12th IFAC International Symposium on
Advanced Control of Chemical Processes
July 14-17, 2024. Toronto, Canada

TabNet-based Self-supervised Fault Diagnosis in Multivariate Time-series Process Data without Labels ⋆

Hae Rang Roh, Jong Min Lee *

* *School of Chemical and Biological Engineering, Seoul National University, Seoul 08826, Republic of Korea (e-mail: hryellow9867@snu.ac.kr, jongmin@snu.ac.kr).*

Abstract: Fault diagnosis is an essential field for the safe operation of chemical processes. In this paper, a self-supervised fault diagnosis method employing a tree-based deep learning model is proposed. The temporal information of multivariate time-series data is compressed with a Long Short-Term Memory structure, and the proposed method is demonstrated by performing the classification of fault types in the Tennessee Eastman process. It showed substantial performance enhancement compared to supervised learning, leveraging the feature representation obtained from unlabeled data. Notably, the tree-based characteristic of the proposed method provides interpretability of model results, illuminating the salient features of each fault type.

Keywords: process monitoring, fault diagnosis, self-supervised learning, tree-based deep learning, TabNet, interpretability

1. INTRODUCTION

There is no denying that safety should be a top priority when operating an industrial process. Especially in chemical processes, there is a risk of irreversible damage in the event of anomalies. Therefore, comprehensive research into the categorization and causation of potential faults becomes imperative for ensuring both the safety and efficiency of process operations. This need becomes the driving force behind the active research in the realm of process monitoring and fault diagnostics over the past several decades. In the early days of research, methodologies grounded in statistical models, such as Principal Component Analysis (PCA) or PCA-based methods like dynamic PCA and kernel PCA, were mainly developed. Subsequently, with advancements in computational power, data-driven machine learning techniques became widely used in the field of fault diagnosis. Notably, Support Vector Machine (SVM) has evolved into the most widely employed traditional machine learning technique, principally designed for binary classification. Its application extends to solving multiple classification challenges through strategies like one-against-one (OAO) or one-against-all (OAA) as used in Yuan and Chu (2006) and Jin et al. (2014). In addition, the exploration of various SVM variants and optimization techniques, including least-square SVM, wavelet SVM, and ensemble SVM, ensued to mitigate the limitations associated with OAO and OAA strategies while enhancing overall performance as reviewed in Hsu and Lin (2002). Furthermore, adaptations and enhancements of the k-nearest neighbor technique have been explored and applied to fault diagnosis problems. However, these traditional machine learning techniques confront inherent limitations, notably the significant impact of a few critical hyperparameters on performance. Moreover, they are constrained by their incapacity to effectively utilize the vast amount of data available in contemporary times.

In recent years, the rapid advancement of deep learning within the domains of Computer Vision (CV) and Natural Language Processing (NLP) has prompted its active application into the realm of fault diagnosis. Jiang et al. (2016) introduced an active learning strategy selecting pivotal features with employing a stacked denoising auto-encoder. Subsequently, Jiang et al. (2017) conducted semi-supervised learning through the utilization of a stacked sparse auto-encoder. Further contributions by Zhang and Zhao (2017), Yu and Yan (2018), and Wang et al. (2020) applied deep belief networks to extract salient features characterized by abnormal fluctuations, thereby facilitating fault detection and diagnosis. Wu and Zhao (2018) introduced a convolutional Neural Network (CNN)-based deep CNN network, while Han et al. (2020) proposed a recurrent Neural Network (RNN)-based architecture to capture the intricate temporal correlations inherent in time series process data. Xia et al. (2022) diagnosed faults of centrifugal chillers by extreme learning machine combined with kernel entropy component analysis. More recently, attention-based models, which account for dependencies within a sequence, have garnered substantial interest, particularly for overcoming limitations associated with RNN-based methodologies. These models have found applicability in diverse domains, including NLP and CV field. Li et al. (2019) incorporated the attention mechanism within a deep CNN-based framework, thereby endowing the model with the capability to discern informative segments within the data during fault diagnosis. Recently, the field of self-supervised learning, which leverages unlabeled data, has gained significant attention. Researchers increas-

⋆ This work was financially supported by Samsung Electronics Co.,Ltd..

ingly use vast volumes of unlabeled data to pretrain models such as BERT and imageNet. These pretrained models are then applied to various tasks. Deep learning benefits from this approach as it allows the use of pretrained architectures in downstream applications, including self-supervised learning or domain adaptation.

