Multi-target Regression Approach for Predictive Maintenance in Oil Refineries Using Deep Learning

Helmi Helmiriawan and Zaid Al-Ars

Abstract—Modern oil refineries typically use a high number of sensors that generate a massive amount of data about various process variables in the infrastructure. This data can be used to perform predictive maintenance, an approach to predict impending equipment failures and mitigate downtime in refineries. This paper presents the use of multi-target regression approach for predictive maintenance. Multi-target regression is a modeling approach that aims to predict multiple targets simultaneously. The relationships between multiple process variables are modeled using deep learning methods, while the model error is evaluated using cumulative sum method to detect faults that might potentially become failures. Unlike many existing solutions, our approach does not rely on the availability of data that captures the presence of faults in the plant. The proposed approach is demonstrated using real industrial data from a crude distiller in Shell Pernis. The results show a speed-up in modeling time by 16x and an improved early fault detection time by 1.2x as compared with the single-regression approach. Furthermore, the proposed approach is also able to isolate the faults by producing higher errors in predicting faulty equipment compared with healthy equipment.

Keywords—cumulative sum, deep learning, machine learning, multi-target regression, predictive maintenance.

I. Introduction

An oil refinery is a group of manufacturing plants that converts crude oil into more useful products, such as gasoline, diesel, and kerosene [1]. It is typically large and complex, containing many different plants and types of equipment. As an example, Shell Pernis, the biggest refinery in Europe, has 60 different plants and is almost as large as 1,000 football fields. Safety is an important aspect of refinery operation. Therefore, a large number of sensors is used for monitoring a variety of process variables in the refinery, such as pressure and temperature.

Typically, the equipment in a plant is kept in good condition by performing regular maintenance activities. However, maintenance requires downtime and incurs high costs because it interrupts on-going operational activities. Traditionally, maintenance can be classified as reactive maintenance, where equipment is replaced after the presence of faults, and preventive maintenance, where equipment is inspected or replaced prematurely without any fault. Predictive maintenance is a modern approach that uses information about current equipment conditions in order to make correct maintenance decisions for maximizing plant availability [2]. It aims to minimize unscheduled downtime due to unforeseen faults and scheduled downtime as recommended by equipment manufacturers or domain experts. Despite its benefit, specific knowledge about the underlying processes in plants is required to perform predictive maintenance. Manually analyzing a huge amount of data from a large number of sensors by domain experts is challenging, time-consuming, and not scalable. Therefore, a number of studies have been performed to automate the process of analyzing sensor data using various methods.

Fault detection and isolation are part of predictive maintenance. A fault can be defined as an undesired state of a system or equipment that deteriorates the performance of the system or equipment [3]. In extreme cases, a fault can degrade the state of the equipment to such a degree that it causes an externally detectable failure. This is usually the result of the progression of the fault over time [4]. Therefore, it is crucial that a fault can be detected as soon as possible before it evolves into a failure. Meanwhile, fault isolation can be described as a process of finding the location of faults in the plant.

Machine learning enables computers to learn certain tasks indirectly by extracting patterns from data using specialized algorithms without having to directly program the exact rules for the target task [5]. It can be used to model a variety of processes in a plant by learning patterns in sensor data. However, most classical machine learning methods still rely on feature engineering [6], [7], which can be described as the process of extracting representations or features from raw data for machine learning algorithms. Deep learning is a family of methods in machine learning, which has the ability to learn the required features from data automatically [8]. Compared to other machine learning methods, deep learning requires less human intervention in feature engineering. Recent studies show that, by using deep learning, it is possible to make computers outperform humans in several tasks, such as recognizing traffic signs [9] or playing the game of Go [10].

A typical plant in a refinery uses hundreds or thousands of control valves to make sure various processes can be performed [11], [12]. As a result, control valves represent one of the most critical equipment in refineries. At the same time, modeling all these valves individually is an expensive, labour-intensive process. Therefore, it is desired that a single model created using machine learning methods can be used to monitor a large number of control valves. Furthermore, it is also desired that when a fault is detected, the location of the fault can also be identified, so-called fault isolation. In this paper, a multi-target regression approach using deep learning methods is proposed to predict the conditions of several control valves in a plant. Multi-target or multi-output regression is an approach that aims to simultaneously predict a set of multiple numerical values given a set of inputs [13]. It can be used to increase the efficiency and scalability of the predictive maintenance approach in large refineries like Shell Pernis.

