# Anomaly Detection with Memory-Enhanced Composite Neural Networks for Industrial Control Systems

Boyang Xia, *Member, IEEE,* Yuxin Zhang, *Member, IEEE,* Zhiwen Pan, *Member, IEEE,* Rui Yao, *Member, IEEE,* Yiqiang Chen, *Member, IEEE,* and Yuting He, *Member, IEEE*

***Abstract*—Advanced Industrial Control Systems (ICS) applications will introduce grand security challenges due to an increase in the attack surface, which raises accumulated concerns in recent years. As one of the promising intrusion detection scheme for ICS, anomaly detection for industrial process data can detect the anomalous ICS operational process caused by active attacks through monitoring and analyzing the industrial process data. However, industrial process data inherently contains sophisticated nonlinear spatiotemporal correlations which are hard to be explicitly described by existing baseline models for anomaly detection. In addition, the detection rate of anomaly detection approaches are affected by the disturbances within normal process data and lack of attack data for training. In this paper, we propose an anomaly detection approach for industrial process data based on Memory-enhanced Composite Convolutional Long Short-Term Memory (Conv-LSTM) Encoder-Decoder (MCCED). To perform fine-grained unsupervised learning, the MCCED model is designed to concurrently perform the reconstruction analysis and prediction analysis on process data stream, and is designed to be trained in an end-to-end fashion based on normal process data. To explicitly describe the spatiotemporal correlations within process data, the Conv-LSTM unit are adopted to form network layers within the MCCED model. To deal with the disturbances within normal process data, a two-stage memory enhancing mechanism is designed to prevent the MCCED model from learning the trivial patterns consisted within the training data. We conducted extensive experiments on two benchmark ICS cybersecurity datasets to demonstrate the effectiveness of our approach.**

***Index Terms*—Auto-encoder, cybersecurity, industrial control systems, intrusion detection, machine learning, time series anomaly detection**

## I. Introduction

SEVERAL years have witnessed an exponential growth of the Cyber-Physical System (CPS), which plays the essential role in the Fourth Industrial Revolution. As a typical and significant paradigm of CPS, the Industrial Control Systems (ICS), is a supporting component a critical infrastructure [1]. These critical utilities, such as water treatment systems, power grids and oil refineries, supply efficient water treatment, smart generation and transmission and high-demanding energy management.

Previously, these Industrial Critical Systems were proprietary systems and isolated from the Internet. However, the emerging industrial applications (such as Enterprise Resource Planning Systems and Manufacturing Execution Systems) have made ICS assets inextricably connected with the enterprise network. Consequently, the interconnected ICS assets have been exposed to the similar vulnerabilities as the conventional IT (Information Technology) assets [2, 3]. To be specific, the attack surfaces of ICS networks are expanding due to the following reasons: 1) ICS communication protocols are vulnerable to protocol attacks since these protocols were originally designed without taking security as a major concern; 2) operating systems of ICS assets are vulnerable to attacks such as virus and Trojan because of legacy isolation assumption; 3) increased connectivity between ICS assets reduce difficulty of conducting Internet-based attack [1].

In recent years, a set of attacks were successfully launched against real-world infrastructures, which make people realize the significance of ICS security. For instances, in 2015, a massive power outage hit the Ukraine, investigated as a result of a cyberattack, which left around 230,000 users without power for hours [4]. In 2013, hackers accessed the core command-and-control system of a dam in Rye Brook, New York successfully [5]. A majority of the existing attacks aim at compromising the integrity of process data (e.g. sensor readings, control signals, actuator states) being exchanged among ICS assets or the integrity of asset operating systems [6]. Once process dataflow or the assets themselves are compromised, the physical process of ICS can be corrupted or even manipulated. Therefore, a promising way to protect ICS networks from attacks against integrity is utilizing Intrusion Detection Systems (IDS) based on industrial process analysis. Specifically, through monitoring and analyzing industrial process data, the IDS based industrial process analysis can detect (or even predict) the anomalous physical process within ICS in a timely manner [7–9].

In general, Intrusion Detection Systems can be categorized as either signature based IDS and anomaly based IDS. Signature based IDS detect anomaly behaviors based on predefined signatures for typical anomalous process events. However, the signature based IDS are disadvantageous for their heavy demands on the domain expert knowledge and high False Negative Rate (FNR), poor capability in recognition of novel attacks [10]. On the other hand, anomaly based IDS [9] build baseline models to describe the patterns of

* The first two authors contributed equally.

Y. Zhang, Z. Pan, and Y. Chen are with Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China and University of Chinese Academy of Sciences, Beijing, China.

E-mail: {zhangyuxin, pzw, yqchen}@ict.ac.cn.

normal behaviors so that any behaviors that deviate from the normal patterns will be regarded as anomaly behavior. Since the zero-day attacks for ICS networks are emerging in recent years, anomaly based IDS are becoming more and more impressive for its ability of detect zero-day attacks. However, the existing anomaly detection approaches for industrial process data have the following drawbacks: 1) For specification based methods, many of baseline models are built manually or by mathematical modelling of complex physical processes, which requires plenty of human efforts and prior knowledge of the target system [11, 12]. 2) For machine learning based methods, the baseline models can be built by modelling complex patterns in an operational training dataset in a data-driven manner. The ICS operational datasets are usually highly imbalanced which means normal data samples are significantly more than anomalous data samples, hence the baseline models generated by supervised machine learning algorithms cannot achieve promising detection accuracy and their detection performances on novel attacks are questionable. 3) As a typical kind of multivariate time series data, ICS process data has two characteristics: the sensor readings and actuator states are cross-correlated (spatial-correlated) with each other by following a certain control logic, the sensor readings and actuator states are deeply auto-correlated (temporal-correlated) by following a historical and periodical trend. These spatiotemporal correlations within the ICS process data are hard to be explicitly described by a baseline model. 4) The normal industrial process data inherently contains noises and disturbances, these noises and disturbances contain trivial patterns that are somehow resemble to anomaly patterns. If a baseline model learns these trivial patterns during the training phase, its detection performance cannot be guaranteed.

To address these challenges, this paper presents an anomaly detection approach for industrial process data based on a Memory-enhanced Composite Convolutional Long Short-Term Memory (Conv-LSTM) Encoder-Decoder (MCCED) model. The MCCED model consists of four major components: 1) an encoder to extract features from the process data input and store sequential information in the hidden states; 2) a reconstruction decoder to calculate the reconstruction error for the data input, by reconstructing data input based on the hidden states; 3) a prediction decoder to calculate the prediction error for the input data, by predicting the data input based on the hidden states; 4) a two-stage memory enhancing mechanism to enhance both the feature extraction of encoder and the predication capability of prediction decoder, by further characterizing the hidden states through addressing prototypical elements in two matrices in manner of attention mechanism. Moreover, each layer of encoder and decoder is embedded with Conv-LSTM units, so that the spatiotemporal correlation of normal process data attributes can be explicitly described. The operational process of the proposed anomaly detection approach is as fol-lows. During the training phase, the MCCED model is trained with normal process dataset, so that a baseline model which can explicitly describe the normal pattern of industrial physical process is generated. Note that, the training is performed in an end-to-end fashion (namely end-to-end training) with one objective function, so that the performance of four components within the MCCED model can be jointly optimized. During the detection phase, the MCCED model calculate the reconstruction error and prediction error for each captured process data sample. Through combining the two errors as a composite anomaly score, a fine-grained anomaly detection decision can be made. To summarize, the major contributions of this paper are as follows:

- A composite encoder-decoder model which can concurrently perform the reconstruction analysis and prediction analysis on industrial process data, and a corresponding anomaly score formation for fine-grained anomaly detection;
- An end-to-end unsupervised training scheme for composite encoder-decoder model which can perform training based on normal process data and jointly optimize the composite model with one objective function;
- A two-stage memory enhancing mechanism which is robust to process disturbances by preventing the model from describing the trivial patterns consisted within the training data;
- The utilization of Conv-LSTM unit which can explicitly describe spatiotemporal correlations within the process data;
- Empirical studies on two well-recognized ICS security datasets, which demonstrate the superior performance of our approach over existing baselines and state-of-the-art approaches.

The rest of this paper is organized as follows. Section II introduces the background on building blocks of the proposed approach and summaries the previous works regarding anomaly detection for ICS data. In Section III, the proposed Memory-enhanced Composite Conv-LSTM Encoder-Decoder model is elaborated. In Section IV we conduct the extensive empirical studies and discuss the experimental results. The conclusion and future directions are presented in Section V.

## II. Background and Related Work

### A. Industrial Control Systems

Industrial Control Systems (ICS) are critical to operational technology sector. It includes systems for monitoring and controlling industrial processes. Usually, ICS collects sensors readings and operational states of controllers during the process. After that, it analyzes and exhibits them for system operators and generates control logic in a local or remote control structures [7]. ICS are typically managed through a monitoring and data acquisition system, SCADA (Supervisory Control and data Acquisition). It provides the operator with a graphical user interface for easy observation of system status, receiving any alarms suggesting illegimate operations, or entering system adjustments to manage controlled processes. The key components of ICS include SCADA Servers, Remote Terminal Units (RTUs), Programmable Logic Controllers (PLCs), Human Machine Interfaces (HMIs), Data Historians, etc [7].

