Partially Supervised Classification for Industrial System using Deep Neural Network

Hao Huang[1], Tapan Shah[2]

[1,2] *GE Global Research, San Ramon, CA, 94582, USA*
[1]*hao.huang1@ge.com*
[2]*tapan.shah@ge.com*

ABSTRACT

A general classification setting requires prior knowledge (i.e. labeled samples) to cover all classes. However, in many industrial problems, prior knowledge usually does not describe all the classes, and the generation of a complete training set that covers all classes often is a time-consuming, expensive and difficult (if not impossible) task. Assuming the dataset contains a small amount of labeled time series that only cover some (treated as known classes) but not all the classes, and a large amount of unlabeled time series from either known or unknown classes, we aim to assign class labels to any of those unlabeled time series. Specifically, our model not only detects the novel time series, but also distinguishes them into different unknown classes. We develop an advanced deep neural networks, Partially Supervised Time Series Classification (*PSTC*), to handle such situation for industrial system. The model consists of one general encoder, and $k+u$ parallel decoders that correspond to k known and u unknown classes, where the labeled time series are used to update the known classes' networks and the unlabeled time series are used to update the networks that give the smallest residual. At the end, each unlabeled time series is assigned to the class that gives the minimum residual. We test our algorithm on two industrial time series classification problems and experiments show that our approach outperforms popular deep learning baselines.

1. INTRODUCTION

The science of AI, especially deep-learning based classification models, have already achieved many significant success and are used in many fields in different industries (Zhang et al., 2019; Rezaeianjouybari & Shang, 2020; Fink et al., 2020; Yucesan, Dourado, & Viana, 2021). However, most of their success, if not all, are based on the assumption that the predefined classification system is closed and complete. In other words, they assume that training set are from a fixed set of classes, and there are no unknown or novel classes in the unseen data. Nevertheless, this assumption may be too strict for the real world. For some problems, we often cannot have knowledge of the entire set of possible classes. For instance, in a fault classification problem on wind turbines, due to the dynamic conditions there are always unseen faults that not yet revealed in the previous knowledge. Therefore, a more realistic scenario is usually open and non-stationary such as fault detection and alarm management approaches in complex industrial processes etc., where unseen classes can emerge unexpectedly that drastically weakens the robustness of the existing classification approaches.

In a classification model that recognizes time series samples collected from industrial system, the ability to handle samples from unseen classes is crucial (Dhamija, Günther, & Boult, 2018). There are several works that try to avoid false positives by rejecting the samples from unseen classes, that sometimes is also called out-of-distribution (OOD) detection (Kaur et al., 2022). Specifically, their target is to assign a negative label to any sample from the new classes (Dhamija et al., 2018; Perera & Patel, 2019), which can be understood as classifiers that can detect outliers outside of training set distribution. But a more sophisticated solution is to **further classify the samples from unseen classes**. To tackle this issue, we propose a multitask deep learning approach that simultaneously conducts classification on samples from known classes and distinguishes unknown samples into different classes.

In particular, we propose an end-to-end deep-learning based approach in which we investigate how the labeled and unlabeled time series can be used to improve the performance on classifying samples from known classes as well as grouping samples from unknown classes. We name our model as **Partially Supervised Time Series Classification** (*PSTC*). Our solution differs from the standard deep classification networks on the following accounts:

1. To the best of our knowledge, this is the first end-to-end deep learning solution for industrial time series data that targets both known and unknown pattern classification simultaneously. Especially, our model can distinguish

different unknown classes, instead of treating all of them as general outliers.

2. We formulate the problem as a multi-paths autoregression problem, and design a learning framework that consists of one general encoder and multiple decoders by minimizing the regression residual across different paths.
3. We introduce a new loss function which takes both known and unknown patterns into account, by encouraging the samples with previously-seen patterns labeled to the corresponding known classes and the samples with unseen patterns labeled to the new classes.
4. We show empirically, that the proposed approach outperforms popular deep learning based time series classification baselines on both synthetic and real world datasets.

2. RELATED WORK

With the rapid development and expansion of sensors, time series data are widely available in many industrial systems, such as wind turbines, aircrafts and power plants. Due to the complex design of those systems, the collected time series data are usually non-stationary, nonlinear, and with noisy nature. This presents an opportunity to bring deep neural networks to bear on the industrial time series learning problems, because of its ability to extract features from the raw signals without the need to perform feature engineering or specify the distance measurement as in typical learning approaches.

There are three major types of deep neural networks that have been widely used in time series classification: recurrent neural networks (*RNN*), temporal convolution networks (*TCN*), and attention-based networks (Transformer).