Data that represent all possible faulty conditions is difficult

H. Helmiriawan and Z. Al-Ars are with the Quantum & Computer Engineering Department, Delft University of Technology, Delft, 2628 CD Delft, The Netherlands. Emails: {h.helmiriawan, z.al-ars}@tudelft.nl.

to obtain, while simulated data may not be realistic. Therefore, real industrial data is used in this paper, where the models are trained only with data that represent normal conditions of the plant. Two variants of deep learning methods, recurrent neural networks (RNNs) and autoencoders, are proposed in this paper. The error between model prediction and measured data from sensors is then evaluated by using cumulative sum (CUSUM) [14] method to determine the condition of control valves. It is desired that during normal conditions, the prediction error is close to zero, while during the presence of faults, the prediction error would deviate strongly from zero.

The contributions of this paper are as follows.

1) Modeling normal condition of a plant in a refinery using deep learning methods through the multi-target regression approach.
2) Identifying the faulty control valve by evaluating the prediction error using the CUSUM method.
3) Validating the proposed approach with real industrial data.

The remainder of this paper is organized as follows. Existing solutions using machine learning methods are discussed in Section II. In Section III, the details of the proposed deep learning and CUSUM methods are presented. The case study and the used dataset are described in Section IV. The experimental results and analysis are presented in Section V. Finally, the summary and conclusions are stated in Section VI.

II. Related Work

A. Machine Learning Methods

Machine learning methods have been proposed in several studies for detecting faults in a number of complex systems [15]–[17]. However, these studies still utilize feature engineering methods to make machine learning models work well in detecting faults. Yan [15] used principal component analysis (PCA) for feature engineering and random forests for classifying aircraft engines conditions. He et al. [16] also used PCA for feature engineering, while support vector machines (SVMs) was used for creating models to detect faults in semiconductor etching systems. Meanwhile, Li et al. [17] used wavelet transform methods to process the raw data before using PCA for generating models to detect faults in building automation systems. These methods use classical machine learning algorithms that require manual feature engineering.

Deep learning has gained popularity lately over other machine learning methods because of its capability to automatically learn useful features from data. Recent studies have demonstrated the capability of deep learning to detect faults in a variety of use cases. Tamilselvan et al. [18] used deep belief networks (DBNs) for classifying aircraft engines and power transformers conditions. Jia et al. [19] used stacked autoencoders trained with backpropagation algorithm for classifying rotating machinery conditions, while Qi et al. [20] used stacked sparse autoencoders trained with the same algorithm for a similar task. In addition, Haidong et al. [21] used deep autoencoders trained with artificial fish swarm algorithm for classifying rotating machinery conditions. To accelerate training and improve the precision of DBNs models, Tang et al. [22] used Nesterov momentum method in model training for classifying rotating machinery conditions. In contrast, Tran et al. [23] used Teager-Kaiser energy operator method for feature engineering and DBNs for creating models to classify different reciprocating compressors conditions, ignoring the capability of deep learning to learn required features from raw data automatically. However, these studies still need data that represent normal and abnormal conditions to train machine learning models, which is typically hard to obtain. Furthermore, these studies have not explored the ability of machine learning methods for fault isolation, which is more challenging than fault detection [4].

B. Multi-target Regression Approach

Machine learning methods have also been proposed for detecting faulty control valves in several studies. Yang et al. [24] used data from normal and abnormal conditions to train SVMs models for classifying valves conditions. Karpenko et al. [25] also used data from normal and abnormal conditions, while using deep neural networks for classifying valves conditions. Meanwhile, Suursalu [26] only used data from normal conditions and explored several machine learning methods for detecting faulty valves in a crude distiller. The results show that RNNs provide better results compared with ridge regression and deep neural networks (without recurrent connections). However, this study only used models with a single target to predict multiple control valves, which have disadvantages over the multi-target regression approach proposed in this paper.

The multi-target regression approach has been explored in a few studies to detect faults in complex systems. Timusk et al. [27] explored various feature extraction methods for feature engineering and several machine learning methods, including PCA and autoencoders, for creating models to detect faults in excavator machinery. Only data from normal conditions was used in model training, while prediction errors were used to indicate the presence of faults in the system. However, this study still utilized feature engineering methods and only used simulated data in evaluating the models. Talebi et al. [28] also used the prediction error to indicate the presence of faults in satellite's attitude control systems. This study has shown that multi-target regression approach using RNNs can be used to detect and isolate faults in complex systems. The results show that when a fault occurs in one of the equipment in the system, only the prediction of that specific equipment produces a high error. However, this study also only used simulated data. Meanwhile, our approach differs by using noisy realistic data and using CUSUM method to handle the noise.