Intrusion Detection Systems(IDS) are indispensable for traditional ICS firewall solutions. In ICS, IDS are devices or application programs or their combinations inspecting the behaviors in the system, detecting anomalous activities or

policy violations by extracting and analyzing all kinds of industrial data (e.g. system logs, ICS protocol traffic, sensor measurements and actuator commands) [8].

### B. ICS Threats

Compared with conventional IT systems, the characteristics of ICS system bring different risks and priorities [3]. Firstly, the attacks targeting ICSs potentially have more severe impacts which include risks to human lives, damages to the environment, financial losses, etc. Secondly, safety and efficiency are two major concerns for the design and operation of ICS, while the goals of efficiency and safety can sometimes conflict with security due to the time delay and resource consumption bringing by the security measures. Thirdly, ICS networks are usually heterogeneous networks which may include legacy assets, legacy operating systems, different protocols (even proprietary protocol), diverse network topologies. These differences between ICS and IT systems create the need for increased sophistication in applying cybersecurity measures [3]. By integrating a set of cybersecurity measures which are tailored to ICS applications, the ICS should be protected from the violations of confidentiality, integrity, and availability.

**Attacks against confidentiality** aim at obtaining undisclosed information and knowledge through gaining unauthorized access to the data or services. These attacks are regarded as passive attacks since they do not interfere the operations of victim system. However, the obtained information and knowledge may be used to plan active attacks. Typical attacks in this category include eavesdropping, traffic analysis, port scanning, etc.

**Attacks against availability** aim at denying the availability of ICS assets and services. The attacks can be launched by either obstructing the data transmission or disabling the ICS assets. Typical attacks in this category include jamming attack, buffer overflow, worm attack, Trojan attack, etc. Since the violation of ICS availability can be somehow reflected by the anomalous ICS operations, the anomaly detection scheme for industrial process data is a potential mitigation measure for attacks against availability.

**Attacks against integrity** aim at deceiving the victim system with falsified data. The attacks can be launched in three manners: 1) tampering existing packets in transit; 2) injecting packets into the traffic; 3) compromising the integrity of asset locally through gaining privilege. Typical attacks in this category include man-in-the-middle attack, spoofing attack, replay attack, insider attack, worm attack, Trojan attack, etc. The attacks against integrity can cause severe impacts since the normal operation of a victim system can be interfered or even manipulated. Recently, a more powerful stealthy attack has been discovered [9], which has been proved effective for bypassing intrusion detection schemes. The stealthy attacks are launched by professional attackers with prior knowledge of the victim ICS systems. By leveraging the physical constraints of ICS operation, attackers can inject well designed false data into ICS in a long period of time by following the expected behavior of the system closely, and finally corrupt the victim ICS. The Secure Water Treatment(SWaT) ICS security benchmark dataset [13] is well-recognized for its description of stealthy attacks. With prior knowledge of the testbed (Secure Water Treatment(SWaT) Testbed[14]), dataset designers designed four attack scenarios against integrity to corrupt the physical process of ICS (see Table I), where the Multi-Stage attacks can be regarded as stealthy attacks since the attacks are consisted of multiple phases to gradually temper process data attributes. The proposed anomaly detection approach in this paper aims at detecting the attacks against integrity.

TABLE I
ATTACK SCENARIOS IN SWAT DATASET

| Scenario | Description |
|---|---|
| SSSP Attack | Launched on exactly one point |
| SSMP Attack | Launched on two or more attack points but on only one stage |
| MSSP Attack | Similar to an SSMP attack except that now the attack is performed on multiple stages |
| MSMP Attack | Similar to SSMP attack but performed in two or more stages |

SSSP: Single Stage Single Point
MSSP: Multi Stage Single Point
SSMP: Single Stage Multi Point
MSMP: Multi Stage Multi Point

### C. Related Work

In the context of securing ICS network, the anomaly detection for industrial process data is an emerging research area which attracts increasingly attention. The well-recognized benchmark datasets include Secure Water Treatment (SWaT) dataset [13], Water Distribution Testbed (WADI) dataset [15], Gasoil Heating Loop (GHL) dataset [16], etc. The existing intrusion detection approaches can be classified into three categories: rule based and model based approaches where manually specified process specifications or mathematical modelling of the physical process are used as a basis to detect attacks [17, 18], supervised learning based approaches where the baseline models are generated with supervised learning algorithm based on both normal and abnormal process data [19, 20], unsupervised learning based approaches where the baseline models are generated with one-class classification algorithm merely based on normal data. The drawback of rule based and model based approaches consist in that they demand the expert knowledge of target ICS system and require a huge amount of labor efforts. Although the supervised learning based approach can automatically generate baseline model in a data driven manner, since the abnormal process data for training is limited, the baseline models cannot achieve promising detection accuracy and their detection performance on novel attacks are questionable. Note that, based on this concern, most of the existing benchmark datasets do not include anomaly process data in their datasets for training [13, 15, 16]. Therefore, the latest trend is to develop unsupervised learning based anomaly detection approach for industrial process data. Initially, the classic one-class classification algorithms such as the OCSVM [21], the PCA [22] and the K-means [23] is utilized to generate baseline models. However, these classic algorithms are incapable of describing the non-linear cross-correlations (namely

spatial correlation) among data attributes. Therefore, anomaly detection approaches based on Convolutional Neural Network (CNN) were proposed [24, 25], so that the non-linear cross correlation among data attributes can be recognized through abstracting the local spatial patterns of process data stream with convolutional kernels. In [24], Kravchik et al. proved that the 1D-CNN model outperforms the conventional auto-encoder and other classic algorithms on anomaly detection task in the SWaT dataset. In [25], Chandy et al. integrated the convolutional layers into variational auto-encoder, and performed anomaly detection by calculating the reconstruction error. The drawback of these CNN based approaches is that the long-term auto-correlation (namely temporal correlation) of each data attribute cannot be effectively recognized. In order to effectively recognize the temporal correlations within process data, plenty of existing approaches utilize Recurrent Neural Network (RNN) to generate baseline models, where Long Short-Term Memory(LSTM) networks are the most popular choices [12, 26–28]. Most of the RNN based methods perform anomaly detection through calculating the prediction error [12, 26–28]. Two types of RNN based approaches include integrating LSTM layers into either auto-encoders [29] or Generative Adversarial Network (GAN) [30, 31]. In [29], a feedforward dimensionality-reduction layer was integrated as an input layer with the LSTM auto-encoder, and the anomaly score was generated based on reconstruction error. In [30] and [31], LSTM was integrated into GAN to capture temporal correlations, and the anomaly score was generated by combining reconstruction errors and discrimination errors. However, the source code provided by [30] and [31] performs detection with huge computational delay, due to the high computational complexity of GAN.

To explicitly describe the spatiotemporal correlations within process data, a recent trend is to adopt Conv-LSTM network which is originally proposed by Shi et al. in [32]. Two well-recognized works are the work [33] which is published in AAAI'19 and the work [34] which is published in KDD'19. In [33], Zhang et al. combined attention based Conv-LSTM network with convolutional auto-encoder to formalize a composite anomaly detection model, where the anomaly score is calculated based on reconstruction error. In [34], a Conv-LSTM network and Probabilistic Principal Component Analyzers are trained separately to generate two baseline models, the prediction error and reconstruction error generated by the baseline models were combined to form a composite anomaly score. A drawback of [33] is that, since the proposed baseline model is a sophisticated model consisted of deep neural layers, such model may describe trivial patterns within the normal training samples, which result in degradation of the detection rate [35, 36]. As to the drawback of [34], the joint performance of the two baseline models cannot reach the global optimum, since two models are trained separately. Moreover, the composite anomaly detection mechanism which combines reconstruction error with prediction error cannot fully mitigate the same high false negative rate problem as appeared in [34].

Compared with these previous works, the proposed anomaly detection approach makes the following contributions: 1) Instead of simply adopting the Conv-LSTM network, we embed Conv-LSTM layers into encoder-decoder to develop a Conv-LSTM encoder-decoder; 2) A composite Conv-LSTM encoder-decoder is proposed to concurrently calculate both prediction error and reconstruction error to generated composite anomaly score, so that the normal disturbances and attacks can be better discriminated(compared with single anomaly score); 3) The composite baseline model is generated based on end-to-end training(training with one global objective function), so that the parameters of all the model components are jointly trained to reach the global optimum; 4) Inspired by the attention mechanism and Memory-enhanced Neural Networks, a novel two-stage memory enhancing mechanism is proposed to prevent the encoder-decoder from learning trivial normal patterns, so that false negative rate and false positive rate of the baseline model can be reduced.