Nervertheless, these deep learning techniques have some limitations. With the exception of a few cases utilizing image data, the predominant format of process data remains tabular. In this context, tree-based approaches such as Extreme Gradient Boosting (XGBoost) or Light Gradient-Boosting Machine (LightGBM) continue to exhibit notable efficacy, characterized by proficient performance with small model sizes and rapid learning rates compared to deep learning. By contrast, the current state-of-the-art models dominating the field of deep learning usually possess several hundred million parameters. In addition, in the case of fault diagnosis in chemical processes, it is not only necessary to determine which fault has occurred, but also to analyze which features are affected by the fault and influential in the model's decision-making. Deep learning, in its intrinsic nature, lacks inherent interpretability and necessitates the incorporation of supplementary explainable AI methodologies.

Therefore, there have been a number of studies to combine the advantages of both tree-based learning and deep learning. Humbird et al. (2018) attempted to represent a decision tree as a Deep neural net (DNN) block, and Tanno et al. (2019) proposed a DNN architecture by adaptively growing from primitive blocks. Most recently, Arik and Pfister (2021) proposed a structure called TabNet and it showed proficient performance by combining the advantages of tree-based methods and deep learning. However, a critical limitation arises in its applicability to multivariate time-series data, a prevalent format in the domain of process data.

In this study, we perform self-supervised fault diagnosis of chemical processes utilizing a tree-based deep learning architecture, TabNet. Before the fault diagnosis process, a Long Short-Term Memory (LSTM) structure is employed to compress the temporal information of process data and identify the effects of each variable on diagnosis results, instead of simply flattening the data. The realm of chemical processes is an ideal context for the application of self-supervised learning that utilizes unlabeled data for feature representation. A vast amount of process data can be collected during continuous plant operations, but it is much more labor-intensive and time-consuming to identify fault labels for specific data. Therefore, leveraging the benefits of obtaining feature representation from an extensive amount of unlabeled data is essential for fault diagnosis. The main contributions are summarized as follows:

- Self-supervised fault diagnosis of chemical process with multivariate time-series data is performed by tree-based deep learning architecture, combining with LSTM structure to compress the temporal information of data.
- The proposed method is demonstrated using the Tennessee Eastman process, a simulated industrial chemical process, and shows enhanced diagnosis performance compared to supervised learning, even with a compact model size, by utilizing unlabeled data widely available in the actual operation of the process.
- The tree-based characteristic of the proposed method provides interpretability of specific data samples by elucidating which features had a dominant impact on the fault diagnosis results of the model.

2. METHOD

2.1 Time compressing with LSTM

Typical tree-based models, including TabNet, are designed to handle tabular data that have only features without a time axis. Therefore, we employed a LSTM architecture to compress the temporal information of each variable within the process data. In order to facilitate unsupervised learning with unlabeled data, an autoencoder (AE) structure was implemented to perform the reconstruction task of input data. To assess the extent of reconstruction, Mean Squared Error (MSE) loss was employed for training.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

To compress the temporal information of each feature into a scalar value, the latent space of the autoencoder was set to one dimension, and then only the encoder part was utilized in the subsequent fault diagnosis step.

2.2 TabNet architecture

TabNet is a tree-based deep learning architecture specialized for tabular data Arik and Pfister (2021). It performs sequential multi-step processing through N decision steps wherein i^{th} step receives processed data from the $(i-1)^{th}$ step. Each step discerns relevant features for utilization and then processes the data through DNN modules. Subsequently, all processed data from each step is integrated to yield the final output. A decision step has the structure of an attentive transformer and a feature transformer.

