and Random Forest (RF), a widely used and effective machine learning algorithm, were also investigated. We demonstrated that both types of CNN could achieve better accuracy on our real dataset from oil field without any need of manual feature engineering, and I-CNN could give the best performance. Besides, we also proposed a procedure to semi-automatically label the dynamometer cards on large scale. This has been proven to be effective in our dataset and saved more than 60% time cost. It could provide insights on problems of dealing with unlabeled big dataset and would be particularly helpful for practical and industrial use since large labeled datasets are usually difficult and expensive to obtain. This work is an extension on our previous work (Zhao, 2016) with more detailed method introduction and experimental data.

The rest of the paper is organized as follows: In Section 2, we briefly state the problem and summarize the types of faults we investigated in this paper including their typical dynamometer cards. In Section 3.1 we introduce how our dataset was prepared and pre-processed. Section 3.2 shows how we efficiently labeled our samples in a semi-automatic way. Section 3.3 describes the predictive models (RF, k-NN, I-CNN and D-CNN) in details. Experiments and results are reported and discussed in Section 4, followed by conclusions and future work in Section 5.

2. Problem Statement

9 most commonly seen faults in sucker-rod pump systems were investigated in this paper, thus we aimed to build a 10-class classifier by treating the normal operation condition as a separate class as well. Using the sensor and configuration data from the pumping system as inputs, the classifier could automatically identify the working condition and fault types of the system. These 10 classes are “Normal Operation (NO)”, “Gas Interference (GI)”, “Fluid Shortage (FS)”, “Piston Stuck (PS)”, “Standing Valve Leakage (SVL)”, “Traveling Valve Leakage (TVL)”, “Oil Tube Leakage (OTL)”, “Down-stroke Pump Bumping (DPB)”, “Sand Production (SP)” and “Abnormal Dynamometer Cards (ADC)”. ADC includes all the cases where some sensor data is missing or wrongly recorded such that the resulting dynamometer card is abnormal.

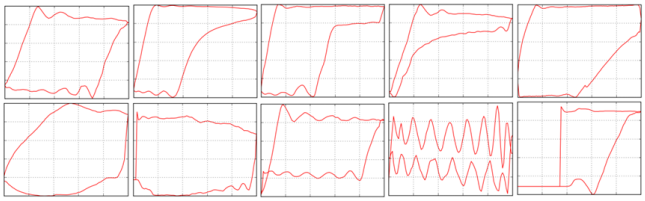


Figure 1. Typical dynamometer card samples for different working conditions (from top left to bottom right): NO, GI, FS, GS, SVL, TVL, OTL, DPB, SP, ADC). The x-axis is displacement and y-axis is the load, both are recorded from the sensors in the pumping system.

Each of these 10 classes has some particular characteristics reflected on the graph shape of dynamometer cards, and their typical examples are shown in *Figure 1*. For instance, GI is featured by an arc curve on the right bottom corner of the graph and FS is reflected by a lack of right bottom part as well but in a flatter way. Such subtle differences between classes make it more difficult to distinguish between them especially for real samples from oil fields, which are much more irregularly and strangely shaped than the typical samples shown here.

3. Proposed New Model and Method

3.1 DATA PREPARATION

We collected over 10000+ records from an oil field in China that contain the sensor data and configuration parameters for over 4500+ oil wells during year 2010 to 2015. Specifically, we collected 1 record per year for each oil well (if there are records for that year) to maximize the generality of our data such that they were not particularly

chosen for our models. The configuration parameters include the maximum stroke (*MaxS*) and minimum stroke (*MinS*), theoretical maximum load (*MaxL*) and minimum load (*MinL*) of wells. The sensor data contains six channels: Displacement (*D*), Load (*L*), Averaged current, Averaged active power, Averaged reactive power and Average power factor. Table I shows an example of these sensor data. Since the data for latter 4 parameters are missing for some of the samples, here we only used the displacement and load data for our models, which commonly forms a dynamometer card for further analysis of the working conditions of pump systems.

Table I. An example of working condition record from an oil well including six sensor channels: Displacement (D), Load (L), Averaged current (AC), Averaged active power (AAP), Averaged reactive power (ARP) and Average power factor (APF)

D	L	AC	AAP	ARP	APF
133	8115	0	0	0	0
123	8166	0	0	57521	2191
102	8166	0	1728	11960	2540
82	8090	0	1441	1644	33341
72	8038	22275	1953	5903	82
61	8038	20954	40535	1728	36127
51	8064	32202	57521	1441	0
41	8038	5084	12439	7529	31
31	7962	1989	40535	12884	31
20	7910	3924	57521	24899	31
10	7885	20316	12439	0	0
10	7834	8	40535	1469	0
10	7885	20972	520	24899	451
0	7987	10486	1498	24899	1075
0	8115	20316	1480	24899	471
0	8346	0	64881	0	0
0	8627	131	16084	0	451
0	8883	12111	44283	0	1075
10	9062	13901	52751	0	481
10	9267	32202	23429	24899	2540
10	9472	5084	5	24899	33341
20	9702	1989	1469	0	92
31	9933	33004	24899	0	3052
41	10163	59638	24899	0	10
51	10394	45883	23429	24899	31
61	10624	7872	0	24899	31
72	10880	2304	0	0	1157
82	11136	1920	0	0	35963
…	…	…	…	…	…