### *D. Encoder-Decoder based Anomaly Detection*

Encoder-Decoder model is a kind of neural network architecture which consist of two components: encoder network and decoder network. Specifically, the encoder network maps the data inputs into latent space so that the data inputs can be abstracted as latent vectors; the decoder network analyze the latent vectors and generate data outputs. The analyses that can be perform by decoder include reconstruction analysis and prediction analysis. Hence, the encoder-decoder based anomaly detection can also be further classified as either reconstruction based anomaly detection and prediction based anomaly detection.

The reconstruction based anomaly detection use decoder network to reconstruct the data input based on the latent vectors, and the residual between data input and reconstructed data output is called the reconstruction error (namely reconstruction residual). During the training phase, the encoder-decoder model is trained with dataset composed of pure normal data samples, so that the reconstruction errors of the model for normal data samples can be minimized. The real-time anomaly detection is performed based on the assumption that the models can produce lower reconstruction errors for normal data and higher reconstruction errors for abnormal data. Reconstruction based anomaly detection can be performed with Auto-encoder, Variational Auto-encoder, etc. The operations of reconstruction based anomaly detection are follows:

$$\boldsymbol{h} = \text{Encoder}(\boldsymbol{x}) \tag{1}$$

$$\hat{\boldsymbol{x}} = \text{Decoder}(\boldsymbol{h}) \tag{2}$$

where the encoder network generates a latent representation $\boldsymbol{h}$ from input $\boldsymbol{x}$, and the decoder network reconstructs the output $\hat{\boldsymbol{x}}$ from this representation $\boldsymbol{h}$.

The prediction based anomaly detection use decoder network to predict the current data input based on the latent vectors describing the previous data input, and the residual between the current data input and predicted data output is called the prediction error (namely prediction residual). During the training phase, the encoder-decoder model is trained with dataset composed of pure normal data samples, so that the

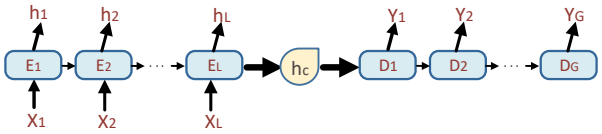



Fig. 1. LSTM based Encoder-Decoder Model.

prediction errors of the model for normal data samples can be minimized. During the real-time anomaly detection, if the current prediction error is higher than expected, alarm will be triggered. The prediction based anomaly detection can be preformed with RNN based encoder-decoder, LSTM based encoder-decoder, etc. Taking the LSTM based encoder-decoder model as an example (see Fig. 1), given input sequence $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L\}$, $\boldsymbol{h}_j$ is the hidden state of encoder at the $j$-th time step for each $j \in \{1, 2, \cdots L\}$, where $\boldsymbol{h}_j \in \mathbb{R}^c$, $c$ is the unit number of a LSTM layer. The final encoding product is context vector $\boldsymbol{h}_c$, that is, the final state $\boldsymbol{h}_L$ of the encoder. Before decoding, $\boldsymbol{h}_c$ is used as the initial state for the decoder. The $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_G\}$is the decoder output of the model. The operations of whole LSTM encoder-decoder can be formulated as (3)-(4).

$$\boldsymbol{h}_c = \text{Encoder}(\boldsymbol{X}) \tag{3}$$

$$\boldsymbol{Y} = \text{Decoder}(\boldsymbol{X}) \tag{4}$$

Thereof the encoding and the decoding processes are both finished in a step-by-step fashion as follows:

$$\boldsymbol{h}_j = f\left(\boldsymbol{h}_{j-1}, \boldsymbol{x}_j; \boldsymbol{\theta}\right) \tag{5}$$

where $\boldsymbol{\theta}$ are the paremeters of the LSTM network. The encoder and decoder are jointly trained to predict the target sequence.

Since the Encoder-Decoder model is a kind of deep neural network which combines data data dimension reduction, feature extraction, and anomaly detection together, the Encoder-Decoder based anomaly detection has proved to be very effective when dealing with high-dimensional sequential data with non-linear correlations. However, as a deep neural network model, the encoder-decoder model faces the "over-fitting" problem, which means that the model is more likely to learn the trivial patterns (e.g. noisy sensor readings, process disturbance) consisted within the training dataset. During the real time detection, the over-fit encoder-decoder based model may generate smaller (reconstruction or prediction) residual for anomalous data inputs whose patterns are somehow similar to these trivial patterns, which result in False-Negative detection results. To solve this problem, two contributions are made in this paper: 1) The proposed Encoder-Decoder model is an composite model which can concurrently perform reconstruction based anomaly detection and prediction based anomaly detection. The residual sequence generated by the proposed model is a composite residual which combines both reconstruction residual and prediction residual, so that the two types of residual can be complementary to each other. 2) Inspired by the attention mechanism, a two-stage memory enhancing mechanism is proposed to map the original process

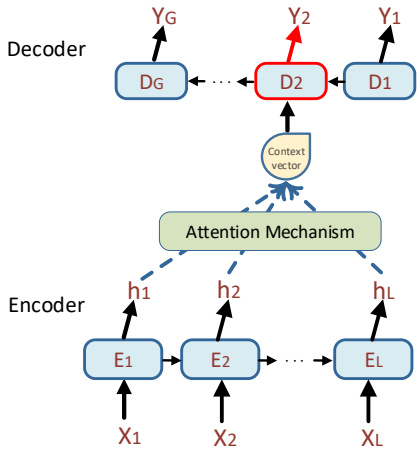



Fig. 2. Attention Mechanism in Encoder-Decoder Model.

data input as similarity matrix. In this way, the Encoder-Decoder model cannot directly get access to the trivial information consisted within the process data. The background knowledge of attention mechanism is introduced in the next section.

### *E. Attention Mechanism for Encoder-decoder Model*

The attention mechanism for encoder-decoder model is originally appeared in Natural Language Processing (NLP) field for the task of Neural Machine Translation[37]. The objective of attention mechanism is to enhance the prediction performance of decoder by concatenating a context vector to map the latent vector $\boldsymbol{h}$ (which is generated by encoder) as weighted vector $\boldsymbol{c}$ (namely context vector), so that the latent vector will be assigned with higher weight. The map is a weighted average of all the components within the latent vector. The non-trivial vector components are assigned with higher weights. By analyzing the context vector(instead of the original latent vector), the decoder can put more efforts on analyzing the non-trivial vector components [38].

The encoder-decoder model with attention mechanism is shown in 2. Given the sequential data input for each time step of the decoder, the context vector is the weighted average of the hidden states generated by the encoder:

$$\boldsymbol{c}_i = \sum_{j=1}^{L} w_{ij} \cdot \boldsymbol{h}_j \tag{6}$$

The weight $w_{ij}$ is calculated from the matching degree between previous hidden state $\boldsymbol{h}_{i-1}$ and the hidden state of each time step $\boldsymbol{h}_j$ produced by the Score$(\cdot)$ function.The Score$(\cdot)$ function can be designed in several manners, such as "dot" (inner product), "general", and "concat", which was elaborated in [38].

$$w_{ij} = \frac{\exp\left(\text{Score}\left(\boldsymbol{h}_{i-1}, \boldsymbol{h}_j\right)\right)}{\sum_{j=1}^{L} \exp\left(\text{Score}\left(\boldsymbol{h}_{i-1}, \boldsymbol{h}_j\right)\right)} \tag{7}$$

For process data and natural language data are both sequential data with temporal correlation, in recent years, some research on anomaly detection algorithms have begun to apply attention mechanism [33, 39].

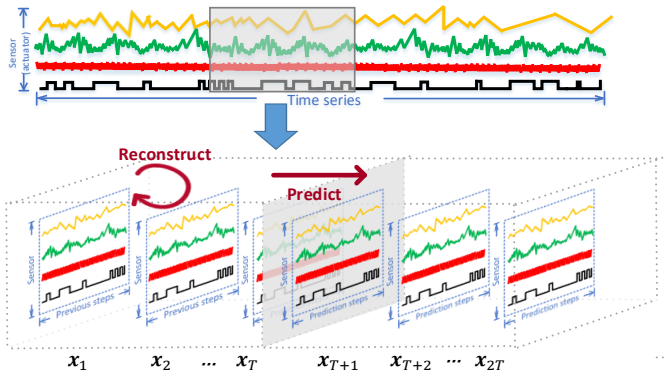



Fig. 3. Fragmenting method of raw data.