C. Evaluation Metrics

Typically, the average prediction error of all outputs is used to evaluate multi-output models [13]. However, our paper is focused on detecting and isolating faults in complex systems; thus evaluating each output individually is preferred. In order to isolate faults in specific output, mean squared error (MSE) is the most commonly used error metric in the regression setting [29]. However, MSE is sensitive to noise, hence it is not suitable for noisy industrial data that is used in this

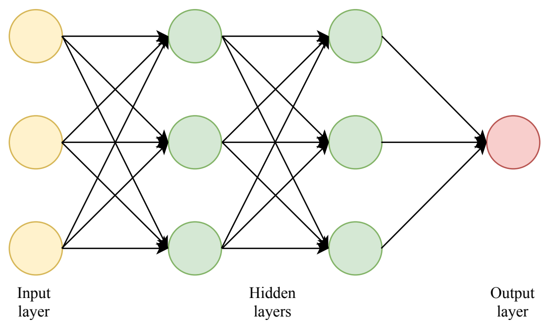


Fig. 1. An example of deep neural networks architecture

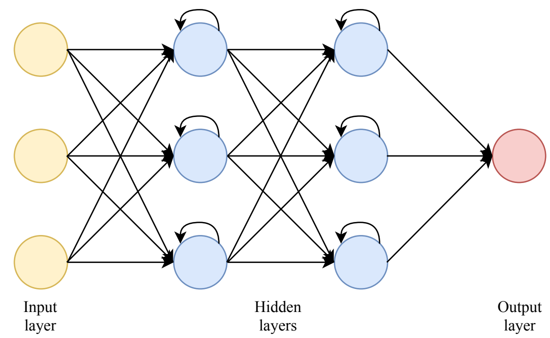


Fig. 2. An example of recurrent neural networks architecture

paper. CUSUM is an alternative method that can be used since it calculates the cumulative sum of sequential data, thus eliminating small noise over time. Goh et al. [30] used CUSUM to reduce the number of false positives in detecting anomalies in a water treatment test bed. In addition, Leger et al. [3] also used CUSUM to detect small deviations in the process variables of a nuclear reactor test bed.

III. Methodology

A. Deep Learning

Deep learning can be described as the application of neural networks, a family of methods in machine learning, with multiple layers where the outputs of each layer are the inputs of the next layer. Unlike other machine learning methods, deep learning methods are not only useful for learning complex models but also for learning features automatically. Typically, manually designing features in machine learning is challenging and time-consuming. The ability of deep learning to learn features automatically can be obtained by composing multiple functions, where each function transforms a representation at one level into another representation at a higher level. For example, a chain of functions made from three functions can be formulated as $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$ [7]. In this example, x is the original set of input features and $f(x)$ is the output of the model. Meanwhile, $f^{(1)}$ is the *first layer*, $f^{(2)}$ is the *second layer*, and so on. The first and second layer are called *hidden layers*, while the last layer is called *output layer*. In the case of learning image data, deep neural networks might learn the presence of edges in the first layer, the presence of corners in the second layer, and the presence of objects in the last layer.

Neural networks are a family of machine learning methods that are inspired by the networks of brains [31]. Like brains, neural networks are made of multiple computing cells called *neurons*. Mathematically, a neuron in neural networks can be defined as:

$$y = g\left(b + \sum_{j=1}^{m} v_j w_j\right) \tag{1}$$

where $v_1, v_2, ..., v_m$ are the inputs, $w_1, w_2, ..., w_m$ are the input weights, b is the bias, g is the activation function, and y is the output of the neuron. In the case of neural networks models, the weights and bias of neurons are the model parameters. Hyperbolic tangent activation function is used in this paper to compute the output of the neuron, which is defined as:

$$g(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \tag{2}$$

When deep neural networks receive an input v, the information flows forward through the networks and produce outputs in the last layer. This process is called *forward propagation*. Furthermore, each layer in deep neural networks can be made of multiple neurons that act in parallel, as can be seen in Figure 1.

Recurrent neural networks (RNNs) are a type of neural networks that use recurrent connections in the networks [32]. It is suitable to process sequential data, such as speech, text, and time-series data. In sequential data, it is assumed that each data point is related to the previous data points. Therefore, when processing a set of inputs at state t, the inputs of the previous states should be considered. In RNNs, the recurrent connections are used to include the inputs of the previous states to produce the outputs of the networks, as depicted in Figure 2. With a sequence of values $x^{(1)}, x^{(2)}, ..., x^{(t)}$ as the inputs, the computation in the hidden layers of RNNs can be defined as [7]:

$$s^{(t)} = f\left(s^{(t-1)}, x^{(t)}; \theta\right) \tag{3}$$

where $s^{(t)}$ is the outputs at state t, $s^{(t-1)}$ is the outputs of the previous state, and $x^{(t)}$ is the inputs at state t.