To remove the discrepancies between different wells, we normalized our sensor data *D* and *L* according to the configuration information of wells:

$$L' = \frac{D - MinL}{MaxL - MinL},\ D' = \frac{D - MinS}{MaxS - MinS}$$

where *L'* and *D'* are normalized load and displacement data. We then redefined the data vector of *D'* as 100 equally spaced points from 0 to 1 as the measured displacement will never exceed the maximum stroke of wells. To proceed, we split the dynamometer cards into upper curve *L'(u)* and the lower curve *L'(l)*, which represent the load data when the pumping rod is traveling up or pushed down as the rod reciprocates in the pumping well, respectively. Operation faults of various types may happen at different stages of these two processes. We then linearly interpolated *L'(u)* and *L'(l)* according to *D'*. This gives us two 100-dimension load vectors for our further training and testing of models.

3.2 SEMI-AUTO LABELING OF SAMPLES

Labeled data of high quality for machine learning could sometimes be very difficult and expensive to obtain especially in industrial applications, as is this case. The labels, namely, the working condition or the fault types of the pump wells, were missing for all the 10000+ samples we obtained, which brought tremendous difficulty for our further predictive modeling. We thus aimed to devise a scheme to semi-automatically label these samples in an efficient way and make the training and testing of our classification models possible.

There has been growing interest in the machine learning area to deal with this scenario, known as active learning (Settles, 2010) or semi-supervised learning (Zhu, 2005), both aims to address the issue of inadequate or scarce labeled data. Specifically, semi-supervised learning refers to the broad areas and methods of making use of both the labeled and unlabeled data for training, and active learning, which is a special case of semi-supervised learning, involves choosing the optimal samples from the unlabeled sample pools to label by querying the user to achieve better prediction performance. In this study, since we ultimately aim to have all the unlabeled data labeled and used in the training and testing of our predictive models instead of choosing the "best" unlabeled samples, we adopted and improved a commonly used semi-supervised technique called self-training or self-teaching.

In this approach, a classification model is built based on the small amount of available labeled data and then used to classify the remaining unlabeled data. Those unlabeled data points with high prediction confidence are then added to the labeled dataset together with their predicted labels. The classifier is then re-trained and the whole process repeated until desired amount of labeled data is obtained. We noticed that one critical drawback of this technique is that in the initial stage, the amount of labeled data is often too small compared to the unlabeled data to train a sufficiently accurate classifier, so we devised a similarity based searching approach to obtain more labeled data, which is introduced in more detail below.

Our labeling method is as *Figure 2* shows. Starting from 0 labeled samples, we firstly chose a small amount of samples, say 100, that cover typical samples for each fault type, and had them labeled by domain experts. For each of these already labeled samples, which we used as a "template", we searched samples from the unlabeled samples that are "similar" to it and labeled them as the same class as the template sample. The similarity can be defined by various metrics and here we used the Pearson correlation coefficients between the load vectors of samples. By setting a high threshold for the correlation coefficients, the newly labeled samples are actually very likely to be the same class as the template. We then manually reviewed these samples together, namely, their corresponding dynamometer cards, and removed the small fraction of samples that should not be the same class as the template, which is far more efficient than manually identifying a mixture of samples that belong to many classes. The same process could be repeated iteratively as we have more labeled samples.

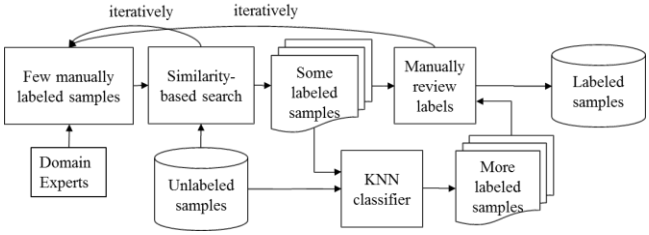


Figure 2. Schematic flow chart for semi-automatic labeling of dynamometer card samples.

After we have got enough labeled samples from the above step, we trained a k-nearest neighbors (k-NN) classifier using these samples as training data and the load (L') vector as features. The remaining unlabeled samples were then classified to some certain classes. We chose k-NN as the classifier thanks to its simplicity (a non-parametric approach) and high efficiency which are crucial in accelerating the whole labeling process. We noticed that some proportion of samples would be misclassified by k-NN, so again we manually reviewed the results, picked out those that were misclassified and adjusted their labels accordingly. In addition, a confidence level could be set in k-NN such that the classified samples have a high probability to be classified correctly. As more and more correctly labeled samples were obtained by either the first or this step, they could be reused in either finding more similar samples to them or being included in the training data of k-NN to improve the accuracy of the model. Through this whole process, we have most of the samples correctly labeled and the remaining samples that were neither identified in the similarity searching nor easily relabeled in the manually reviewing process, which only possesses a small proportion (in our experiments they are less than 10%) of the overall sample size, were then manually labeled as the last step.