Recurrent neural networks (*RNN*) are neural networks with recurrent connections, which are capable of modelling sequential data for time series recognition and prediction (Bengio, Simard, & Frasconi, 1994; Salehinejad, Sankar, Barfett, Colak, & Valaee, 2017). *RNNs* are made of high dimensional hidden states with non-linear dynamics. The structure of hidden states work as the memory of the network, and state of the hidden layer at a time is conditioned on its previous state (Salehinejad et al., 2017). This structure enables the *RNNs* to store, remember, and process past complex signals for long time periods. Therefore, *RNNs* can map an input sequence at a time to the output for prediction and recognition. Long Short Term Memory (*LSTM*) (Hochreiter & Schmidhuber, 1997) is one of the most popularly used *RNN* models.

Temporal Convolutional Networks (*TCN*) (Lea, Vidal, Reiter, & Hager, 2016; Koh, Lim, Rahimi, Woo, & Gao, 2021; Yan, Mu, Wang, Ranjan, & Zomaya, 2020) are a family of feed-forward models with causal dilated convolutions that encode spatiotemporal information locally. The structure usually consists of multi-level convolutions, where the input of current level are from the output of the previous level. The cells in the current level capture longer time window contexts than those from the previous level. The output of the final level captures high-level temporal relationships across the whole input sequence, which is usually fed into a classification layer to recognize time series classes.

Transformer (Vaswani et al., 2017) adopts the mechanism of self-attention, differentially weighting the significance of each part of the input sequence. Specifically, it collects a sequence of short-term information over a certain period of time, then applies a temporal self-attentive module to enhance some parts of the sequence while diminishing other parts, in order to learn the final nonlinear information to recognize the whole time series. It was originally designed to handle Natural language processing (*NLP*) problems but now widely used in time series prediction and classification (Li et al., 2019; Cai, Janowicz, Mai, Yan, & Zhu, 2020; Oh, Wang, & Wiens, 2018; Rußwurm & Körner, 2020; Zerveas, Jayaraman, Patel, Bhamidipaty, & Eickhoff, 2021).

Under a common closed set assumption: the training and testing data are drawn from the same label and feature spaces, deep neural networks have already achieved significant success in a variety of time series classification tasks in many industrial domains (Zerveas et al., 2021; Geng, Huang, & Chen, 2020). In the real world, however, data distributions shift over time in a complex, dynamic manner. Even worse, new concepts (e.g. new categories of objects) can be presented to the model at any time. Such distribution shift and unseen concepts both may lead to catastrophic failures since the model still attempts to make predictions based on its closed-world assumption (Hsu, Shen, Jin, & Kira, 2020). In addressing general classification problem, besides recognizing samples from known classes, labeling something new, novel or unknown should always be a valid outcome (Bendale & Boult, 2015). This leads to what is sometimes called “open set” recognition, in comparison to systems that make closed world assumptions or use “closed set” evaluation (Scheirer, de Rezende Rocha, Sapkota, & Boult, 2012).

Although recent researches start to focus on open set recognition (Kaur et al., 2022; Yoshihashi et al., 2019; Geng et al., 2020; Sun, Yang, Zhang, Ling, & Peng, 2020; Joseph, Khan, Khan, & Balasubramanian, 2021; Fang, Lu, Liu, Liu, & Zhang, 2021; Frittoli, Carrera, Rossi, Fragneto, & Boracchi, 2022; Chambers & Gaber, 2022), they have the following disadvantages:

1. They can only detect samples that possibly come from unknown classes, but are **incapable of further distinguishing different unknown classes**. Specifically, given k known classes, open set recognition approaches output probability vectors in $\mathbb{R}^{k+1}$ space where the additional entry describes the possibility of samples drawn from outside of known classes.

2. They are designed for either image or text data, but **not for time series classification in industrial domains**.

In this work, we propose an end-to-end deep-learning based approach for industrial time series learning applications, which can classify time series from known classes and distinguish time series among different new classes.

3. PROBLEM SETTING AND MODEL ARCHITECTURE

3.1. Notation and Problem Setting

In modern industrial systems, time series are collected by sensors with high frequency (e.g. $100Hz$ sampling rate on wind turbines). A time series with m variables and n timestamps [1] is noted as $X \in \mathbb{R}^{m \times n}$. Given a set of k known classes $\{C_1, C_2, ..., C_k\}$, a labeled training time series is noted as $\langle X, Y \rangle$, where $Y \in \{1, 2, ..., k\}$. We assume that there are u unknown classes $\{C_{k+1}, ..., C_{k+u}\}$, where u is predefined or estimated by domain experts.