Autoencoders are neural networks that are designed to convert their inputs into a *code* and then reconstruct the inputs back from the code [33]. The results from the reconstruction are the outputs of the networks, thus the number of neurons in the input layer of autoencoders is always the same as the output layer, as depicted in Figure 3. Therefore, this type of neural networks can be seen as consisting of two components: an encoder, which encodes the inputs to the code, and a decoder, which reconstructs the inputs from the code. The prediction error of autoencoders is evaluated by using a cost function defined as:

$$J(\theta_e, \theta_d) = \sum_{k=1}^{n} \left(x_k - d\left(e(x_k; \theta_e); \theta_d\right)\right) \tag{4}$$

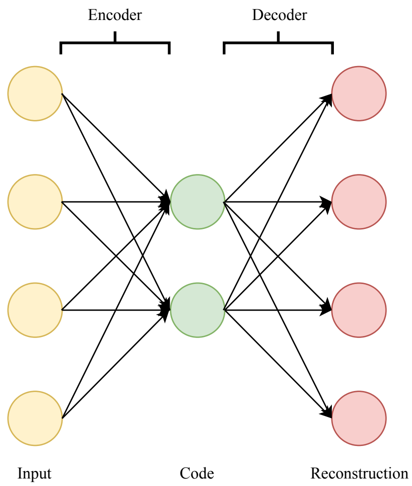


Fig. 3. An example of autoencoder architecture

where $x_1, x_2, ..., x_n$ are the model inputs, e is the encoder function, θ_e is the encoder parameters, d is the decoder function, and θ_d is the decoder parameters.

B. Cumulative Sum

Cumulative sum (CUSUM) is a sequential analysis method that is suitable to detect changes in sequential data. Unlike simulated or artificial data, real industrial data often contains noise. CUSUM calculates the cumulative sum of errors over time to handle noise. The CUSUM formula that is used in this paper is defined as:

$$C_t = max[0, e_t - K + C_{t-1}] \quad (5)$$

where e_t is the error at state t, K is the *slack variable*, C_t is the cumulated error until state t, and C_{t-1} is the cumulated error until previous state. Slack variable can be defined as the variable that determines the sensitivity of the proposed approach to measure the error. When the error is greater than the slack variable, the CUSUM value will start growing until the error is smaller than the slack variable.

Each output variable of deep learning models has its own CUSUM value. The idea is when there is a fault in the plant, the CUSUM values can be used to discover the source of the fault and isolate it. It is desired that when there is no fault in the plant, CUSUM values of all outputs remain close to zero. Meanwhile, when there is a fault, one or some CUSUM values are deviating from zero. It is crucial that when a CUSUM value starts growing, the information can be discovered as early as possible to prevent failures.

C. Predictive Maintenance Pipeline

This paper proposes a multi-target regression approach for predictive maintenance using deep learning and CUSUM methods where only data from normal conditions is needed to train a model for detecting and isolating faults in refineries. Since only data from normal conditions is used in model training, the proposed approach provides more advantages

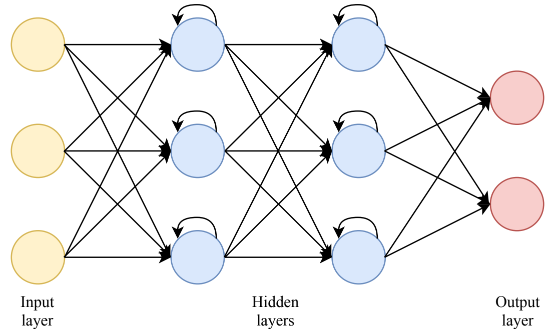


Fig. 4. An example of multi-target neural networks architecture

because data from abnormal conditions is difficult to obtain. The reason for using the multi-target regression approach is the number of control valves in refineries that need to be monitored, which can be hundreds or thousands. The multi-target regression approach can be implemented by using more than one neuron in the output layer of recurrent neural networks, as can be seen in Figure 4.