It is necessary to emphasize that all the samples were manually reviewed at some stages before they were finally labeled, so the accuracy of labeling, which is crucial to our further predictions, is ensured. This step is not discussed in previous theoretical literature but is indispensable for practical use. In our scheme, this manual viewing process is significantly expedited since we "clustered" similar samples together and only need to pick out the small fraction of mislabeled samples both in the stages of similarity search and k-NN. As a result, this whole process accelerated the labeling work to a great extent compared to manually labeling one by one from a mix of all the samples. By our estimation, this labeling framework has saved over 60% of time cost. Without loss of generality, it may also be applied in other machine learning problems that deal with unlabeled dataset in industrial big data analytics.

3.3 PREDICTIVE MODELS

All the samples we collected were labeled and we ended up with 11210 samples that would be used in our predictive models. They are then randomly split into training and testing datasets with a 4:1 ratio, resulting in 8968 training samples and 2242 test samples. We investigated four different predictive models: 1) Random Forest 2) k-Nearest Neighbors 3) Data-based CNN, and 4) Image-based CNN.

3.3.1 RANDOM FOREST

Random Forest is a machine learning algorithm based on ensembles of decision trees and has been proved to be a very effective predictive model for both regression and classification problems. Similar with many other widely used algorithms such as Support Vector Machines, a good engineering of features is of great importance for better

prediction accuracy. Feature extraction of dynamometer cards have been investigated and applied in a number of previous studies, but it still remains uncertain which features are the most predictive ones.

We extracted the following five categories of features for our model training based on the normalized and interpolated dynamometer cards, which we think are the most indicative and straightforward ones to distinguish between different fault types:

a) The load data itself, namely, *L'(u)* and *L'(l)*.

b) The subtraction of *L'(u)* and *L'(l)*.

c) The number of data points of *L'(u)* and *L'(l)* in each equally spaced intervals of the longitudinal axis of the dynamometer cards, after all the data points have been projected onto that axis.

d) Key geometric parameters and statistics of the dynamometer cards. For instance, the total area enclosed by the upper and lower curves; the maxima and minima, arithmetic averages and standard deviations of *L'(u)* and *L'(l)*.

e) Other features particularly designed for certain types of faults. For instance, we counted the number of load data points larger than 0.9 due to the fact that for OTL the load values are typically apparently lower than *MaxL*. Also, we counted the number of load data points that are negative in the first 20 points of b). This is because when DPB happens the 'tail' at the lower-left part of the dynamometer cards will result in large numbers of negative values for the subtraction. These features are actually further derived from a) and b) that may identify some faults more straightforwardly.

From the five categories above we extracted 349 features in total for each sample. We then conducted feature selection according to the feature importance learned by Random Forest. We selected the top 200 important features for our model training and test. Other model parameters were optimized by cross validations.

3.3.2 K-Nearest Neighbors

We adopted k-NN, a non-parametric and simple classifier that we already used in the semi-auto labeling, as a baseline model for comparison. The same engineered features as random forests were used and k was chosen to be 5 in our model.

3.3.3 Image-based CNN

It has been well known that neural network is capable of nonlinear and multi-class classification problems in machine learning. Particularly, CNN allows weight-sharing and down-sampling at certain stages of the network and could thus vastly reduce the number of parameters, providing an efficient architecture for large scale object classifications. It was pioneered in 1998 (Lecun, 1998) and has been popularized in computer vision community since then. It has found broad applications in image recognition (Szegedy, 2014), face detection (Sun, 2013), natural language processing (Grefenstette, 2014) and other areas.

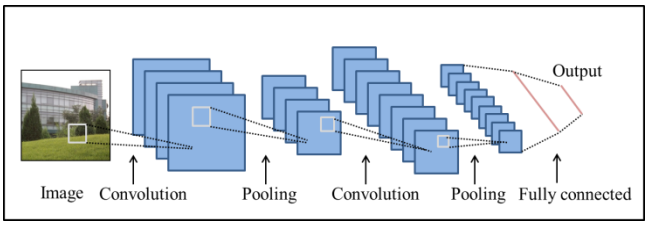


Figure 3. A typical CNN architecture consisting of input, convolution, pooling, fully connected and output layers. The feature maps are shown in blue.

A typical architecture of CNN is as *Figure 3* shows. The input layer is an image represented as pixel values. The convolutional layer (Conv) is the key building block of CNN, which computes the output of neurons connected only to small local areas of the input image by calculating the dot product between the weights and the input pixel values in those local regions. This is usually followed by a ReLU layer which applies an activation function. The spatial pooling layer basically reduces the size of weights by down-sampling along the spatial dimensions. For instance, a (2, 2) max-pooling layer takes the maxima of a 2 * 2 region in the feature maps as the input for the next layer, thus decreasing the size of parameters by 3/4. A fully connected (FC) layer is similar to normal neural network layers, so each neuron is fully connected to the previous layer. The output of FC layers is the classification result by class scores. Some of these layers could be repeated for a deeper learning of features. Besides, dropout methods are commonly used in certain layers that disable some portion of neurons to avoid over-fitting.