In Natural Language Processing (NLP) field, the Neural Machine Translation task has been greatly improved by integrating the attention mechanism into the encoder-decoder architecture. By the attention mechanism, the decoder can focus on a small set of states most related to decoding process. In recent years,

### F. Convolutional LSTM Unit

The seminal Conv-LSTM network architecture was developed by Shi et al. in [32]. The Convolutional LSTM (Conv-LSTM) uses convolution operation to replace the weight matrix within the Fully-Connected Long Short Term Memory network (FC-LSTM) unit. The FC-LSTM layer compresses 2D spatiotemporal information into 1D as input, which causes a loss of overall spatial information [32]. To alleviate this issue, [32] replaced the weight matrix connected to the inputs $\boldsymbol{x}$ and the hidden states $\boldsymbol{h}$ in the FC-LSTM with multiple convolution filters to extract local features in manner of sparse connection. The whole formulation of the Conv-LSTM unit is summarized in equation (8)-(12),

$$\boldsymbol{i}_t = \sigma\left(w_{xi} * \boldsymbol{x}_t + w_{hi} * \boldsymbol{h}_{t-1} + w_{ci} \circ \boldsymbol{c}_{t-1} + b_i\right) \tag{8}$$

$$\boldsymbol{f}_t = \sigma\left(w_{xf} * \boldsymbol{x}_t + w_{hf} * \boldsymbol{h}_{t-1} + w_{cf} \circ \boldsymbol{c}_{t-1} + b_f\right) \tag{9}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \tanh\left(w_{xc} * \boldsymbol{x}_t + w_{hc} * \boldsymbol{h}_{t-1} + b_c\right) \tag{10}$$

$$\boldsymbol{o}_t = \sigma\left(w_{xo} * \boldsymbol{x}_t + w_{ho} * \boldsymbol{h}_{t-1} + w_{co} \circ \boldsymbol{c}_t + b_o\right) \tag{11}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh\left(\boldsymbol{c}_t\right) \tag{12}$$

where $*$ denotes convolution operator, $\circ$ denotes Hadamard product, $\sigma$ is sigmoid function, $w_{xi}, w_{hi}, w_{ci}, w_{xf}, w_{hf}$ are convolutional kernels connected to input and hidden states, $b_i, b_f, b_c, b_o$ are bias parameters. In contrast to LSTM units, all inputs $\boldsymbol{x}_t$, cell states $\boldsymbol{c}_{t-1}, \boldsymbol{c}_t$, the hidden states $\boldsymbol{h}_{t-1}, \boldsymbol{h}_t$, the gates $\boldsymbol{i}_t, \boldsymbol{f}_t, \boldsymbol{o}_t$ change from 2D vectors into 3D tensors (the latter two dimensions carry spatial information).

## III. Methodology

### A. Prerequisites

Since the proposed model is using Conv-LSTM neuron (instead of LSTM neuron) for process data analysis, the data input of the model should be 3 Dimensional tensor instead of 2 Dimensional tensor. The formation of 3 Dimensional tensor is as follows. Given the process data: $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \in \mathbb{R}^{m \times N}$, where $m$ is set as number of sensors, and $N$ denotes the number of time points. First, we generate sequence segments by sliding window of size $L$. Next, we divided each segment into two set of sub-segments including $T$ previous sub-segments and $T$ current sub-segments, the size of each sub-segment is $L/2T$. The $T$ previous subsegments $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ are used as the input of encoder and the ground truth of reconstruction residual. The $T$ current subsegments $\{\boldsymbol{x}_{T+1}, \boldsymbol{x}_{T+2}, \ldots, \boldsymbol{x}_{2T}\}$ are used as the ground truth of prediction residual. In this way, the 3D tensor is formalized as a 3-dimensional tuple: `(subsegment, time stamp, sensor dimension)` $\in \mathbb{R}^{T \times l \times m}$. Output of the proposed model are reconstruction residuals corresponding to each data samples within the previous sub-segments and prediction residuals corresponding to each data samples within the current sub-segments. Fig. 3 shows the whole fragmenting process.

### B. Architecture of Memory-enhanced Composite Conv-LSTM Encoder-Decoder

The proposed model is a Composite Encoder-Decoder model with three sub-models: 1) a encoder to perform feature extraction by abstracting the historical process data (namely previous sub-segments) as latent vectors; 2) a reconstruction decoder to calculate the reconstruction residual by reconstructing the latent vectors as the the historical process data; 3) a prediction decoder to calculate the prediction residual by predicting the values of current process data (namely current sub-segments) based on the latent vectors. In order to prevent the model from learning trivial patterns such as sensor noise and process disturbance, a Two-stage Memory-enhanced mechanism(MEM) is proposed: 1)the first-stage Memory-enhanced mechanism is designed for encoder, to remove the trivial information from the latent vectors; 2) the second-stage Memory-enhanced mechanism is designed for prediction decoder, to prevent the prediction decoder from make unbounded prediction. The complete model architecture is depicted in Fig. 4. The following sub-sections are organized as follows: Next we will elaborate on the structure and components of the model.

**Composite Model** In this composite model, the encoder is responsible to encode the information of input sequences into latent representation stored in the last hidden state. The latent representation is output to the reconstruction decoder and the prediction decoder respectively. Given a process segment $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{2T}\} \in \mathbb{R}^{2T \times l \times m}$, the encoder is input with the first $T$ segment $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$, the reconstruction decoder is to compare the output with the current input in reverse $\{\boldsymbol{x}_T, \boldsymbol{x}_{T-1}, \ldots, \boldsymbol{x}_1\}$ and the prediction decoder is to compare the output with future input $\{\boldsymbol{x}_{T+1}, \boldsymbol{x}_{T+2}, \ldots, \boldsymbol{x}_{2T}\}$. The reconstruction and prediction decoder are trained to reconstruct and predict input sequences with as low errors as possible. The loss function of composite model in training process should be like (when using mean squared error as

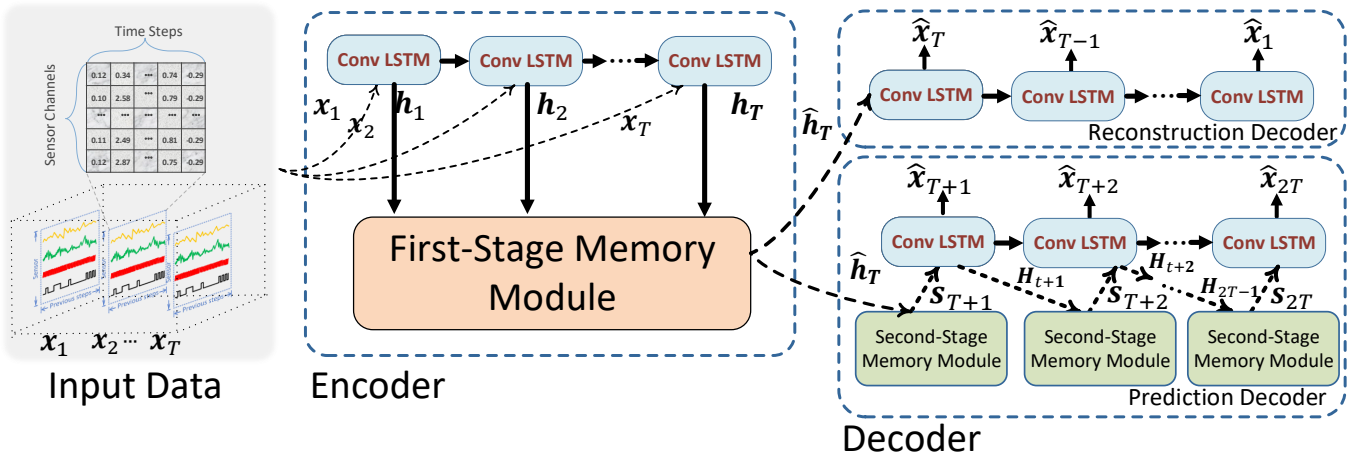



Fig. 4. Two-stage Memory-enhanced Composite Encoder-Decoder.

general loss function):

$$\mathcal{L}(X) = \sum_{i=1}^{N} \left( \sum_{t=1}^{T} \left\| \widehat{\boldsymbol{x}}_t^* - \boldsymbol{x}_t^* \right\|_F^2 + \sum_{t=T+1}^{2T} \left\| \widehat{\boldsymbol{x}}_t^\dagger - \boldsymbol{x}_t^\dagger \right\|_F^2 \right) + \Omega, \quad (13)$$

where $N$ is the number of samples, $\|\cdot\|_F$ is Frobenius norm, $\widehat{\boldsymbol{x}}_t^*$ is the reconstructed value at time $t$, $\boldsymbol{x}_t^*$ is the input in reverse. And $\widehat{\boldsymbol{x}}_t^\dagger$ is the predicted value at time $t$, $\boldsymbol{x}_t^\dagger$ is the groundtruth value in future. $\Omega$ is the regularization term which will be formulated in Section III-C.

**Two-stage Memory-enhanced Encoder-Decoder** Low discrimination between normal and abnormal sequence in encoder-decoder is an important challenge for unsupervised anomaly detection tasks. This is due to the fact that, the training data containing sensor noise and process disturbance will inevitably be learnt by the encoder-decoder, which causes the latent vector $h$ carrying too much trivial information. During the encoding process, we solve this problem through introducing the First-stage Memory Module (hereinafter called *1st MEM*) into encoder (as shown in Fig. 5 ). The operational principle of *1st MEM* is that, the latent vector $h$ is represented as a rectified vector $M_t$, which describes the similarity between the latent vector $h$ and non-trivial normal patterns. Specifically, the normal patterns is stored in a memory matrix ( implemented as a full connected neural network) which is trained during the training process. The rectified vector is calculated as the cumulative sum of similarities between the latent vector $h$ and all the instances within the memory matrix. Since the a majority of normal patterns within the memory matrix is non-trivial, a hard-shrinkage operation is performed during the similarity calculation, so that trivial similarities is removed from the rectified vector.