The predictive maintenance pipeline used in this paper is depicted in Figure 5. First, historical data that represents normal conditions of the plant is used to create a model using deep learning method. The training process is performed by adjusting the model parameters such that the prediction error of all outputs is minimized. Once the model is able to achieve a certain target performance, it is then used to analyze new unseen data. The trained model uses new data to make predictions, which will be evaluated using CUSUM method. If the cumulated error exceeds a certain threshold, a notification can be sent to trigger further investigation or maintenance activities. Unnecessary maintenance activities can be minimized while still preventing failures to happen. Therefore, by using predictive maintenance strategy, the plant availability time can be maximized.

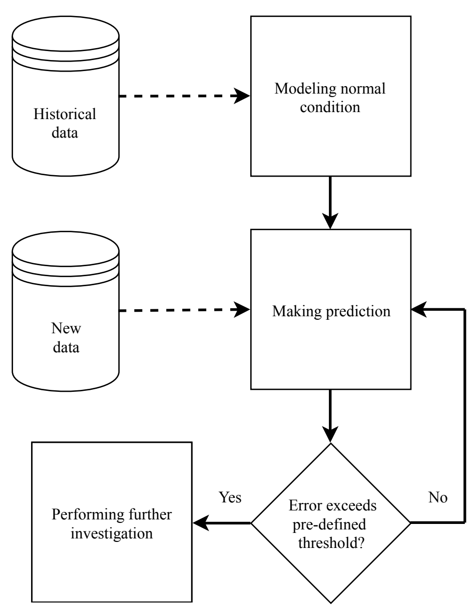


Fig. 5. Predictive maintenance pipeline

IV. Case Study at Shell Pernis

The proposed multi-target regression approach using deep learning and CUSUM method is demonstrated with real industrial data from a crude distiller at Shell Pernis. Crude distiller is a typical plant that is used to distill crude oil into several fractions which will be processed further in other plants. It is a complex processing unit that is controlled by a distributed control system using a variety of sensors and control valves. Control valves and the dataset used in this paper will be described in this section.

A. Control Valve

Control valves are a type of valves that are controlled by signals from a controller to control the passage of liquid or gas in the plants. This type of equipment is not only used in crude distillers but also other types of plants. There could be thousands of control valves used in a refinery. Therefore, control valves are considered as one type of critical equipment in refineries. In this case, the controller is a smart and complex system that controls many control valves based on a variety of sensors, as illustrated in Figure 6. During model training, the deep learning models are trying to learn the status of multiple control valves based on data that represent the normal conditions of the crude distiller. When new data that represent the current conditions does not match with the models, a fault might occur somewhere inside the crude distiller.

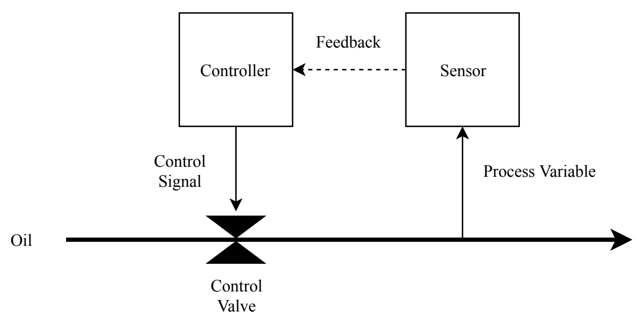


Fig. 6. Smart control system in the plant

The amount of liquid or gas that goes through a valve at a time, or flow rate, is determined by the opening position of the valve. In this case, the opening position of control valves is measured by percentage. A valve is fully close if the opening position is 0% and fully open if the opening position is 100%. Therefore, increasing the opening position of the valve will increase the flow rate as well. However, there are various types of relationships between the opening position of control valves and the flow rate, which are known as flow characteristics. Typically, there are three types of flow characteristics: quick-opening, linear, and equal-percentage [34], as can be seen in Figure 7. The linear flow characteristic means that the increment of flow rate is proportional to the increment of valve opening. In quick-opening characteristic, a small change of valve opening from the closed position creates a significant change of flow rate. The further the opening position from 0%, the smaller the change of flow rate. In equal-percentage characteristic, each increment of valve opening increases the flow rate by a certain percentage of the previous flow. The further the opening position from 0%, the larger the change of flow rate.

B. Dataset

The dataset used in this paper is collected from a crude distiller at Shell Pernis. It consists of 6-month data from normal conditions and 5-month data from abnormal conditions, where only data from normal conditions is used to train machine learning models. The abnormal condition in the crude distiller is caused by a fault in one of the control valves. Based on the inspection, it is found that the plug of the valve was slowly detaching from the stem, creating small disturbance in the crude distiller that is worsening over time. However, it is difficult to identify the disturbance, because the smart controller was trying to handle it. Therefore, at the early stage of the abnormal condition, everything was still working. However, after several months, the fault evolved into a failure and caused significant damage in the crude distiller.