In this case, we firstly plotted the dynamometer card from sensor data and then filled the regions between the upper and lower curve in each dynamometer card to get a grayscale image. They were then resized to 64 by 64 images and the pixel values were extracted and used as the I-CNN inputs. This data preprocessing procedure for a randomly chosen sample is illustrated in Figure 4. The motivation of filling the region between curves is to strengthen the contrast around the curve boundary, thus probably enabling the CNN to learn and identify the shape of the curves more easily. We also noticed that the filled grayscale image could be resized to any other sizes instead of 64 by 64, but from

our experiments, images with higher resolutions than 64 by 64 do not significantly improve the prediction accuracy while require much more computational resources, so we took 64 by 64 as a compromise of model accuracy and complexity, or computational cost.

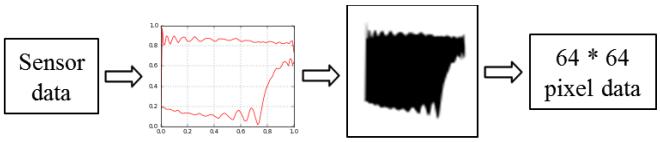


Figure 4. Data preprocessing for I-CNN

Table II. The Architecture for I-CNN.

Layer	Type	Feature Maps	Filter Size	Dropout
0	Image	64 ×64	-	-
1	Conv	60 ×60 ×32	5 ×5	0
2	Conv	56 ×56 ×32	5 ×5	0
3	Pool	28 ×28 ×32	2 ×2 max	-
4	Conv	24 ×24 ×64	5 ×5	0
5	Conv	20 ×20 ×64	5 ×5	0
6	Pool	10 ×10 ×64	2 ×2 max	-
7	FC	1 ×1 ×256	-	0.5
8	Output	1 ×10	-	0.5

The architecture and parameters of our I-CNN network is as *Table II* shows. A stride of 1 and 0 padding was used in the convolutional layers. We implemented and trained CNN in Lasagne, an open source Python library for deep neural networks based on Theano. In the learning process, the weights were initialized as a standard Gaussian distribution. They were optimized in the training process in terms of cross-entropy loss by stochastic gradient descent. The learning rate was set to be gradually decaying during the training. A softmax was applied in the output layer to get the probabilities of each class that sum to 1 and the predicted class was taken as the one with the highest probability.

3.3.4 DATA-BASED CNN

Although CNN has been well known for its broad applications in image recognitions, it is capable of dealing with non-image inputs as well in a similar way. While random forest requires feature design from the dynamometer cards, in our D-CNN the data forms the curves, namely, the load vector itself was directly used as input of the network. This is because CNN is designed to mimic the organization of neurons in human brain and could derive and learn the features for the particular prediction task by itself.

We combined the load data *L'(u)* and *L'(l)* and ended up with a vector of dimension 200 as the input for each sample. We adopted a slightly deeper network than I-CNN considering the difficulty of directly learning from the load data. The architecture for D-CNN consisting of 14 layers is as *Table III* shows, and a similar learning process was applied as the I-CNN. Despite deeper network, D-CNN actually resulted in less training time than I-CNN due to smaller parameter size.

Table III. The Architecture for D-CNN.

Layer	Type	Feature Maps	Filter Size	Dropout
0	Data	1 ×200	-	-
1	Conv	1 ×198 ×16	1 ×3	0.1
2	Conv	1 ×198 ×16	1 ×1	0.1
3	Pool	1 ×99 ×16	1 ×2 max	-
4	Conv	1 ×97 ×32	1 ×3	0.2
5	Conv	1 ×97 ×32	1 ×3	0.2
6	Pool	1 ×49 ×32	1 ×2 max	-
7	Conv	1 ×47 ×64	1 ×3	0.2
8	Conv	1 ×47 ×64	1 ×1	0.2
9	Pool	1 ×24 ×64	1 ×2 max	-
10	Conv	1 ×22 ×128	1 ×3	0.2
11	Conv	1 ×22 ×128	1 ×1	0.2
12	FC	1 ×1 ×200	-	0.5
13	FC	1 ×1 ×200	-	0.5
14	Output	1 ×10	-	-

4. Results And Discussions

We firstly visualized the dataset to better understand the structures of the data. In *Figure 5(a)*, the distribution of various fault types was plotted in terms of counts of samples, showing a highly imbalanced dataset which is commonly seen in fault analysis problems. We also applied t-Distributed Stochastic Neighbor Embedding (t-SNE) technique, a dimensionality reduction method developed in recent years (Maaten, 2008) to visualize the data. t-SNE could embed high-dimensional data into a space of two or three dimensions in a way such that similar objects in the original high dimension space will end up close to each other in the lower dimension space. A scatter plot after the sample data was embedded in 2-D space is shown in *Figure 5(b)*. 5000 samples were shown in the plot and the load vector of dimension 200 (*L'(u)* and *L'(l)* combined) was used as features when applying t-SNE. Each data point in the figure represented a corresponding dynamometer card and was labeled and colored in the same way as in *Figure 4(a)*.

As we can see from *Figure 5(b)*, samples in class 0 (NO), 5 (TVL) and 9 (ADC) were represented as relatively independent clusters, meaning that these classes might be easier to be recognized and classified. Conversely, samples of other fault types were entangled and mixed with each other, which might be difficult to distinguish. For instance, class 1 (GI, shown in blue) and class 2 (FS, shown in green) are hardly separable in the plot due to the high similarity of their corresponding dynamometer cards. This was also shown in our classification results in later discussions.