Compared with reconstruction analysis, the prediction analysis is more challenging[40]. Specifically, the reconstruction analysis is a reverse-encoding procedure for the latent vector $h$ to regenerate the original data input, and the prediction analysis is an multivariate regression procedure for the latent vector $h$ to estimate the future data changes. Although the encoder with *1st MEM* can suppress the reconstruction and

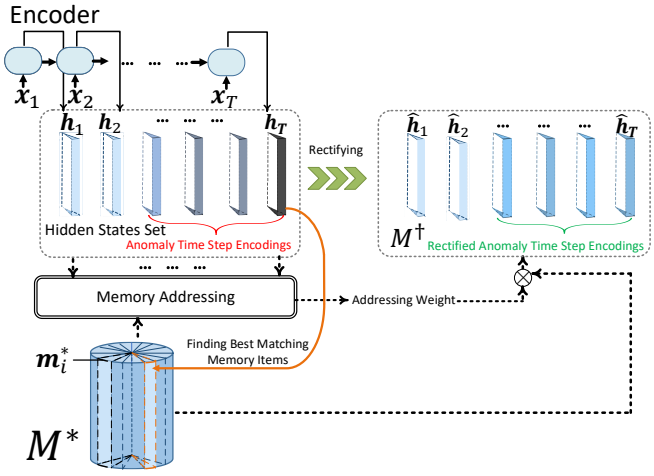



Fig. 5. Encoder with First-stage Memory Module.

prediction ability of decoders for anomaly data, the prediction ability of decoder for normal data may also be suppressed, which increases the false positive rate. This is because the rectified vectors generated by *1st MEM* maybe too sparse to carry enough informative normal patterns. Therefore, we introduce the Second-stage Memory Module (hereinafter called *2nd MEM*) along with Attention Module into prediction decoder ,to further enhance the performance prediction analysis for normal data, as shown in Fig. 6.

*1) Encoder with 1st MEM:*

*a) Dictionary Memory:*

During the data training phase, the Dictionary Memory matrix is designed as a matrix $M^* \in \mathbb{R}^{N \times C}$ to store patterns derived from normal data as memory instances. Here $M^*$ denotes the memory capacity implemented by full-connected layer and $C$ is the hidden dimension of the feature vector derived from flattened latent representation (flattened to shape `(sub-segment, hidden dimension)`$\in \mathbb{R}^{T \times C}$, $C = l \times m$). Each memory instance $\boldsymbol{m}_i \in \mathbb{R}^{1 \times C}, \forall i \in [N]$

represents an informative data sample retained in training.

*b) Content based Memory Addressing:*

Considering the situation of $t$ step encoding, we utilize hidden state $\boldsymbol{h}_t$ and the memory matrix $M^*$ to derive encoding $M_t^\dagger$ at $t$ moment, where $\boldsymbol{h}_t \in \mathbb{R}^C, M^\dagger \in \mathbb{R}^{T\times C}$ and $\boldsymbol{m}_i^*$ is the $i$-th dictionary record of the memory matrix $M_i^*$ for $\forall i \in N$. By utilizing the attention mechanism, the memory reading block quantifies the similarity between latent vector and every memory instance. For a latent vector $\boldsymbol{h}_t$, the corresponding $M_t^\dagger$ can be obtained by reading the Dictionary Memory:

$$M_t^\dagger = \text{Read}^{\text{C}}\left(M^*, \boldsymbol{h}_t\right), \tag{14}$$

where $\text{Read}^{\text{C}}$ is memory addressing operation. Specifically, $M_t^\dagger$ is derived by the the weighted sum of each memory instance $m_i^*$ within $M^*$:

$$M_t^\dagger = \sum_{i=1}^{N} w_{ti}^* \cdot \boldsymbol{m}_i^* \tag{15}$$

where $w_{ti}^*$ is the addressing weight used to address the Dictionary Memory. The addressing weight $w_{ti}^*$ is calculated by the scoring function $\text{Score}(\cdot)$ and normalized by $\text{Softmax}(\cdot)$ function:

$$w_{ti}^* = \frac{\exp\left(\text{Score}\left(\boldsymbol{h}_t, \boldsymbol{m}_i^*\right)\right)}{\sum_{i=1}^{N}\exp\left(\text{Score}\left(\boldsymbol{h}_t, \boldsymbol{m}_i^*\right)\right)} \tag{16}$$

Here we use the rescaled inner product as the $\text{Score}(\cdot)$ function:

$$\text{Score}\left(h_t, m_i^*\right) = \frac{\boldsymbol{h}_t \boldsymbol{m}_i^{*T}}{k} \tag{17}$$

Here $k$ is a scale factor to limit the value of inner product, which is a hyperparameter we will discuss in Section IV. The addressing weight vector $w_{ti}^*$ mainly represents how well the $i$-th record of Dictionary Memory $M^*$ and the hidden state $\boldsymbol{h}_t$ matches.

In spite of rectification effect of the module on trivial patterns, complex memory addressing weight matrix may still involve trivial patterns contained in memory items in memory addressing process [35]. To solve the problem, we improve the sparsity of memory addressing weights via "hard shrinkage" and entropy regularizer method which is proposed in [35]. As shown in (18) and (19), hard shrinkage limits the weights to specific value $\lambda$ or larger, and the weights are normalized by its 1-norm. These operations eliminate the connections to the trivial memory items.

$$\widehat{w}_{ti} = \frac{\max\left(w_{ti} - \lambda, 0\right) \cdot w_{ti}}{|w_{ti} - \lambda| + \varepsilon} \tag{18}$$

$$\widehat{w}_{ti} = \widehat{w}_{ti} / \|\widehat{w}\|_1 \tag{19}$$

Here $\widehat{\text{w}}_{ti}$ denotes the $i$-th weight for the according memory item in $t$-th time step after shrinkage, $\lambda$ denotes the shrinkage threshold, $\varepsilon$ is a very small positive scalar. Entropy regularizer will be formulated in Section III-C.

The Context Memory $\left\{\widehat{\boldsymbol{h}}_1, \widehat{\boldsymbol{h}}_2, \ldots, \widehat{\boldsymbol{h}}_{\text{t}}\right\}$ is defined as the final output of the encoder and the input of two decoders.

*2) Decoder with Attention Module and 2nd MEM:*

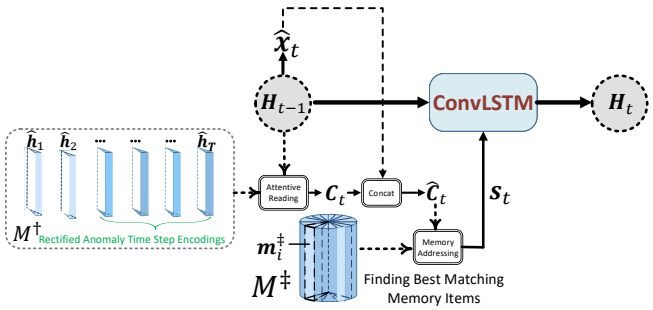



Fig. 6. Operation of A Single Time Step in Decoder with *2nd MEM* and Attention Module.

*a) Attention Module:*

The way that 2nd MEM module and attention module are combined is shown in Fig. 6. Through combining the unbounded memory (provided by the attention module) with the bounded memory (provided by the *2nd MEM* module), the combined modules helps decoder effectively access to the informative prior knowledge for prediction. The combined modules are deployed in all the layers(where one layer correspond to the prediction of one time step) of prediction decoder( as shown in the lower right section of Fig. 4), so that the prediction performance for every single time step can be enhanced. The operating principle of 2nd MEM module is roughly similar to 1st MEM module. The only difference between the two MEM modules is that, in 2nd MEM module ,the rectified vector is calculated by taking the context vectors (which is generated by attention mechanism) as input. Correspondingly, the memory matrix of 2nd MEM module stores the normal patterns of context vectors, instead of the latent vectors. Operational procedures of the combined module are as follows.