In total, there are 175 process variables in the dataset, including 16 variables that represent the opening position of 16 control valves as the target variables. Due to some issues in data collection, the data needs to be cleaned. Some features in a few observations are missing, thus these observations need to be excluded. Furthermore, some observations that represent several manual interventions during normal conditions also need to be excluded, because they create unusual patterns in the data. No further data cleaning steps are performed, even though some periods seem suspicious.

V. Experiment and Analysis

All experiments are performed on an Amazon r3.4xlarge instance, a memory-optimized cloud server with Intel Xeon Ivy Bridge processors and 122 GB of RAM. The hyperparameter setting for RNNs models in this paper is similar to RNNs models in Suursalu [26], except the number of neurons in the output layer. RNNs models consist of 4 layers and 256 neurons in each hidden layer are evaluated in the experiment. Meanwhile, the suitable hyperparameter

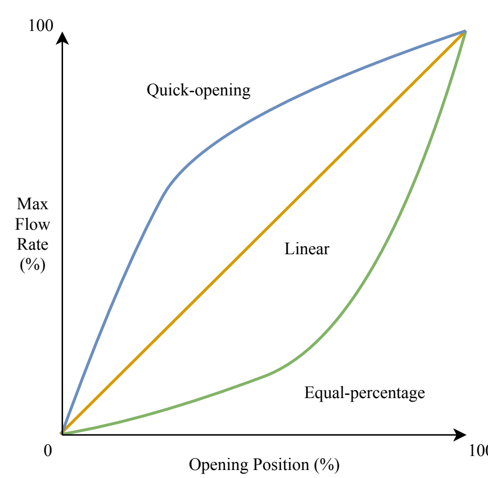
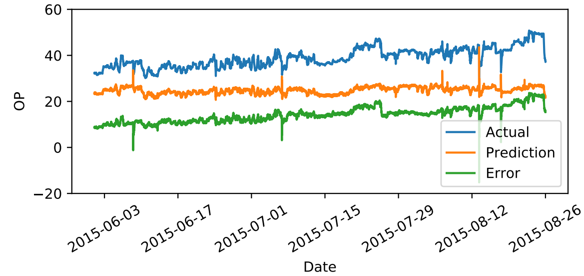


Fig. 7. Flow characteristics of control valves

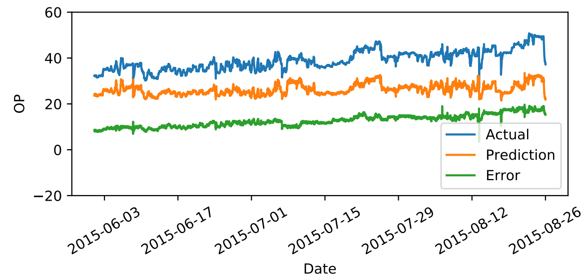
setting for autoencoders is obtained using hyperparameter optimization. The optimization process is performed until the accuracy of autoencoders models is similar to RNNs models in predicting normal control valves. An autoencoders model consists of 3 hidden layers with 85, 40, and 85 neurons in the hidden layers is evaluated as the optimal architecture in this case. Both RNNs and autoencoders models are trained using minibatch algorithm with a batch size of 256 and 20 epochs. Tensorflow [35], a free open source library for machine learning, is used to create deep learning models. For comparison, RNNs and elastic net [36] models with a single target are also utilized as the baseline. However, the elastic net models are created using MLlib [37], another open source library for distributed machine learning that provides more efficient data processing.

The prediction results for the faulty control valve during the abnormal condition from all models are shown in Figure 8. In general, all models are able to show higher error in predicting the faulty valve compared with healthy valves. However, more than one single-target model are needed to monitor more than one control valve. Meanwhile, only one multi-target model is required to monitor a large number of control valves. It is found from the experiment that the training time for single-target and multi-target RNNs models are similar, which is around 13 minutes. The additional computation from adding more neurons in the output layer of the multi-target RNNs model is very small or negligible. Thus, the required time to model 16 control valves using multi-target RNNs is 16 times faster compared with single-target RNNs. Furthermore, it only needs less than 2 minutes to train autoencoders models, which is 104 times faster compared with single-target RNNs or 6.5 times faster compared with multi-target RNNs. The speed-up can be obtained because no recurrent connection is used in autoencoders, which reduce the number of parameters that need to be calculated during model training. Moreover, since autoencoders try to copy its inputs to its outputs, the prediction error of other process variables apart from the opening position of control valves can also be analyzed for diagnosing faults. However, analyzing other process variables and diagnosing faults are beyond the scope of this paper.