A pivotal aspect of TabNet lies in its ability to learn masks facilitating sparse feature selection. By retaining salient features while excluding unimportant ones, the model's efficiency and compactness are enhanced. These masks in eq. 2 are derived from an attentive transformer in each step, which consists of a fully-connected (FC) layer, batch normalization (BN) and sparsemax function.

$$M[i] = sparsemax(P[i-1] \cdot h_i(a[i-1])) \tag{2}$$

Sparsemax normalization encourages sparsity by mapping the Euclidean projection on the probabilistic simplex, which is superior in performance and provides salient features for explainability. h_i of $M[i]$ is a trainable function, using a FC layer and BN, and $a[i-1]$ is the processed features from the preceding step. $P[i]$ is the prior scale term, denoting how much a particular feature has been used previously.

$$P[i] = \Pi_{j=1}^{i}(\gamma - M[j]) \tag{3}$$

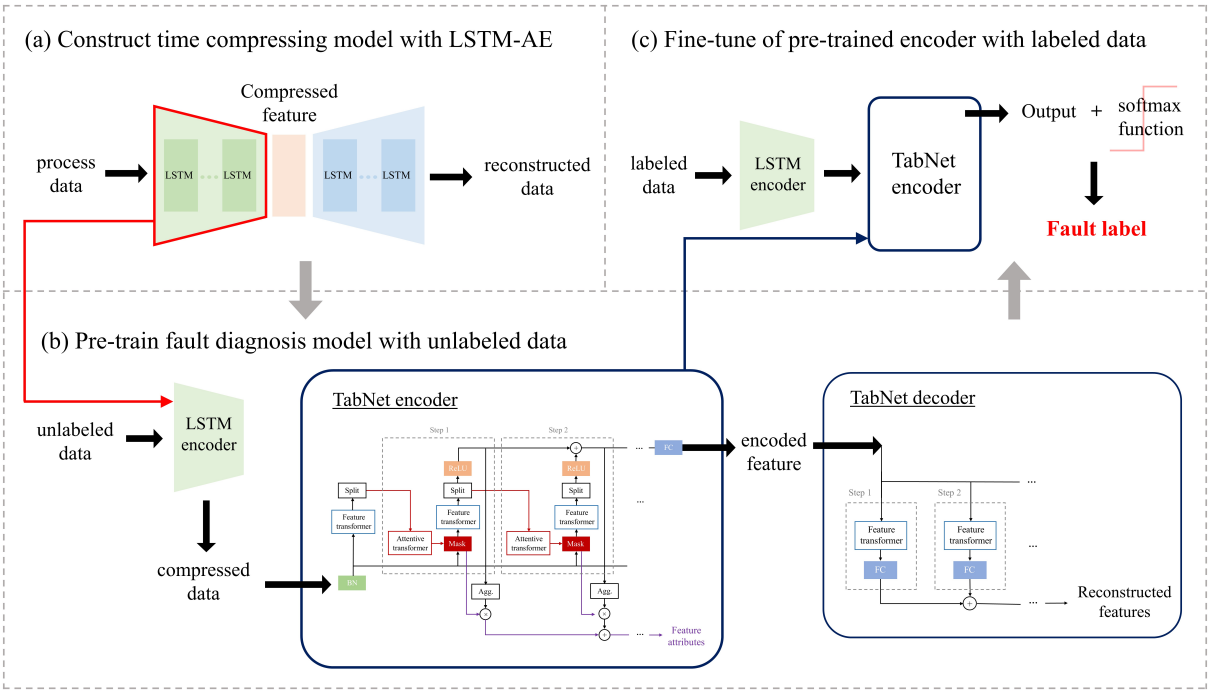

Fig. 1. Overall procedure of self-supervised learning for fault diagnosis of chemical process

$P[i]$ is determined by excluding previously utilized masks based on a relaxation parameter, γ. Consequently, when γ equals 1, each feature is employed only once across all decision steps, and with an increasing γ, a feature can be utilized in multiple steps. Furthermore, to control the sparsity of the selected features, TabNet uses sparsity regularization in the form of entropy and a sparsity regularization term multiplied by a sparsity coefficient, λ_{sparse}, added to the overall loss. The hyperparameter λ_{sparse} offers control over the sparsity of feature selection.