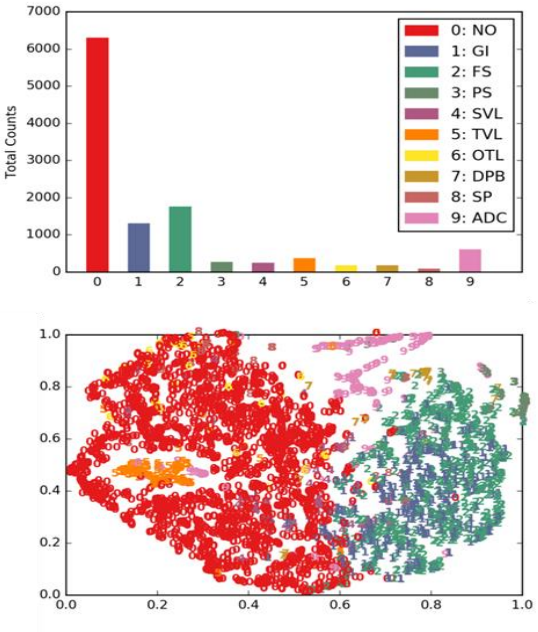


Figure 5. Visualization of the dataset. (up) Distribution of fault types by sample counts. (bottom) t-SNE embedding in 2-D space.

We then made predictions on the same test dataset using the four models described in the method section. The overall accuracy rate, defined by the total number of correctly classified samples divided by the total number of all samples, is summarized in *Table IV*. As we can see, k-NN gives the lowest prediction accuracy as expected. Also, both D-CNN and I-CNN achieve better accuracy than RF, despite the fact that CNN use no designed features while RF includes more than 300 particularly engineered features.

Table IV. Overall Prediction Accuracy of the Models.

Method	*k*-NN	RF	D-CNN	I-CNN
Accuracy	82.4	90.9	92.0	93.5

Since the dataset is highly imbalanced, other metrics must be considered as well to better evaluate the performances of models. In *Table V* we summarized the precision, recall and F1-score of each class for RF, D-CNN and I-CNN. In multi-class classification problems, for each class the precision is defined as TP / (TP + FP) and the recall is defined as TP / (TP + FN), where TP, FP and FN denote the true positive, false positive and false negative for that class, respectively. F1-score is the harmonic mean of precision and recall which combines both metrics. In the table we also included the proportion of samples each class takes in the whole dataset. We noticed that k-NN is much less accurate so it was not included in the comparison. It is clear from *Table V* that generally speaking, we could rank these three models as I-CNN > D-CNN > RF in terms of all these three metrics especially the F1-score. Notably, RF behaved poorly on the recall of SVL (0.222) and SP (0.200) due to the low proportions of these two classes, but I-CNN could still give decent results on them (0.622 and 0.800) as well as other classes with very low frequencies. This indicates that I-CNN could learn useful features very effectively from very limited number of samples.

The confusion matrix obtained from the prediction of the test dataset by I-CNN is shown in *Figure 6*. The numbers on the diagonal indicate the samples that were correctly classified and those off-diagonal are the misclassified samples. We clearly see that a very large proportion of samples sit on the diagonal, especially for Class 0, 1, 2 and Class 9 indicated by their colors. A direct insight from the matrix is that class 1 (GI) and 2 (FS) are the two classes that are most difficult to distinguish from each other, since as many as 22 GI samples were misclassified as FS and 23 vice versa. This is in accordance with the fact that there is

only slight difference on their dynamometer cards as discussed in Section 2 as well as the t-SNE visualization result. Some other classes, on the other hand, are most likely to be misclassified as Class 0 (NO), including Class 4 (SVL), Class 5 (TVL), Class 6 (OTL) and Class 8 (SP).

Even though a high accuracy of 93.5% was achieved, we further analyzed the performance of I-CNN by looking at the misclassified samples and the prediction results from a probability perspective. We conclude that some of the misclassifications are due to the ambiguity of the labels. Some samples could be very difficult to determine a clear fault type even for human experts or contains more than one

Table V. Precision, Recall and F1-score of Predictions for RF, D-CNN and I-CNN.

		Precision			Recall			F1-score		
Class	Freq	RF	D-CNN	I-CNN	RF	D-CNN	I-CNN	RF	D-CNN	I-CNN
NO	0.565	0.946	0.953	0.967	0.981	0.967	0.980	0.963	0.960	**0.973**
GI	0.110	0.779	0.810	0.835	0.789	0.817	0.846	0.784	0.813	**0.840**
FS	0.162	0.866	0.879	0.922	0.874	0.923	0.912	0.870	0.900	**0.917**
PS	0.025	0.864	0.926	0.893	0.895	0.877	0.877	0.879	0.901	**0.885**
SVL	0.020	0.769	0.846	0.824	**0.222**	**0.244**	0.622	**0.345**	**0.379**	**0.709**
TVL	0.032	0.969	0.917	0.970	0.875	0.917	0.903	0.920	0.917	**0.935**
OTL	0.012	0.720	0.667	0.720	0.692	**0.385**	0.692	0.706	**0.488**	**0.706**
DPB	0.015	0.935	0.853	0.914	0.879	0.879	0.970	0.906	0.866	**0.941**
SP	0.007	1.000	0.541	0.706	**0.200**	0.867	0.800	**0.333**	0.666	**0.750**
ADC	0.053	0.915	0.947	0.965	0.907	0.915	0.941	0.911	0.931	**0.953**