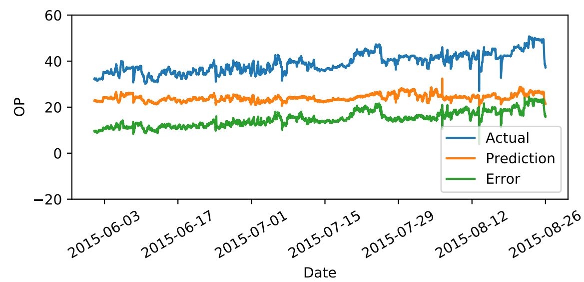
Furthermore, it is important that faults can be detected as early as possible to prevent failures. When the CUSUM errors exceed a certain threshold, a notification can be sent to maintenance engineers to warn them. Therefore, it is interesting to investigate whether the multi-target regression approach is able to improve fault detection time. The CUSUM errors of the faulty valve from all models measured in the last three months before the failure happened are depicted in Figure 9. It can be seen from the figure that the multi-target RNNs model produces higher CUSUM errors over time compared with other models. By comparing the multi-target and single-target RNNs models, the multi-target regression approach improves fault detection time by 1.2x compared with the single-target regression approach. Meanwhile, the CUSUM errors of the autoencoders model are lower than the elastic net model but still better than the single-target RNNs model. The autoencoders model only improves fault detection time by 1.1x compared with the single-target RNNs model.



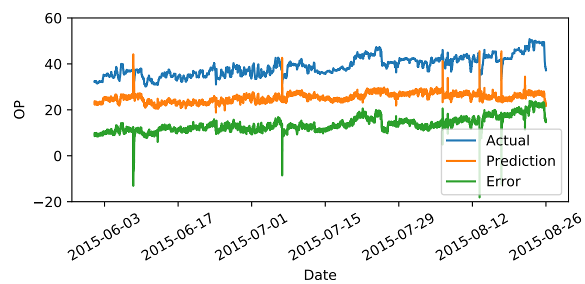
(a) Single-target elastic net



(b) Single-target RNNs



(c) Multi-target RNNs



(d) Multi-target Autoencoders

Fig. 8. Prediction results for the faulty control valve from different models. The plots display opening position in percentage.

VI. CONCLUSION

This paper presents a multi-target regression approach using deep learning and CUSUM method for predictive maintenance in oil refineries. The effectiveness of the proposed approach is verified using a real industrial dataset from a crude distiller at Shell Pernis. Such real datasets are usually noisy, incomplete, and sometimes incorrectly labeled, which makes them much harder to use to train effective machine learning models. This dataset contains a massive amount of data involving many

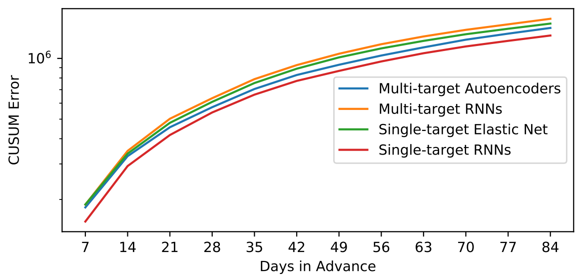


Fig. 9. Cumulative sum of error before the failure.

process variables that represent different conditions of the crude distiller. However, in this paper, only data from normal conditions is used for modeling the target processes. The experimental results show that the proposed approach is able to detect and isolate actual faults in the crude distiller. We used three modeling approaches: elastic net, (single and multi-target) RNNs, and autoencoders. Comparison of the models indicates that multi-target RNNs can improve fault detection time by a factor of 1.2x compared to single-target RNNs, while reducing the modeling time by 16x. Furthermore, autoencoders are able to reduce training time by 115x, while improving the detection time by 1.1x.

Acknowledgment

The authors would like to thank Peter Kwaspen, Sander Buhling, Maurice Hendrix, Arnold Hes, Ilsa-Maria Sillekens, Jan Limbeck, and Tijl van den Bos from Royal Dutch Shell for providing support and feedback on the research. This work has been supported by the Fitoptivis European ECSEL project no. ECSEL2017-1-737451.

References

[1] J. G. Speight, *The Chemistry and Technology of Petroleum*. CRC Press, 1999.

[2] K. Wang and Y. Wang, "How AI affects the future predictive maintenance: A primer of deep learning," in *Advanced Manufacturing and Automation VII*. Springer Singapore, 2018, pp. 1–9.