At time step $t$, context memory $M^\dagger$ is first read by hidden state $\boldsymbol{H}_{t-1}$ of attention module to generate context vector $\boldsymbol{C}_t$.

$$\boldsymbol{C}_t = \text{Read}^{\text{Atten}}\left(M^\dagger, \boldsymbol{H}_{t-1}\right) \tag{20}$$

According to attention mechanism, the calculation process of $\text{Read}^{\text{Atten}}$ is unrolled as:

$$\boldsymbol{C}_t = \sum_{j=1}^{L} w_{tj} \cdot \boldsymbol{m}_j^\dagger \tag{21}$$

$$w_{tj} = \frac{\exp\left(\text{Score}\left(\boldsymbol{H}_{t-1}, \boldsymbol{m}_j^\dagger\right)\right)}{\sum_{j=1}^{L}\exp\left(\text{Score}\left(\boldsymbol{H}_{t-1}, \boldsymbol{m}_j^\dagger\right)\right)} \tag{22}$$

where $\boldsymbol{m}_j^\dagger$ is a memory item of context memory $M^\dagger$ corresponding to the $j$-th time step. $\boldsymbol{H}_{t-1}$ is a hidden state generated at $t$-1-th step in decoding. $w_{tj}$ is the alignment weight between the hidden state at the $t$-1-th moment of prediction decoder and the $j$-th context memory item generated by encoder. It determines whether decoder needs to give a strong focus on the information of the $j$-th time step in the encoder when decoding the information of the $t$-th time step.

Here we use the rescaled inner product as the Score$(\cdot)$ function as the same as that in *1st MEM*.

$$\text{Score}\left(\boldsymbol{H}_{t-1}, \boldsymbol{m}_j^{\dagger}\right) = \frac{\boldsymbol{H}_{t-1}\boldsymbol{m}_j^{\dagger T}}{k} \tag{23}$$

Then $\boldsymbol{C}_t$ is merged with $\widehat{\boldsymbol{x}}_{t-1}$ to get the conditioned context vector $\widehat{\boldsymbol{C}}_t$ by concatenation in channel dimension.

$$\widehat{\boldsymbol{C}}_t = \text{Concat}\left(\boldsymbol{C}_t, \widehat{\boldsymbol{x}}_{t-1}\right), \tag{24}$$

where $\widehat{\boldsymbol{x}}_{t-1}$ denotes the output of decoder at $t$-1-th moment.

*b) Memory Enhancing in 2nd MEM:*

The Transition Memory $M^{\ddagger}$ is the core module similar to The Dictionary Memory in structure. Functionally, it is used to store the external memory necessary for prediction which is ignored by the Conv-LSTM unit in the prediction decoder. The conditional context vector $\widehat{\boldsymbol{C}}_t$ is retrieved as a query in the Transition Memory, which generates memory state $\boldsymbol{s}_t$. Then $\boldsymbol{s}_t$ is input into the Conv-LSTM unit for further prediction to generate the current hidden state $\boldsymbol{H}_t$ in prediction decoder.

$$\boldsymbol{s}_t = \text{Read}^{\text{T}}\left(M^{\ddagger}, \widehat{\boldsymbol{C}}_t\right) \tag{25}$$

$$\boldsymbol{H}_t = \text{ConvLSTM}\left(\boldsymbol{H}_{t-1}, \boldsymbol{s}_t\right) \tag{26}$$

$\text{Read}^{\text{T}}$ is as the same as the Dictionary Memory addressing block structure in the encoder with *1st MEM*. We use the content based addressing method to read the Transition Memory. The weight generation and synthesis process should be formulated as:

$$\boldsymbol{s}_t = \sum_{i=1}^{N} w_{ti}^{\ddagger} \cdot \boldsymbol{m}_i^{\ddagger} \tag{27}$$

$$w_{ti}^{\ddagger} = \frac{\exp\left(\text{Score}\left(\widehat{\boldsymbol{C}}_t, \boldsymbol{m}_i^{\ddagger}\right)\right)}{\sum_{i=1}^{N}\exp\left(\text{Score}\left(\widehat{\boldsymbol{C}}_t, \boldsymbol{m}_i^{\ddagger}\right)\right)} \tag{28}$$

where $w_{ti}^{\ddagger}$ is the addressing weight vector, $\mathbf{m}_i^{\ddagger}$ is the i-th Transition memory record. We use the same Score$(\cdot)$ function as that in encoder with *1st MEM*. We also implement hard shrinkage operation to derived weights as the same as in *1st MEM*.

### C. Training Encoder-Decoder

Given training samples $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{2T}\}$ which can be divided into past fragment $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ and future fragment $\{\boldsymbol{x}_{T+1}, \boldsymbol{x}_{T+2}, \ldots, \boldsymbol{x}_{2T}\}$. The loss function can be calculated as:

$$\begin{aligned}\mathcal{L}(\boldsymbol{X}) = \sum_{i=1}^{N}\left(\sum_{t=1}^{T}\left\|\widehat{\boldsymbol{x}}_t^{*} - \boldsymbol{x}_t^{*}\right\|_F^2 + \sum_{t=T+1}^{2T}\left\|\widehat{\boldsymbol{x}}_t^{\dagger} - \boldsymbol{x}_t^{\dagger}\right\|_F^2\right) \\ +\alpha\left\|\boldsymbol{h}_{Enc}\right\|_1 + \beta E(W_*) + \beta E(W_{\ddagger})\end{aligned} \tag{29}$$

where $\widehat{\boldsymbol{x}}_t^{*}$ is the reconstructed value and $\widehat{\boldsymbol{x}}_t^{\dagger}$ is the predicted value at time $t$. $H_{Enc}$ refers to the last hidden state in the encoder. $W_*$ is the addressing weights of Dictionary Memory and $W_{\ddagger}$ is the addressing weights of Transition Memory. $\alpha$, $\beta$ are the weights that control the importance of according regularization term. $\|\cdot\|_F$ is Frobenius norm and $\|\cdot\|_1$ is $\ell 1$ norm.

Through encoder with *1st MEM*, the normal patterns can be stored in Dictionary Memory. When the model learns a lot of noise (trivial patterns) in normal training set due to excessive model complexity, it will reduce the rectification effect of the encoder with *1st MEM* on the hidden states set with *1st MEM* on the hidden states set may be weakened. So we reduce the complexity of encodings via $\ell 1$ norm of last hidden state in encoder, and reduce the complexity in addressing weight $W^*$, via entropy regularizer borrowed from [35], formulated in (30). We also use entropy regularizer in the Transition Memory $W^{\dagger}$ for the same reason.

$$E\left(\widehat{w}_{ti}\right) = \sum_{i=1}^{T} -\widehat{w}_{ti} \cdot \log\left(\widehat{w}_{ti}\right) \tag{30}$$

### D. Anomaly Score Formulation

We calculate the residual between the predicted vectors and the ground truth to infer whether the system state is abnormal at the moment. If the real situation is normal at this moment and is consistent with predicted and reconstructed result, the residual value will be relatively low. Oppositely, if the true status is abnormal, and the predicted and reconstructed result is normal, the prediction error will increase, which means at the moment system is deviated from the normal status.

We normalize the similarity metric at each moment to 0-1 range, called the Anomaly Score, which characterizes the abnormal probability. The larger the anomaly score is, the greater the probability at the moment being determined as abnormal status, and vice versa. We specify a certain value $\tau$. When the abnormal score is higher than value $\tau$, an abnormal event (that is, an attack) occurs in the system at this moment. As shown in the following formula:

$$\begin{cases}\text{Anomaly Score} \ \geq \tau, \text{Attacked} \\ \text{Anomaly Score} < \tau, \text{Normal}\end{cases} \tag{31}$$

The threshold characterizes the degree of confidence in the anomaly. Therefore, distinguishing the abnormal points from the normal ones effectively counts on two factors: the calculation of the similarity metric between the true value and the predicted value, and the reasonable selection of the threshold. We will introduce threshold selection in hyperparameter selection of Section IV. Given the points in the multi-dimensional time series (which can be expressed as the $m$-dimensional vector $\boldsymbol{x}$, where $m$ is the number of sensors) and its predicted value $\widehat{\boldsymbol{x}}$, we can measure the anomaly score by calculating the similarity.

$$AnomalyScore = f(\boldsymbol{x}, \widehat{\boldsymbol{x}}) \tag{32}$$

### E. Anomaly Decision and Smoothing Method

There are some irregularities and abrupt state changes in trends of some sensors and actuators (e.g. pump on/off state). Although such deviations exist, none of them signify a cyberattack event. Inspired by [25], in order to reduce false alarms on short-term deviations, we give anomaly decision in a relatively smoothing manner. We judge the data of an time point as anomaly only when that the anomaly score has been

**Algorithm 1:** Anomaly Decision and Smoothing Method

**Input:**
Anomaly Score $A_t$ of time points in Test Set ($t \in [0 : N_{\text{te}}]$ ), Threshold $\tau$, Tolerance Factor $\omega$
**Output:**
Anomaly Decision $De_t$ of time points in Test Set ($De_t \in \{0, 1\}$)
**Initialization:** Set Starting Counting Point $f_{\text{start}} = -1$, End Counting Point $f_{\text{end}} = -1$
**for** $t = 1 : 1 : N_{te}$ **do** Decision Loop
  **if** $A_t \geq \tau$ **then**
    **if** $f_{start} = -1$ **then**
      $f_{start} \leftarrow t, f_{\text{end}} \leftarrow t$
    **else**
      $f_{end} \leftarrow t$
    **end**
  **else**
    **if** $f_{end} != -1$ **then**
      **if** $f_{end} - f_{start} > \omega$ **then**
        **for** $t = f_{start} : 1 : f_{end}$ **do** Labelling
          $De_t \leftarrow 1$
        **end**
      **else**
        $De_t \leftarrow 0$
      **end**
    **else**
      $De_t \leftarrow 0$
    **end**
    $f_{start} \leftarrow -1, f_{\text{end}} \leftarrow -1$
  **end**
  $t \leftarrow t + 1$
**end**

exceeded the threshold $\tau$ for at least a specified duration time $\omega$, that is, the Tolerance Factor, before current time point. The detailed process of the anomaly decision and smoothing method is presented in Algorithm 1.