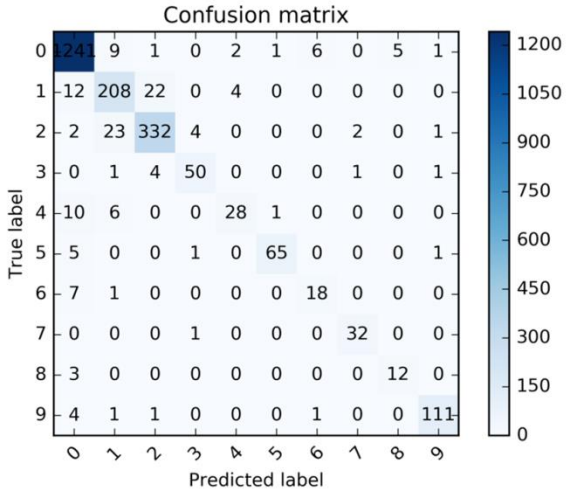


Figure 6. The confusion matrix for the prediction by I-CNN. The row numbers are the actual labels of the test samples while the column numbers are the predicted labels.

fault types simultaneously, thus bringing some label noises in the whole datasets. This is further demonstrated in *Table VI* where we showed the top k (k = 1, 2, 3) overall accuracy by I-CNN. The top k accuracy is defined as the prediction of a class as one of the top k classes by the CNN. In this case we see that 98.6% of the samples were correctly predicted in the top 2 probabilities, which indicates that most of the misclassified samples were predicted with the second highest probability.

Table VI. Top k (k = 1, 2, 3) Overall Accuracy by I-CNN.

	Top 1	Top 2	Top 3
Accuracy (%)	93.5	98.6	99.6

We noticed that an important factor that limits the prediction accuracy of our models is the imbalance of the data between classes. For instance, Class 0 (NO) possesses more than 50% of the whole dataset while 6 classes (PS, SVL, TVL, OTL, DPB and SP) accounts for less than 5% of it, resulting in much lower recall for these minor classes as shown in Table V. The most commonly used technique to counter the effect of imbalance is resampling the training set, either by down-sampling the major class or over-sampling the minor classes to balance the sizes of all classes. Here we over-sampled the minor classes considering the limited sizes of them in our dataset. Instead of simply randomly replicating the minor classes, we applied a more advanced resampling approach, Synthetic Minority Over-sampling Technique (SMOTE). The idea of SMOTE is creating "synthetic" training samples from real ones by performing certain operations in the feature space of the original data. For

a more detailed introduction to SMOTE we refer to (Chawla, 2002).

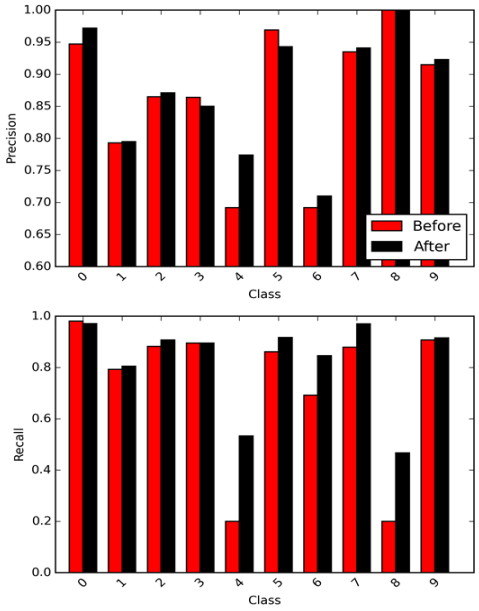


Figure 7. The Precision (up) and Recall (bottom) of each class for the prediction by RF before (red) and after (black) the over-sampling of the training samples.

Taking Class 0 (NO) as the major class, we over-sampled the remaining 9 classes such that the size ratio of Class 0 and each of the remaining class is 1: 0.8. This resulted in a total of 36784 training samples obtained from the original 8092 training samples. The test dataset remains the same as the over-sampling was only taken on the training set. Considering the model training time, we only investigated the effect of over-sampling on RF here but the same method could be applied to CNN and any other model as well. *Figure 7* summarizes the precision and recall of each class for the prediction by RF before and after using the over-sampled training dataset. As we see, generally both the precision and recall are improved by over-sampling for most of the classes. Notably, the recall of class 4 (SVL) and class 8 (SP) are improved by more than two times, both from 0.2 to around 0.5. Our result also shows that the overall prediction accuracy is increased from 90.9% to 92.2% for RF by the use of over-sampling and becomes comparable with D-CNN. This demonstrates the effectiveness of over-sampling technique especially SMOTE in addressing the imbalance problem in fault detection applications. On the other hand, the prediction performance by RF still could not outperform I-CNN even with over-sampling, both shown in the overall accuracy, precision and recall. Naturally, we could also expect a better performance for I-CNN if over-sampling is applied, which would be examined in our future work.