In this work, we consider the following problem setting: given input data that consists of n_u unlabeled time series $\{X_1, ..., X_{n_u}\}$ and n_k labeled time series $\{\langle X_1, Y_1 \rangle, \langle X_2, Y_2 \rangle, ..., \langle X_{n_k}, Y_{n_k} \rangle\}$ where $Y_i \in \{1, 2, ..., k\}$, our model aims to assign each unlabeled time series to any of the $k + u$ classes.

3.2. Model Architecture

When the classification model encounters samples from novel (unknown) classes, each novel class should be incorporated into the learning process respectively, in order to not only distinguish unknowns from knowns but also distinguish different patterns within unknowns. In this work, we train networks with multi-paths autoregression of input time series. Similar to typical autoencoder setting, our model also tries to reconstruct the input time series. But it is a special type of autoencoder in that our model consists of one general encoder, and $k + u$ decoders that correspond to k known and u unknown classes, as shown in Figure 1. The general encoder learns representation of input time series so as to preserve information useful for separating different classes. Each decoder learns the reconstruction way for one class, either from knowns or unknowns. Once the training is complete, each unlabeled time series is labeled by the smallest residual among the $k + u$ reconstructions.

3.2.1. Encoder

The first part of our model is a general encoder to acquire the nonlinearly transformed features from the input time series. The encoder is designed as a temporal convolutional network (*TCN*).

TCN is originally proposed in (Oord et al., 2016) and popularly applied in various sequence modeling tasks e.g. (Bai, Kolter, & Koltun, 2018; Franceschi, Dieuleveut, & Jaggi, 2019). Different from recurrent neural network, it is not a recursive structure therefore suffer less from gradient vanishing issues (Bai et al., 2018). To be adaptable to length-varying inputs and capable to learn more nonlinearity, *TCN* usually consists of multiple levels. As shown in Figure 1, here we apply multi-level *TCN* on the input time series X. In this design, the input of one level are from the output of the previous level. At the end, the output of the last *TCN* level captures high-level temporal relationships across the whole input time series.

On each *TCN* level we design five steps: a temporal (*1D*) convolution filter, a Batch Normalization, a scaled exponential linear unit (*SELU*), a dropout function, and an average pooling operator.

3.2.2. Decoders

As shown in Figure 1, the second part of our model is decoder part to reconstruct the input time series from encoder output. It consists of $k + u$ decoders that correspond to k known and u unknown classes respectively. Similar to encoder part, each decoders is designed as a multi-level *TCN* but in a deconvolved way.

We apply five steps on each level: a temporal (*1D*) deconvolution filter (or transposed convolution operator), a Batch Normalization, a scaled exponential linear unit (*SELU*), a dropout function, and an upsampling operator. However, it is worth to mention that the last level of each decoder only contains a temporal (1D) deconvolution filter without the other four steps. Given input time series as X, the output of the j-th decoder is noted as $\hat{X}^{(j)}$, where $j \in \{1, 2, ..., k + n\}$.

3.2.3. Loss function and Real Time Decision

The final residual is obtained by:

$$
\begin{aligned}
loss = & \frac{1}{n_u} \sum_{i=1}^{n_u} \min_{j \in \{1,..,k+u\}} \|X_i - \hat{X_i}^{(j)}\|_2^2 \\
& + \frac{1}{n_k} \sum_{i=1}^{n_k} \underset{Y_i = C_j \text{ and } j \in \{1,...,k\}}{\|X_i - \hat{X_i}^{(j)}\|_2^2} .
\end{aligned} \tag{1}
$$

Equation (1) consists of two parts: the first part accounts the unlabeled time series where the residuals come from the minimum reconstruction error among all $k + u$ decoders' output; while the second part considers the labeled time series of which residuals come from the j-th decoder if that time series belong to C_j ($j \in \{1, ..., k\}$).

At real time, each (unlabeled) time series is assign to the class

[1] For notational convenience, we assume all time series are with the same length. However, this model can deal with time series of different lengths by using padding techniques.