[3] R. Leger, W. Garland, and W. Poehlman, "Fault detection and diagnosis using statistical control charts and artificial neural networks," *Artificial Intelligence in Engineering*, vol. 12, no. 1, pp. 35–47, 1998.

[4] E. Sobhani-Tehrani and K. Khorasani, *Fault Diagnosis of Nonlinear Systems Using a Hybrid Approach*. Springer US, 2009.

[5] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, Jul. 1959.

[6] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[9] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 International Joint Conference on Neural Networks*. IEEE, Jul. 2011, pp. 2809–2813.

[10] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016.

[11] A. Kayihan and F. J. Doyle, "Friction compensation for a process control valve," *Control Engineering Practice*, vol. 8, no. 7, pp. 799–812, 2000.

[12] M. Choudhury, N. Thornhill, and S. Shah, "A data-driven model for valve stiction," *IFAC Proceedings Volumes*, vol. 37, no. 1, pp. 245–250, 2004.

[13] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, Sep. 2015.

[14] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, Jun. 1954.

[15] W. Yan, "Application of random forest to aircraft engine fault diagnosis," in *The Proceedings of the Multiconference on "Computational Engineering in Systems Applications"*, vol. 1. IEEE, Oct. 2006, pp. 468–475.

[16] Q. P. He and J. Wang, "Principal component based k-nearest-neighbor rule for semiconductor process fault detection," in *2008 American Control Conference*. IEEE, Jun. 2008, pp. 1606–1611.

[17] S. Li and J. Wen, "A model-based fault detection and diagnostic methodology based on PCA method and wavelet transform," *Energy and Buildings*, vol. 68, pp. 63–71, 2014.

[18] P. Tamilselvan and P. Wang, "Failure diagnosis using deep belief learning based health state classification," *Reliability Engineering & System Safety*, vol. 115, pp. 124–135, 2013.

[19] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 303–315, 2016.

[20] Y. Qi *et al.*, "An automatic feature learning and fault diagnosis method based on stacked sparse autoencoder," in *Advanced Manufacturing and Automation VII*. Springer Singapore, 2018, pp. 367–375.

[21] H. Shao, H. Jiang, H. Zhao, and F. Wang, "A novel deep autoencoder feature learning method for rotating machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 95, pp. 187–204, 2017.

[22] S. Tang *et al.*, "Rolling bearing fault diagnosis using deep learning network," in *Advanced Manufacturing and Automation VII*. Springer Singapore, 2018, pp. 357–365.

[23] V. T. Tran, F. AlThobiani, and A. Ball, "An approach to fault diagnosis of reciprocating compressor valves using Teager-Kaiser energy operator and deep belief networks," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4113–4122, 2014.

[24] B.-S. Yang, W.-W. Hwang, M.-H. Ko, and S.-J. Lee, "Cavitation detection of butterfly valve using support vector machines," *Journal of Sound and Vibration*, vol. 287, no. 1, pp. 25–43, 2005.

[25] M. Karpenko, N. Sepehri, and D. Scuse, "Diagnosis of process valve actuator faults using a multilayer neural network," *Control Engineering Practice*, vol. 11, no. 11, pp. 1289–1299, 2003.

[26] S. Suursalu, "Predictive maintenance using machine learning methods in petrochemical refineries," Master's thesis, Delft University of Technology, Jul. 2017.

[27] M. Timusk, M. Lipsett, and C. K. Mechefske, "Fault detection using transient machine signals," *Mechanical Systems and Signal Processing*, vol. 22, no. 7, pp. 1724–1749, 2008.

[28] H. A. Talebi, K. Khorasani, and S. Tafazoli, "A recurrent neural-network-based sensor and actuator fault detection and isolation for nonlinear systems with application to the satellite's attitude control subsystem," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 45–60, Jan. 2009.

[29] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013.

[30] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, Jan. 2017, pp. 140–145.

[31] S. Haykin, *Neural networks and learning machines*. Pearson, 2009.

[32] R. Silipo, *Neural Networks*. Springer Berlin Heidelberg, 2003, pp. 269–320.

[33] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and helmholtz free energy," in *Advances in Neural Information Processing Systems 6*. Morgan-Kaufmann, 1994, pp. 3–10.

[34] Emerson, "Control valve handbook," Emerson, Tech. Rep., 2017.

[35] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*. USENIX Association, 2016, pp. 265–283.

[36] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[37] X. Meng *et al.*, "MLlib: Machine learning in apache spark," *Journal of Machine Learning Research*, vol. 17, no. 34, pp. 1–7, 2016.