The data, filtered by the mask obtained from the attentive transformer, undergoes processing in the feature transformer. The feature transformer comprises several small sets of layers consisting of a FC layer, BN, and a gated linear unit (GLU). For parameter-efficient and robust learning with high capacity, some sets of layers in a feature transformer are shared across all decision steps, as well as sets for decision step-dependent. The data processed by the feature transformer is split for the decision step output, $d[i]$ and information for the subsequent step, $a[i]$.

$$[d[i], a[i]] = f_i(M[i] \cdot f) \tag{4}$$

The outputs from all steps are aggregated and passed through a FC layer to generate the final output.

For unsupervised learning, a decoder is necessary to perform reconstruction task of the input data. Similar to the encoder, the decoder comprises multiple decision steps, each featuring a feature transformer and an FC layer. The outputs from each step are summed to obtain reconstructed features. The MSE loss between the reconstructed features and the input of the encoder is backpropagated to drive the learning process.

2.3 Overall procedure

The overall self-supervised fault diagnosis procedure is shown in Fig. 1. Firstly, an LSTM autoencoder for compressing the temporal information was trained using the total process dataset without labels. Through the trained encoder, multivariate time-series data were converted into data containing only features. Converted training data without labels were then utilized for training the TabNet encoder and decoder to obtain the feature representation of process data. The pretrained TabNet encoder possessed optimized model parameters for describing the feature representation, facilitating improved fault diagnosis performance compared to using only a small portion of data with labels. Eventually, for fine-tuning, only the pretrained encoder was imported to conduct the classification task of faults, leveraging trained parameter values as initial parameters. Additionally, the softmax function was applied to the final output of the FC layer, serving as a classifier, and the cross-entropy loss was used between the predicted labels and the actual labels.

3. RESULTS

3.1 Process description

The chemical process selected for the application of the proposed model is the Tennessee Eastman process (TEP). The TEP is a simulated process based on a real chemical process designed by Downs and Vogel (1993) for evaluating process control techniques. Over time, TEP has become a benchmark chemical process widely utilized in the domains of process control and fault diagnosis. The TEP dataset is acquired from Harvard Dataverse [1] .

[1] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/6C3JR1

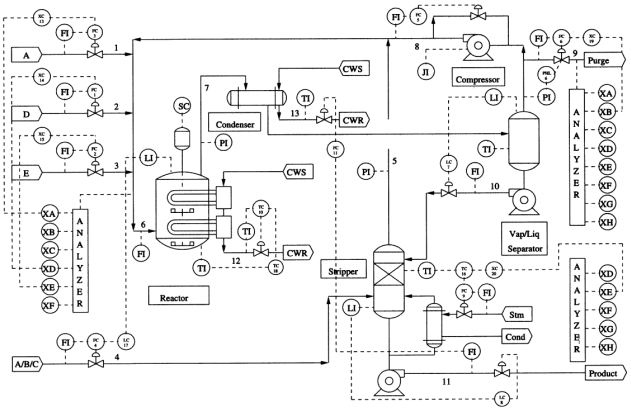

Fig. 2. A process flowsheet for the TEP

A process flowsheet of the TEP is depicted in Fig. 2. The TEP data has 11 manipulated variables and 41 measured variables, but the variables 'compressor recycle valve' and 'stripper steam valve' are not changed in any fault scenario. Consequently, a total of 50 variables were employed, excluding these two manipulated variables. In addition, the TEP is designed to introduce 15 faults with discernible causes, but three specific faults (fault 3, 9, and 15) pose considerable diagnostic challenges due to slight variation in the mean and variance of the variables in comparison to the normal scenario. Therefore, we used data from a total of 13 fault scenarios and normal scenario, and the fault list is shown in Table. 1.

Table 1. Faults utilized in the TEP

Fault ID	Description	Type
IDV1	A/C Feed ratio	Step
IDV2	B composition	Step
IDV4	Reactor cooling water inlet temperature	Step
IDV5	Condenser cooling water inlet temperature	Step
IDV6	A feed loss	Step
IDV7	C Header pressure loss	Step
IDV8	A, B, C Feed composition	Random variation
IDV10	C Feed temperature	Random variation
IDV11	Reactor cooling water inlet temperature	Random variation
IDV12	Condenser cooling water inlet temperature	Random variation
IDV13	Reaction kinetics	Slow drift
IDV14	Reactor cooling water valve	Sticking

For training, 500 simulation data points were used for all faults and normal scenarios, and 100 simulation data points were used for testing. Each training simulation encapsulated 25 hours of operation, with variables measured every 3 minutes, resulting in 500 time steps. The test simulation extended over 48 hours, with a total of 960 time steps. For training, both the training and test data were segmented into samples with 20 time steps and a window size of 16. All data were utilized without labels indicating the type of faults, and a certain proportion of the training data was employed with labels in the latter supervised fine-tuning step.