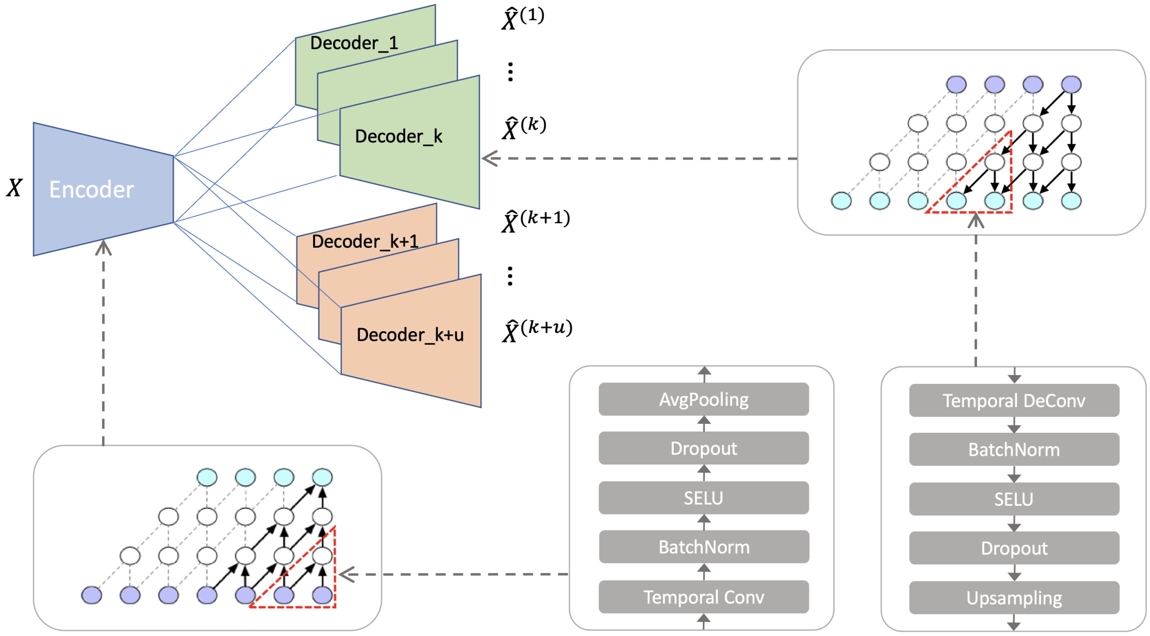


Figure 1. Our *PSTC* model framework. It consists of one general encoder and $k + u$ decoders that correspond to k known and u unknown classes respectively.

that provides the minimum residual j:

$$\hat{Y}_i = argmin_{j\in\{1,..,k+u\}} ||X_i - \hat{X}_i^{(j)}||_2^2. \qquad (2)$$

This design enhances the learned representation so as to preserve information for not only separating unknowns from knowns, but also discriminating classes in both knowns and unknowns. Our novel partially supervised learning utilizes latent representations learned from general encoder for reconstruction and enables robust unknown detection and grouping without harming the known-class classification accuracy.

4. EXPERIMENT SETUP

4.1. Baselines

In the experiment, we compare our Partially Supervised Time Series Classification (*PSTC*) with three popular time series classification approaches using deep neural networks:

- *LSTM-FCN*: In their work (Karim, Majumdar, Darabi, & Harford, 2019) Karim et.al. proposed Long Short Term Memory Fully Convolutional Network, which is the first work on multivariate time series classification by combining *LSTM* with fully convolutional network. The authors extended the squeeze-and-excite block to the case of *1D* sequence models that augments *LSTM* to enhance classification accuracy.
- *TCN*: Temporal convolutional networks (Fawaz, Forestier, Weber, Idoumghar, & Muller, 2019) consist of three parts: firstly, the networks compute low-level features from input signals using convolutional filters that encode spatial-temporal information; secondly, the model feed these low-level features into higher convolutional levels to extract more nonlinear features; and a softmax operator is applied in the last layer for classification.
- *TST*: Time Series Transformer was proposed (Zerveas et al., 2021) for the first time for multivariate time series representation for classification. The framework includes a pre-training scheme, which the authors show that it can offer substantial performance benefits over fully supervised learning, even without leveraging additional unlabeled data.

In our experiments, we apply grid-search for hyperparameter selections to each baselines on a separate validation set of labeled data. The final selections are based on the minimum loss.

To have a fair comparison, we also combine **out-of-distribution** (***OOD***) with *TCN*. That is, any OOD time series that detected by (Kaur et al., 2022) will be assigned as new class (with class index $k+1$), while all the in-distribution (*iD*) will be assigned class labels by *TCN*.

4.2. Evaluation Metrics

Since the classification targets include both known and unknown classes, we use two metrics to evaluate the learning result: ***accuracy*** and ***NMI*** (normalized mutual information).