The selection of threshold $\tau$ is essential to the evaluation of results. Normally, the threshold selection is based on the anomaly score of the data in Training Set. Since our hypothesis is that anomaly score of abnormal moment in test set should be greater than the normal moment, our method is to multiply the max value of the anomaly score at the Training Set by a coefficient $\theta$.

The size of the tolerance factor has a big impact on the performance of the model. Excessive tolerance factors will result in false negatives of shorter attacks, while too small tolerance factors will result in higher false positives. In order to balance false positives and false negatives, we perform a grid search over tolerance factors from 50 to 300, optimizing for the best F1 to ensure higher confidence. It turns out that the best tolerance factor for SWaT is 250, and for WADI this parameter is 300.

## IV. Empirical Study

### A. Dataset

The dataset we used to evaluate the effectiveness of our approach include SWaT [13] and WADI[15], which are both created by researchers at Singapore University of Technology and Design. . The two datasets consist of process data collected from hardware-based testbeds. The details regarding the two datasets are as follows.

*1) SWaT:* The data was extracted from the SWaT testbed. During the working process, both physical data and network traffic data are collected, but in our research we only use the physical dataset. The testbed continuously runs from an empty state to a full operational state for a total of 11 days, during which in first 7 days system performs normally, and in the last 4 days the system are under both cyberattacks and physical attacks launched by researchers. During the last 4 days, a total of 36 attacks are launched, of which 26 are single-stage single-point attacks, 4 are single-stage multi-point attacks, 2 are multi-stage single-point attacks and 4 are multi-stage multi-point attacks. A detailed description of the attack scenarios is provided in [13].

*2) WADI:* This dataset is based on the WADI testbed. It is an extension of the SWaT system, by taking in a portion of SWaT reverse osmosis permeate and raw water to form a complete and realistic water treatment, storage and distribution network. WADI consists of three main processes: P1 (Primary Grid Water Supply), P2 (Elevated reservoir, Booster Station, Consumer Tank) and P3 (Return Water). For the purpose of anomaly detection, we consider sensor data from P1 and P2 since the return process is only implemented for water recycle purpose. In stages P1 and P2, the data from 82 sensors are collected every second by running WADI non-stop for a total of 16 days, of which 14 days' data are collected under normal operation and 2 days' with xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx attack scenarios. A detailed description of the attack scenarios is provided in [15].

### B. Experiment Settings

We utilize Keras Library[41] with Tensorflow backend as experimental tool and Intel(R) Xeon(R) CPU E5-2637 v4 3.50GHz and 2 NVIDIA GTX 1080Ti graphics cards as experimental environment. Some of the comparative experiments are implemented by utilizing PYOD Tool Box [42].

*1) Hyperparameters:*

*a) Network Architecture:* : In our task, the function of convolution is to extract local temporal pattern and local spatial pattern among sensor channels. After grid search in experiments, we set the number of convolution kernels to be 32, and filter size as $3 \times 3$, similar to many CNN settings. For all the memory modules, we empirically evaluate effects of different capacity of memory among 50, 100, 500, 1000. There appears the most desirable result when capacity equals to 50. For hard shrinkage in two memory enhancing module, the hyperparameter shrinkage threshold $\lambda$ was set to 0.03 after grid search. For scale factor $k$ in two memory modules and

TABLE II
RESULTS OF COMPARATIVE EXPERIMENTS WITH DIFFERENT MODELS

| Method | SWaT | | | WADI | | |
|---|---|---|---|---|---|---|
| | Pre | Rec | F1 | Pre | Rec | F1 |
| PCA [22] | 0.2492 | 0.2163 | 0.2300 | 0.3953 | 0.0563 | 0.1000 |
| OCSVM [12] | 0.9250 | 0.6990 | 0.7963 | N/A | N/A | N/A |
| IForest [43] | 0.1924 | 0.8347 | 0.3127 | 0.1426 | **0.4307** | 0.2143 |
| AE | 0.9410 | 0.7140 | 0.8100 | 0.7841 | 0.4095 | 0.5200 |
| MSCRED [36] | 0.9230 | 0.6428 | 0.7578 | 0.3636 | 0.3832 | 0.3731 |
| MemAE [35] | 0.9796 | 0.7074 | 0.8216 | 0.8010 | 0.4012 | 0.5346 |
| MLP-ED [44] | 0.9670 | 0.6960 | 0.8120 | N/A | N/A | N/A |
| 1DCNN-ED [25] | 0.8670 | 0.8540 | 0.8600 | N/A | N/A | N/A |
| LSTM-ED | 0.9585 | 0.7151 | 0.8191 | 0.8653 | 0.4016 | 0.5486 |
| MAD-GAN [30] | **0.9897** | 0.6374 | 0.7700 | 0.4144 | 0.3392 | 0.3700 |
| **MCCED** | 0.9796 | **0.7964** | **0.8761** | **0.8885** | 0.4042 | **0.5556** |

attention module, we set it equal to the subsegment number described later in Section IV-B1c.

*b) Regularizer:* In terms of last hidden state of the encoder, we set the coefficients of the $\ell 1$ regularization as $\alpha = 1 \times 10^{-8}$ in SWaT and $\alpha = 1 \times 10^{-3}$ in WADI after grid search. And in the memory addressing weight, we select the coefficient of the regularization as $\beta = 2 \times 10^{-4}$, which exactly the same setting in [35].

*c) Data Fragmenting:* We empirically study the effect of different window sizes of (reconstruction or prediction) on the results. It is finally determined that the input/target sequence of length 120 is divided into $6 \times 20$ structures ($m$ denotes the variable numbers). That is, the input and target of the network are the "frames" whose length is 20 and width $m$ is the number of sensor channels. And 6 consecutive "frames" is input to to encoder at one time.

*d) Tolerance Factor:* The size of the tolerance factor has a big impact on the performance of the model. Excessive tolerance factors will result in false negatives of shorter attacks, while too small tolerance factors will result in higher false positives. In order to balance false positives and false negatives, we perform a grid search over tolerance factors from 50 to 300, optimizing for the best F1 to ensure higher confidence. It turns out that the best tolerance factor for SWaT is 250, and for WADI this parameter is 300.

*2) Approaches for Comparison:* In the comparative experiments, we used the following methods to verify the experimental results with many baselines and state-of-the-arts, including the classic baselines for unsupervised anomaly detection and the novel anomaly detection model based on the encoder-decoder model.

**Linear Models** We chose PCA and OCSVM as comparisons. The PCA [22] is an outlier detection method based on the fact that the data does not perform well after being projected in a low-dimensional space. OCSVM [12] finds the optimal single-class boundary by maximizing the margin of the feature space and the margin of the zero.

**Proximity based Models** We chose IForest [43] as representative of proximity based models. IForest detects anomalies by dividing the hyperplane to calculate the number of hyperplanes needed to "isolate" a sample. We conduct experiments following the implement in PYOD library [42].

**Encoder-Decoder based Models** The models are divided into two categories, reconstruction based model and prediction based model. Auto-Encoder [45] is baseline model based on reconstruction. We also use some novel works such as MSCRED [36], MemAE [35] for comparison. For prediction based method, we utilize MLP-ED (Multilayer Perceptron Encoder Decoder) [44], 1DCNN-ED (One-Dimensional CNN Encoder Decoder) [25] and LSTM-ED (LSTM Encoder Decoder) as comparisons.

**Generative Adversarial Networks** We use Generative Adversarial Network (GAN) model in [30] as comparison. [30] utilized the Long-Short-Term-Memory (LSTM) Network as the generator and discriminator in the GAN framework to capture the temporal dependency of time series and detect anomalies based on both reconstruction loss and discrimination loss.