As for training efficiency, due to the process of learning features by itself, CNN may need more training time than traditional machine learning methods, depending on the complexity of the network and the hardware configuration of the training platform. However, once the network is trained, it could be restored and repeatedly used for model predictions. Moreover, as the rapid improvements on hardware support such as Graphic Processing Units (GPUs), deep learning based approach is becoming more efficient, feasible and reliable for industrial applications.

5. CONCLUSIONS

We have investigated and developed convolutional neural network based approaches for fault diagnosis of sucker-rod pumping systems by the automatic recognition of dynamometer cards. By comparisons with traditional machine learning algorithms such as Random Forest, we demonstrated that both the sensor data based CNN and image based CNN could outperform traditional methods without manually designed features. Specifically, I-CNN achieved 93.5% accuracy in our datasets from real oil field that covers 9 major faults in oil production industry. The technique could be generalized to all configurations of pump wells and other fault types as well as long as necessary data is provided. This is the first time that deep learning techniques have been applied in this problem, which could substantially reduce the labor costs and make an automatic and reliable diagnosis pipeline possible. Besides, we developed a scheme for labeling dynamometer cards in large scale semi-automatically and it has been proven to reduce the time cost by about 60%. This is particularly useful for obtaining high quality data from big datasets for future learning when the labels are missing and expensive to be identified manually.

For future work, it would be interesting to make the same comparisons if more channels of sensor data is involved. As mentioned in the method section, there are various sensors apart from the displacement and load sensors embedded in the rod pump system such as temperature, power or current sensors. These extra features together with the dynamometer cards may improve the prediction performance further. For instance, data based CNN would be expected to work better since more dimensions of sensor data is

incorporated rather than a simple 1-D load data. Moreover, if the relations between sensors have physical meanings and are indicative of the working condition such as the dynamometer cards, image based CNN could be applied by creating and identifying more relation graphs between different features. Another approach that could potentially further improve the prediction performance, especially for practical use, is to ensemble more than one predictive models, which has been proven in many cases to outperform any single model. For instance, a simple strategy here could be taking the majority vote on the predictions obtained by RF, D-CNN and I-CNN. In the end, the deep learning framework we developed in this paper is not limited in the intelligent fault detection of oil pumping systems, but could be widely applied in cognitive technologies and big data analytics of other industries that incorporate Internet of Things.

6. ACKNOWLEDGMENT

The work is funded by IBM Great Minds Program which is one of summer intern programs in IBM. We thank Jijiang Song (IBM GBS) for providing important experiment data, business requirements and insightful domain knowledge. We also thank Shao Chun Li (IBM Research) for their great support and help on this research.

7. REFERENCE

Rogers, J. D., Guffey, C. G., & Oldham, W. J. B. (1990). Artificial neural networks for identification of beam pump dynamometer load cards, *SPE Annual Technical Conference and Exhibition*, New Orleans, 23-26 September, 1990.

Nazi, G. M., Ashenayi, K., Lea, J. F., & Kemp, F. (1994). Application of artificial neural network to pump card diagnosis, *SPE Computer Application*, v(6), pp. 9-14.

Xu P, Xu SJ, Yin HW. (2007). Application of self-organizing competitive neural network in fault diagnosis of sucker rod pumping system. *Journal of Petroleum Science and Engineering*, v(58), pp. 43–48.

Tian, J., Gao, M., Li, K., & Zhou, H. (2007). Fault detection of oil pump based on classify support vector machine, *Proceedings of the international conference on control and automation*, Guangzhou China, 30 May-1 June, 2007, pp. 549-553.

Tian, J., Gao, M., Liu, Y., Zhou, H., & Li, K. (2007). The fault diagnosis system with self-repair function for screw oil pump based on support vector machine, *Proceedings of the international conference on robotics and biomimetics*, Sanya China, December, 2007, pp. 2144-2148.

De Souza, A., Bezerra, M., Barreto Filho, M. D. A., & Schnitman, L. (2009). Using artificial neural networks for pattern recognition of downhole dynamometer card in oil rod pump system, *Proceedings of the 8th WSEAS international conference on artificial intelligence, knowledge engineering and data bases*, Cambridge, February 2009, pp. 230-235.

Wu W., Sun W. L., Wei H. X. (2011). A fault diagnosis of sucker rod pumping system based on wavelet packet and RBF network, *Advanced Materials Research*, v(189), pp. 2665-2669.

Li, K., Gao, X., Tian, Z., & Qiu, Z. (2013). Using the curve moment and the PSO-SVM method to diagnose downhole conditions of a sucker rod pumping unit. *Petroleum Science*, v(10), pp. 73-80.

Yu, D., Zhang, Y., Bian, H., Wang, X., & Qi, W. (2013). A new diagnostic method for identifying working conditions of submersible reciprocating pumping systems. *Petroleum Science*, 10(1), pp. 81-90.