Accuracy is a statistical measure of how well a classification test correctly identifies or excludes a condition. In other words, it is the proportion of correct predictions. When computing accuracy in multiclass classification, it is simply the fraction of correct classifications:

$$accuracy = \frac{1}{n_k + n_u} \sum_{i=1}^{n_k+n_u} \mathbf{1}(Y_i = \hat{Y}_i), \tag{3}$$

where $\mathbf{1}(*)$ is the indicator function. Accuracy is presented on a range from 0 to 1 where a score of 1 is reserved for the perfect predictions. Since accuracy measurement requires alignment between the true and predicted classes, we apply a permutation function to the predicted labels: each predicted class is assign to the most common truth labels among data points within that predicted class.

NMI is a variant of a common measure in information theory called Mutual Information. Mutual Information accounts to the "amount of information" one can extract from a distribution regarding a second one. *NMI* is a normalization of the Mutual Information score that normalized by generalized mean of entropy of ground truth and predicted labels: :

$$NMI = \frac{2 \times I(Y; \hat{Y})}{H(Y) + H(\hat{Y})}, \tag{4}$$

where Y are ground truth and $\hat{Y}$ are predicted labels, $H(*)$ measures entropy and $I(*, *)$ measures mutual information. *NMI* is between 0 (no mutual information) and 1 (perfect correlation), and it is independent of the label permutations of the clusters therefore it is popularly used in clustering evaluation.

5. EXPERIMENTS

5.1. Experiments on Wind Turbine Failure Dataset (WF)

To conduct a systematic comparison between our *PSTC* and the baselines on partially supervised setting, we perform experiments on a confidentially labeled data set collected from wind turbines. This data set includes 137 time series collected from different offshore wind turbines. Each time series is associated with one of four failure classes. The size of these four classes are: 18, 28, 35, and 56.

All time series have $100Hz$ sampling rate and six input variables that record properties of the wind turbines such as the generator speed and degree of rotor position, etc. The lengths of these time series are all 8000 (timestamps).

In our experiments, we assume two scenarios: 1) we have labeled time series from two classes, and the other two are unknown without any labeled sample; 2) three classes are known and have labeled time series, and the left one is unknown.

More formally, for the first scenario, we test thorough combination of two known classes (a two-combination from the four classes). In each selection, we randomly choose 10%, 20%, 30% and 40% time series from the selected two known classes as labeled samples, while the rest of known and all the samples from unknown are unlabeled. The performance are evaluated on all unlabeled samples.

The experiment result are shown in Figure 2a and 2b. Obviously, all of the three popular classification baselines have poor performance ($< 30\%$ in *NMI* and $< 45\%$ in accuracy of our *PSTC*) due to their incapacity of perceiving unknown classes. In other words, they can only return false positives when they see samples from unknown classes. To have better understanding, we also include two additional baselines: a fully supervised *TCN* of which labeled data cover all of the four classes, and an *OOD+TCN* that detects *OOD* time series and classify those *iD* time series. We can see that even though our *PSTC* only know information from half of classes, it achieves similar performance ($> 90\%$) with fully supervised *TCN*. On the other hand, the classification quality from *OOD+TCN* is worse because it cannot further distinguish different new classes.