3.2 Experimental setup

The LSTM autoencoder for compressing the temporal information had four LSTM layers in each encoder and decoder. The batch size was set to 64, and the learning rate was set to 0.001. For unsupervised pretraining, the TabNet encoder and decoder included three decision steps ($N_{steps} = 3$). The feature transformer comprised two shared blocks across decision steps ($N_{shared} = 2$) and two step-dependent blocks ($N_{independent} = 2$). The dimension of the decision step output and information for the subsequent step was set to 8 identically ($N_d = 8$, $N_a = 8$). During the reconstruction task, a certain percentage of masking was arbitrarily applied to the encoder's input, and the masking ratio was set to 0.8 through case studies. A relaxation parameter, γ, was set to 1.3, and a sparsity parameter, λ_{sparse}, was fixed at 0.001. The batch size was set to 1024 for fast learning, and the learning rate was set to 0.02. For the fine-tuning phase, the learning rate was initiated at 0.02 and scheduled to decrease at a rate of 0.9 every 10 epochs with the Adam optimizer.

3.3 Application

To assess the efficacy of different time compression methods, self-supervised fault diagnosis was conducted on the temporally compressed data. The TabNet structure utilized was the final model outlined in Section 3.2, with 10% of labeled data. A comparative analysis was performed between a two-layer LSTM autoencoder and a fully-connected autoencoder. The fully-connected autoencoder, featuring two layers with 64 and 16 nodes, exhibited the lowest performance in fault diagnosis, as shown in Table 2. In cases involving an LSTM autoencoder, slightly better performance was observed with four layers, prompting the adoption of a four-layer LSTM autoencoder.

Table 2. Comparison of various types of time compression method

	Linear	2 layer LSTM	4 layer LSTM
accuracy	43.13	70.48	76.17

Subsequently, we compared the fault diagnosis performance across varying mask percentages with 10% labeled data for the reconstruction task of the TabNet autoencoder, as shown in Table 3. Generally, excessively low values can cause ineffective learning, while excessively high values hinder the reconstruction task, thus impairing performance. In this case study, the pretraining effect peaked as the mask ratio increased, indicating that a stringent training regimen leads to superior feature representation.

Table 3. Impact of the mask ratio on the pretrain model

mask ratio	0.6	0.7	0.8
accuracy	73.89	74.87	76.17

After pretraining the TabNet encoder, the classification task of fault diagnosis was conducted with labeled data. To check the performance of self-supervised learning, we compared it with the case of supervised learning utilizing the same amount of labeled data without pretraining, while increasing the amount of labeled data. According to Fig. 3, self-supervised learning outperformed supervised learning in all cases. It can be inferred that the feature representation extracted from unlabeled data is beneficial for the classification across different fault types. The performance of both self-supervised learning and supervised

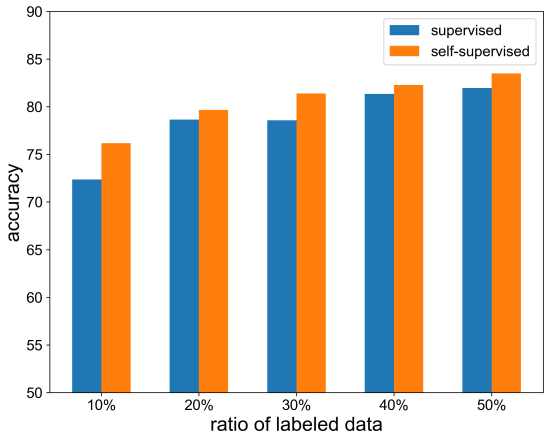

Fig. 3. Performance according to the quantity of labeled data in supervised and self-supervised learning

learning generally increased in proportion to the amount of labeled data. The more labeled data we used, the smaller the difference between the performance of self-supervised learning and supervised learning. Therefore, the effectiveness of pretraining using unlabeled data is maximized in situations where labeled data is extremely scarce. Additionally, the tree-based model has an advantage in achieving good performance even with a relatively small model size. The proposed self-supervised model utilized fewer than 50k parameters instead of possessing heavy models with several million parameters.