Martinez, E. R., Moreno, W. J., Castillo, V. J., & Moreno, J. A. (1993). Rod pumping expert system. *SPE petroleum computer conference*, New Orleans, 11-14 July 1993.

Liang, X.; Li, W.; Zhang, Y. & Zhou, M. (2015). An adaptive particle swarm optimization method based on clustering, *Soft Computing*, v(19), pp. 431-448

Gao, Q., Sun, S., & Liu, J. (2015). Working Condition Detection of Suck Rod Pumping System via Extreme Learning Machine, *2nd International Conference on Civil, Materials and Environmental Sciences (CMES)*.

de Lima, F. S., Silva, D. R., & Guedes, L. A. (2012). *Comparison of Border Descriptors and Pattern Recognition Techniques Applied to Detection and Diagnose of Faults on Sucker-Rod Pumping System*, INTECH Open Access Publisher.

Yu, Y., Shi, H., & Mi, L. (2013). Research on feature extraction of indicator card data for sucker-rod pump working condition diagnosis, *Journal of Control Science and Engineering*, v(2013), no. 6.

Wu, Q., Ding, G., Xu, Y., Feng, S., Du, Z., Wang, J. and Long, K. (2014). Cognitive internet of things: a new paradigm beyond connection. *IEEE Internet of Things Journal*, 1(2), pp.129-143.

Zhang, M., Zhao, H., Zheng, R., Wu, Q., & Wei, W. (2012). Cognitive internet of things: Concepts and application example. *International Journal of Computer Science*, 9(6), pp.151-158.

Siriweera, T. H. A. S., Paik, I., Kumara, B. T., & Koswatta, K. R. C. (2015). Intelligent Big Data Analysis Architecture Based on Automatic Service Composition. *IEEE International Congress on Big Data*, pp. 276-280.

Nural, M. V., Cotterell, M. E., Peng, H., Xie, R., Ma, P., & Miller, J. A. (2015). Automated Predictive Big Data Analytics Using Ontology Based Semantics, *International Journal of Big Data*, v(2), pp. 43-56.

Zhao, H., Wang, J., Gao, P. (2016). Cognitive IoT: A Case Study of Applying Deep Learning Approach in Fault Diagnosis of Sucker-Rod Pumping Systems, *Proceedings of S2 International Conference on Internet of Things*, Zhangjiajie China, 2016

Settles, B. (2010). Active learning literature survey, *Computer Sciences Technical Report 1648*, University of Wisconsin, Madison.

Zhu, X. (2005). Semi-supervised learning literature survey, *Computer Sciences Technical Report 1530*, University of Wisconsin, Madison.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE 86*, v(11), pp. 2278–2324.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going Deeper

with Convolutions, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.

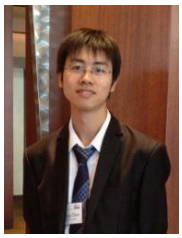
Sun, Y., Wang, X., & Tang, X. (2013). Deep Convolutional Network Cascade for Facial Point Detection, *The IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3476-3483

Blunsom, P., de Freitas, N., Grefenstette, E., & Hermann, K. M. (2014). A Deep Architecture for Semantic Parsing, *Proceedings of the ACL 2014 Workshop on Semantic Parsing*.

Maaten, L. V. D., & Hinton, G. (2008). Visualizing Data Using t-SNE, *Journal of Machine Learning Research*, v(9), pp. 2579-2605.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, v(16), pp. 321-357.

Authors



Hangqi Zhao is a graduate student in the Department of Electrical and Computer Engineering in Rice University, USA. He obtained his B.S. degree in Information Engineering from Zhejiang University in 2013 and M.S. degree in Electrical and Computer Engineering in Rice University in 2015, and he is now pursuing his Ph.D. degree. He has been interested in research on model reduction of large systems, computational electromagnetics and data mining. He was a research intern in IBM Research – China under IBM Great Minds Global Internship Program in 2016. He is an author/co-author of 8 publications and 2 patents.

Dr. Wang Jian is a Research Staff Member in the Internet of Things and Services Research department at IBM Research - China. He received his B.S. and M.S. degrees in Aircraft Manufacturing Engineering from the Northwestern Polytechnic University in 1994 and 1997, received Ph.D. degree in Aeronautics and Astronautics Manufacturing Engineering from the Northwestern Polytechnic University in 2000, respectively. He subsequently joined IBM Research - China, where he has worked on e-commerce, model-driven business transformation, immersive web, internet of things, cloud service scalability research. In 2013, he received the title of IBM Master Inventor. He is the author or coauthor of 36 patents and 14 technical papers.

Dr. Gao Peng is a Research Staff Member of IBM Research - China. He received Ph.D. from Tongji University in 2010, and joined IBM subsequently. His research interests include: Deep learning, Complex Science, Intelligent Transportation System, Multi-agent System & Simulation. With over 11+ publications and 25+ patents, he designed and developed many simulation / decision support systems, which played a big role in production environment. E.g. Decision Support System for Operation of Urban Mass Transit (provide decision support for transport plans of Olympics 2008); Safety Assessment and Venue Flow Management Planning for the Main Entrance of Shanghai Expo 2010; Decision Support System for Daily Locomotive Scheduling; Planning Support System for Urban Transit Networks etc.