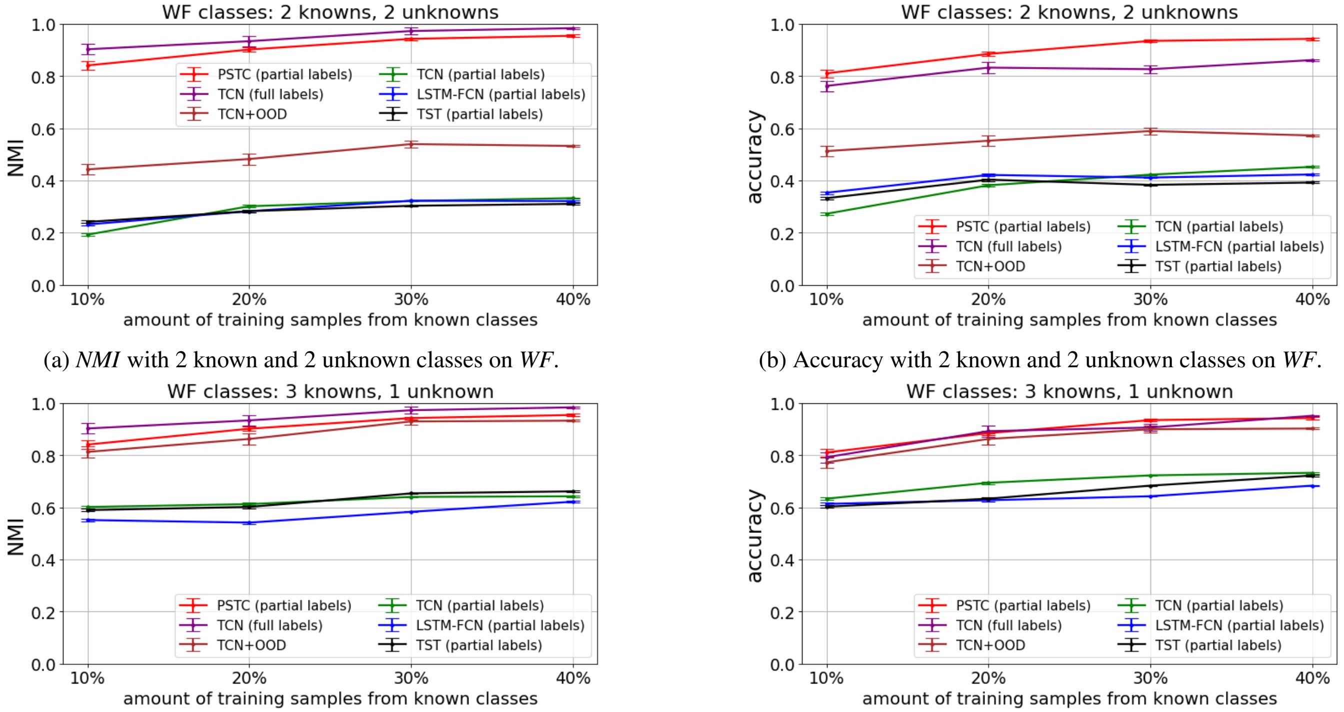
For the second scenario, we also test different combination of three known classes. The results are shown in Figure 2c and 2d. Although there is only one unknown class, our *PSTC* still outperforms three baselines with $+45\%$ *NMI* ($+30\%$ accuracy) on average. Again, our performance is very close to that from fully supervised *TCN*. It is worth noting that *OOD+TCN* has similar performance with our *PSTC* because there is only one unknown class in this scenario.

In this experiment, the structure of our *PSTC* is detailed as follows:

- The encoder has four levels: the convolution operator of first level has 6 input channel and 32 output channels; the convolution operator of the second level projects channel space from 32 to 64; the third projects to 128 and the fourth projects to 256 channels. All levels have the same kernel size as 3, average pooling size as 2, and stride and dilation sizes are all 1.
- The decoders have four levels: the deconvolution operator of first level projects input channel 256 to output channel 128; the second level projects channel space from 128 to 64; the third projects to 64 and the last level projects to original input space 6.

5.2. Experiments on Tennessee Eastman Process (TEP)

Tennessee Eastman Process (*TEP*) is essentially a realistic simulation of a chemical process that has been widely used in process control studies. It was modeled computationally in 1993 by Downs and Vogel (Downs & Vogel, 1993). The dataset is consistently used for comparing and benchmarking algorithms in industrial time series learning (Yin, Ding, Haghani, Hao, & Zhang, 2012). The entire process contains



(a) *NMI* with 2 known and 2 unknown classes on *WF*.

(b) Accuracy with 2 known and 2 unknown classes on *WF*.

(c) *NMI* with 3 known and 1 unknown classes on *WF*.

(d) Accuracy with 3 known and 1 unknown classes on *WF*.

Figure 2. Classification performance comparison on Wind Turbine Failure (*WF*) dataset

52 different variables that record properties of the system such as the flowrates, pressures, temperatures, levels, mole fractions and compressor power outputs. The data are sampled every 3 minutes for 25 hours. The *TEP* dataset used here includes five failure class, each class has 100 time series and each time series has 500 timestamps.

Here we also assume two scenarios: 1) we have labeled time series from three classes, and the other two are unknown without any label; 2) two classes are known with labeled time series, and the other three are unknown.