Table 4. Fault diagnosis results of supervised and self-supervised learning

Fault ID	supervised	self-supervised
IDV1	85.00	**86.71**
IDV2	**83.24**	79.27
IDV4	23.63	**26.61**
IDV5	79.10	**82.14**
IDV6	86.29	**99.27**
IDV7	**94.22**	93.82
IDV8	95.29	**96.49**
IDV10	85.04	**86.39**
IDV11	58.16	**69.12**
IDV12	95.76	**96.02**
IDV13	**93.63**	93.12
IDV14	92.84	**96.80**

Table 4 shows the test accuracy for each fault in self-supervised learning and supervised learning, utilizing 30% labeled data. In most faults, self-supervised learning indicated improved performance compared to supervised learning. Fault 4 posed challenges for both learning methods, given its fault magnitude of 9% relative to the normal scenario, representing one of the faults with the lowest magnitude difference. Consequently, it is difficult to distinguish fault 4 from normal data when using a limited amount of labeled data. Nevertheless, the use of unlabeled data contributed to a slight increase in accuracy.

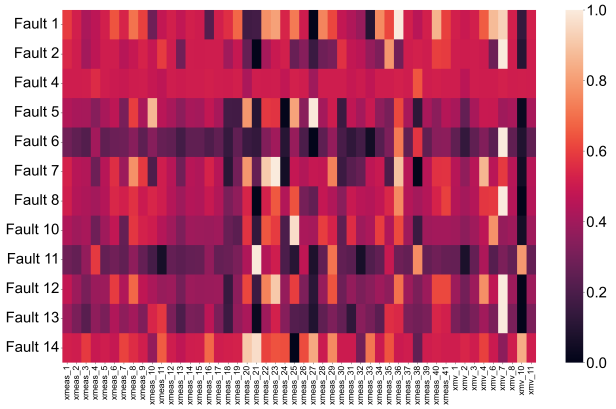

Fig. 4. Interpretability scores for each variable obtained from the aggregation of masks of test samples

As mentioned in Section 2.2, the tree-based characteristic of TabNet employs learnable masks, only utilizing salient features while excluding unimportant ones. After training, aggregation of all masks at each step provides interpretability of the model outputs for individual test samples. In this study, raw aggregation of all masks showed biased outcomes, wherein some common remarkable features existed across both normal scenarios and all types of fault scenarios, regardless of the fault types. Therefore, we subtracted the aggregation of masks of normal samples from the aggregation of masks of each fault sample to identify features specialized for a specific fault type, regardless of the characteristics of the process itself.

Fig. 4 shows the result, illustrating the influence of each variable on the model results. For each fault type, the score is the mean of all samples of the same faults and then scaled between the maximum and minimum values to compare with the score of other faults. In Fig. 4, white parts can be thought of as salient features explaining the model results, and black parts as meaningless features, on the contrary. For example, the variable 'xmv 10' of 'reactor cooling water flow' explains only Fault 11 and Fault 14, which represent the faults related to 'reactor cooling water inlet temperature' and 'reactor cooling water valve,' respectively. This result demonstrates that these faults are problems arising from reactor cooling water. Actually, Fault 4 is another fault related to reactor cooling water, but the model cannot illuminate any specific features along with low diagnosis accuracy, as shown in Fig. 4. On the other hand, the variable 'xmv 7' of 'separator pot liquid flow (stream 10)' is highlighted in faults related to reactor feed or reaction kinetics, such as Faults 2, 6, 8, and 13. It indicates that the composition of the separator output is the most prominently differentiating factor for these fault types compared to the normal scenario. In chemical processes, variables in the rear end of the process are highly dependent on other variables in the front end, and especially in the case of the TEP process which has a recycling unit, it was difficult to identify the detailed root cause of faults or derive one or two sparse features that thoroughly represent each fault. Nonetheless, the tree-based deep learning architecture implies the possibility of a deep learning model with inherent interpretability of model results.