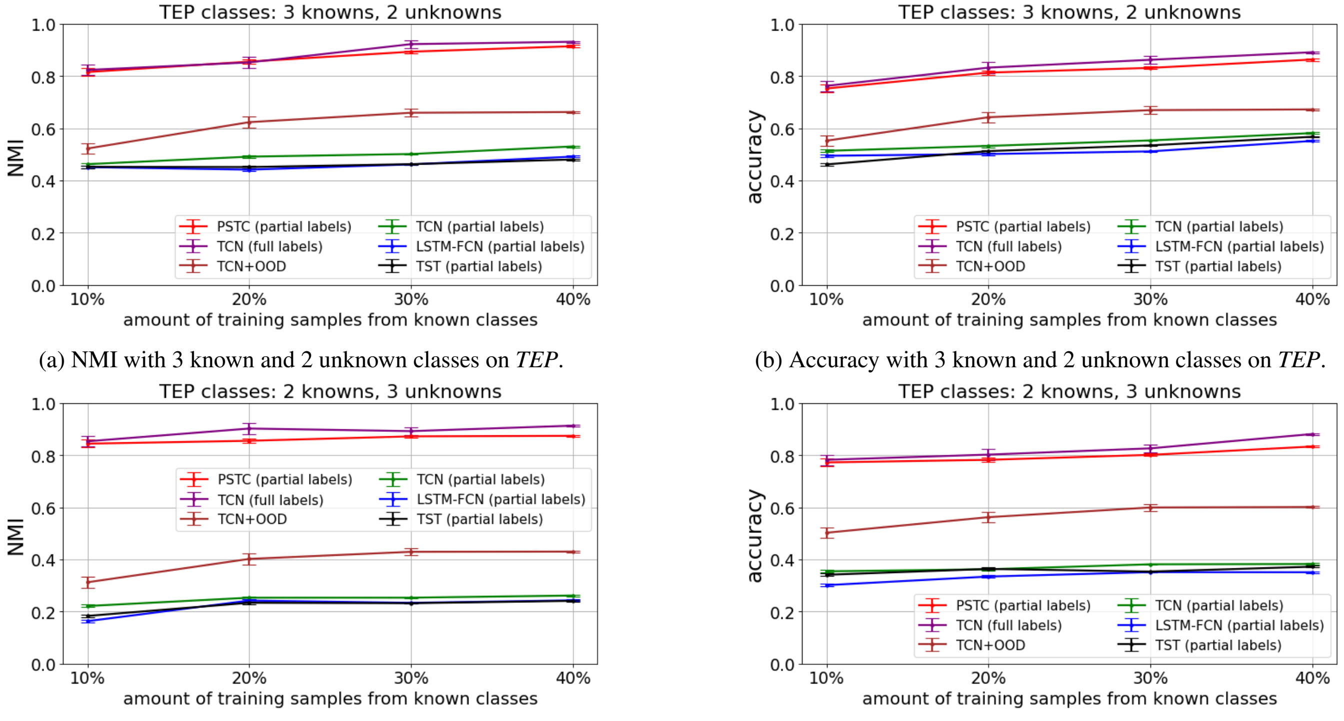
The experiment result are shown in Figure 3. We can see that when the number of unknown classes increase, the performance of baselines become worse. In contrast, our *PSTC* maintains reasonable performance due to its capability to discover and distinguish both known and unknown classes. Moreover, it has comparable result with fully supervised baseline *TCN*.

In this experiment, the structure of our *PSTC* is detailed as follows:

- The encoder has three levels: the convolution operator of first level has 52 input and 200 output channels; the convolution operator of the second level projects channel space from 200 to 300 channels; the third projects to 500 channels. All levels have the same kernel size as 3, average pooling size as 2, and stride and dilation sizes are all 1.
- The decoders have three levels: the deconvolution operator of first level projects 500 input to 300 channels; the second level projects channel space from 300 to 200 channels; the third projects to original input space 52.

6. CONCLUSION

In industrial applications, classification problems usually evolve over time, which require classifiers that can incorporate novel classes of data. When the classification system encounters a novel class, that class should be incorporated into the learning process. However, existing deep learning based classifiers rely on neural networks that trained in a fully supervised manner; this causes specialization of learned representations to known classes only and makes it hard to distinguish different unknown classes. In order to classify samples from both known and unknown classes, we design a deep neural networks with multi-paths autoregression of input time series data. This enhances the learned representation so as to preserve information for separating unknowns from knowns, as well as discriminating classes within either knowns or unknowns. Our novel partially supervised learning utilizes latent representations learned from general encoder and enables robust unknown detection and grouping by multiple decoders without harming the classification accuracy on knowns. Extensive experiments reveal that the proposed approach outper-



(a) NMI with 3 known and 2 unknown classes on *TEP*.

(b) Accuracy with 3 known and 2 unknown classes on *TEP*.

(c) NMI with 2 known and 3 unknown classes on *TEP*.

(d) Accuracy with 2 known and 3 unknown classes on *TEP*.

Figure 3. Classification performance comparison on *TEP* dataset

forms existing deep learning based classifiers in synthetic and real world time series datasets. One drawback of this work is that it requires the number of unknown classes is known in advance (or from "best guess" by domain experts). In the future work, we aim to automatically determine this number in a data-driven way.

REFERENCES

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Bendale, A., & Boult, T. (2015). Towards open world recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1893–1902).

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157–166.

Cai, L., Janowicz, K., Mai, G., Yan, B., & Zhu, R. (2020). Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, *24*(3), 736–755.

Chambers, L., & Gaber, M. M. (2022). Deepstreamos: Fast open-set classification for convolutional neural networks. *Pattern Recognition Letters*.

Dhamija, A. R., Günther, M., & Boult, T. (2018). Reducing network agnostophobia. *Advances in Neural Information Processing Systems*, *31*.

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

Fang, Z., Lu, J., Liu, A., Liu, F., & Zhang, G. (2021). Learning bounds for open-set learning. In *International conference on machine learning* (pp. 3122–3132).

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, *33*(4), 917–963.

Fink, O., Wang, Q., Svensen, M., Dersin, P., Lee, W.-J., & Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence*, *92*, 103678.

Franceschi, J.-Y., Dieuleveut, A., & Jaggi, M. (2019). Unsupervised scalable representation learning for multivariate time series. *arXiv preprint arXiv:1901.10738*.

Frittoli, L., Carrera, D., Rossi, B., Fragneto, P., & Boracchi, G. (2022). Deep open-set recognition for silicon wafer production monitoring. *Pattern Recognition*, *124*, 108488.

Geng, C., Huang, S.-j., & Chen, S. (2020). Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *43*(10), 3614–3631.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hsu, Y.-C., Shen, Y., Jin, H., & Kira, Z. (2020). Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 10951–10960).

Joseph, K., Khan, S., Khan, F. S., & Balasubramanian, V. N. (2021). Towards open world object detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 5830–5840).

Karim, F., Majumdar, S., Darabi, H., & Harford, S. (2019). Multivariate lstm-fcns for time series classification. *Neural Networks*, *116*, 237–245.

Kaur, R., Sridhar, K., Park, S., Jha, S., Roy, A., Sokolsky, O., & Lee, I. (2022). Codit: Conformal out-of-distribution detection in time-series data. *arXiv preprint arXiv:2207.11769*.

Koh, B. H. D., Lim, C. L. P., Rahimi, H., Woo, W. L., & Gao, B. (2021). Deep temporal convolution network for time series classification. *Sensors*, *21*(2), 603.

Lea, C., Vidal, R., Reiter, A., & Hager, G. D. (2016). Temporal convolutional networks: A unified approach to action segmentation. In *European conference on computer vision* (pp. 47–54).

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., & Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, *32*.

Oh, J., Wang, J., & Wiens, J. (2018). Learning to exploit invariances in clinical time-series data using sequence transformer networks. In *Machine learning for healthcare conference* (pp. 332–347).

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., … Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Perera, P., & Patel, V. M. (2019). Deep transfer learning for multiple class novelty detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 11544–11552).

Rezaeianjouybari, B., & Shang, Y. (2020). Deep learning for prognostics and health management: State of the art, challenges, and opportunities. *Measurement*, *163*, 107929.

Rußwurm, M., & Körner, M. (2020). Self-attention for raw optical satellite time series classification. *ISPRS journal of photogrammetry and remote sensing*, *169*, 421–435.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., & Boult, T. E. (2012). Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, *35*(7), 1757–1772.

Sun, X., Yang, Z., Zhang, C., Ling, K.-V., & Peng, G. (2020). Conditional gaussian distribution learning for open set recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 13480–13489).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Yan, J., Mu, L., Wang, L., Ranjan, R., & Zomaya, A. Y. (2020). Temporal convolutional networks for the advance prediction of enso. *Scientific reports*, *10*(1), 1–15.

Yin, S., Ding, S. X., Haghani, A., Hao, H., & Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process. *Journal of process control*, *22*.

Yoshihashi, R., Shao, W., Kawakami, R., You, S., Iida, M., & Naemura, T. (2019). Classification-reconstruction learning for open-set recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 4016–4025).

Yucesan, Y. A., Dourado, A., & Viana, F. A. (2021). A survey of modeling for prognosis and health management of industrial equipment. *Advanced Engineering Informatics*, *50*, 101404.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., & Eickhoff, C. (2021). A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th acm sigkdd conference on knowledge discovery & data mining* (pp. 2114–2124).

Zhang, L., Lin, J., Liu, B., Zhang, Z., Yan, X., & Wei, M. (2019). A review on deep learning applications in prognostics and health management. *Ieee Access*, *7*, 162415–162438.