4. CONCLUSION

In this paper, a self-supervised fault diagnosis method employing a tree-based deep learning architecture with a time-compressing process is proposed. To identify the influence of each feature on the model results, the temporal information of multivariate time-series data is compressed with an LSTM encoder. Subsequently, self-supervised learning using the TabNet architecture is conducted with only a limited amount of labeled data. The proposed method is demonstrated on the simulated industrial TEP dataset and exhibits outstanding performance compared to supervised learning. A substantial enhancement in performance is achieved through pretraining with unlabeled data, and the effect is particularly pronounced when labeled data is scarce. Given the constant operation of chemical processes, acquiring unlabeled raw data is straightforward, but the manual labeling of data to identify faults is a labor-intensive task. Thus, the capacity to leverage abundant unlabeled data for feature representation, not accessible in supervised learning, is of particular significance. Moreover, the tree-based approach offers interpretability of model results by illuminating the salient features for each fault type. By examining the masks of TabNet for test samples, we can reason about the approximate issues caused by the process faults. Possibilities for further research may exist, focusing on identifying the definite root cause of faults.

ACKNOWLEDGEMENTS

This work was financially supported by Samsung Electronics Co.,Ltd..

REFERENCES

Arik, S.Ö. and Pfister, T. (2021). Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 6679–6687.

Downs, J.J. and Vogel, E.F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3), 245–255.

Han, Y., Ding, N., Geng, Z., Wang, Z., and Chu, C. (2020). An optimized long short-term memory network based fault diagnosis model for chemical processes. *Journal of Process Control*, 92, 161–168.

Hsu, C.W. and Lin, C.J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2), 415–425.

Humbird, K.D., Peterson, J.L., and McClarren, R.G. (2018). Deep neural network initialization with decision trees. *IEEE Transactions on Neural Networks and Learning Systems*, 30(5), 1286–1295.

Jiang, L., Ge, Z., and Song, Z. (2017). Semi-supervised fault classification based on dynamic sparse stacked auto-encoders model. *Chemometrics and Intelligent Laboratory Systems*, 168, 72–83.

Jiang, P., Hu, Z., Liu, J., Yu, S., and Wu, F. (2016). Fault diagnosis based on chemical sensor data with an active deep neural network. *Sensors*, 16(10), 1695.

Jin, X., Feng, J., Du, S., Li, G., and Zhao, Y. (2014). Rotor fault classification technique and precision analysis with kernel principal component analysis and multi-support vector machines. *Journal of Vibroengineering*, 16(5), 2582–2592.

Li, X., Zhang, W., and Ding, Q. (2019). Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism. *Signal Processing*, 161, 136–154.

Tanno, R., Arulkumaran, K., Alexander, D., Criminisi, A., and Nori, A. (2019). Adaptive neural trees. In *International Conference on Machine Learning*, 6166–6175. PMLR.

Wang, Y., Pan, Z., Yuan, X., Yang, C., and Gui, W. (2020). A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA Transactions*, 96, 457–467.

Wu, H. and Zhao, J. (2018). Deep convolutional neural network model based chemical process fault diagnosis. *Computers & Chemical Engineering*, 115, 185–197.

Xia, Y., Ding, Q., Jiang, A., Jing, N., Zhou, W., and Wang, J. (2022). Incipient fault diagnosis for centrifugal chillers using kernel entropy component analysis and voting based extreme learning machine. *Korean Journal of Chemical Engineering*, 39(3), 504–514.

Yu, J. and Yan, X. (2018). Layer-by-layer enhancement strategy of favorable features of the deep belief network for industrial process monitoring. *Industrial & Engineering Chemistry Research*, 57(45), 15479–15490.

Yuan, S.F. and Chu, F.L. (2006). Support vector machines-based fault diagnosis for turbo-pump rotor. *Mechanical Systems and Signal Processing*, 20(4), 939–952.

Zhang, Z. and Zhao, J. (2017). A deep belief network based fault diagnosis model for complex chemical processes. *Computers & Chemical Engineering*, 107, 395–407.