### C. Evaluation Metrics

We use the Precision, Recall, and F1 value of the attack detection to evaluate the performance of model in attack detection tasks.

$$Precision = \frac{TP}{TP + FP} \tag{33}$$

$$Recall = \frac{TP}{TP + FN} \tag{34}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{35}$$

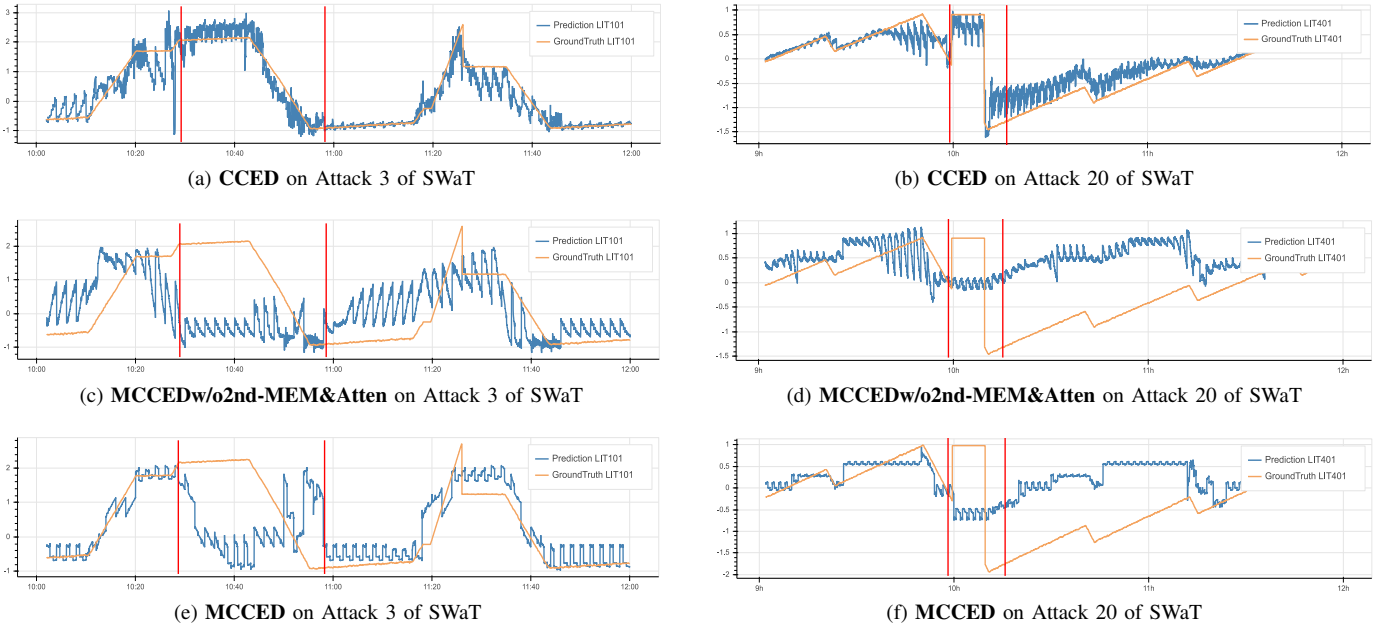
TP refers to true positive, which is correctly identified attack ($Anomaly\ Score > \tau$ and ground truth log is "Attack"). FP is false positive, which is the falsely detected attack ($Anomaly\ Score > \tau$ but ground truth log "Normal"). FN means false negative, which is falsely ignored attack ($Anomaly\ Score < \tau$ but ground truth log is "Attack"). TN means true negative, correctly identified attack ($Anomaly\ Score < \tau$ and ground truth log is "Normal").

### D. Results

Table II summarizes the results on the two datasets using our proposed method (**MCCED**) and compared methods. Obviously, the **MCCED** outperforms the state-of-the-art methods on both datasets. In SWaT dataset, we achieve the highest F1

TABLE III
RESULTS OF COMPARATIVE EXPERIMENTS WITH MODEL VARIANTS

| Method | SWaT | | | | | | WADI | | |
|---|---|---|---|---|---|---|---|---|---|
| | With Attack 23 | | | Without Attack 23 | | | | | |
| | Pre | Rec | F1 | Pre | Rec | F1 | Pre | Rec | F1 |
| CCED Baseline | 0.9432 | 0.7835 | 0.8566 | 0.541 | 0.5557 | 0.5475 | 0.8582 | 0.3768 | 0.5237 |
| MCCEDw/o1st-MEM | 0.9608 | 0.7827 | 0.8627 | 0.5918 | 0.5225 | 0.5551 | 0.8711 | 0.3875 | 0.5363 |
| MCCEDw/o2nd-MEM&Atten | 0.9699 | 0.7845 | 0.8675 | 0.5588 | 0.5555 | 0.5571 | 0.8924 | 0.3763 | 0.5325 |
| MCCEDw/o2nd-MEM | 0.9722 | 0.7874 | 0.8701 | 0.5630 | 0.5586 | 0.5612 | 0.8713 | 0.3918 | 0.5406 |
| MCCEDw/oRecon | 0.9542 | 0.7703 | 0.8524 | **0.6911** | 0.467 | 0.5574 | **0.8989** | 0.3796 | 0.5337 |
| **MCCED** | **0.9796** | **0.7964** | **0.8761** | 0.5653 | **0.6009** | **0.5826** | 0.8885 | **0.4042** | **0.5556** |




(a) **CCED** on Attack 3 of SWaT
(b) **CCED** on Attack 20 of SWaT
(c) **MCCEDw/o2nd-MEM&Atten** on Attack 3 of SWaT
(d) **MCCEDw/o2nd-MEM&Atten** on Attack 20 of SWaT
(e) **MCCED** on Attack 3 of SWaT
(f) **MCCED** on Attack 20 of SWaT

Fig. 7. Prediction Value and Groundtruth Value Curves of Specific Attacked Sensor of different models on detection in Attack 3 and Attack 20 of SWaT. The areas enclosed by red line are anomalies and the rest of area is normal data. **CCED**: Conv-LSTM Composite Encoder-Decoder, **MCCEDw/o2nd-MEM&Atten**: **MCCED** without Second-Stage Memory Module and Attention Module.

value 0.8761. In WADI dataset, we achieve the highest F1 value 0.5520.

Compared with the traditional linear model and similarity estimation based method, it can be seen that the methods based on deep neural networks achieve better performance. Such deep auto-encoder extracts the discriminative deep features when reconstructs the input sequence. Therefore, it has stronger discrimination between normal and abnormal, and provides a strong baseline result.

Compared to the general fully-connected Encoder-Decoder, models such as LSTM, 1DCNN Network are more suitable of capturing time patterns, allowing to better model normal status.

The following state-of-the-art methods considered stronger modeling capability, improvement on discriminate normal and abnormal data, or composite anomaly decision mechanism. Here we summarize the ingenuity of their work briefly.

- Considering the low discrimination between the normal and abnormal samples when reconstructing, MemAE introduces external memory to auto-encoder and solves the problem to some extent [35].
- [36] performs the special preprocessing of multivariate time series, correlation matrix characterization, and structurally modeled the regularity of time series segments. It uses Fully Convolutional Networks (FCN) with Conv-LSTM to extract spatiotemporal features and achieves better results than its baselines through modelling temporal and spatial correlation.
- The MAD-GAN model generates sequence by using LSTM as a generator and determines the anomaly by combining the discrimination error from discriminator and the reconstruction error from generator [30]. Using composite decision critrion (discrimination and reconstruction error) helps the model more robust to detect anomaly.

Because we comprehensively consider the low discrimination between normality and abnormality, good modelling of temporal and spatial correlation and composite decision criterion for anomaly detection can be achieved. Meanwhile we achieve better results than all of the SOTAs. The result

of comparative experiments fully illustrates the efficacy of MCCED. We will explain the effectiveness of our model in detail below.

### E. Ablation Study

In this section, a series of careful comparative experiments among our MCCED and its simplified variants are implemented. The results demonstrate the effectiveness of each component in our architecture. Here we use the Conv-LSTM Composite Encoder Decoder as the base model. All the variants are listed as follows:

- **CCED**: Conv-LSTM Composite Encoder Decoder
- **MCCEDw/o1st-MEM**: MCCED without 1st MEM
- **MCCEDw/o2nd-MEM**: MCCED without 2nd MEM
- **MCCEDw/o2nd-MEM&Atten**: the MCCED model from which the 2nd MEM and Attention Module in the decoder are both removed
- **MCCEDw/oRecon**: MCCED without the Reconstruction Decoder in the decoder part

The experimental results in Table III show that there are different degrees of drops on performance when removing different components above. Below we will conduct an in-depth analysis of effectiveness of these components.

*1) 1st MEM (First-stage Memory Module):*

The removal of the 1st MEM makes F1 score drop from 0.8761 to 0.8627 in SWaT and from 0.5556 to 0.5363 in WADI (see **MCCEDw/o1st-MEM**). The Dictionary Memory plays an important role of rectifying latent representations. Without this module, when anomalous sequence is input, both the reconstructions decoder and the prediction decoder will decode the latent representation containing anomalous information into sequence resembling the inputs, thus getting low prediction and reconstruction errors. Therefore, it is hard to distinguish between normal and abnormal sequence, resulting in a rapid increase on false negatives.

However, the effect is not obvious in SWaT dataset compared to that in WADI. We find that attack 23 in SWaT lasts for 35894 seconds, which accounts for 65.71% of total anomalous points. And attacked sensors in this attack changes obviously, which makes it easier to be detected than other attacks. These facts make attack 23 play a dominant role in the anomaly detection and we infer that attack 23 have a non-ignorable effect among different variants. With this in mind, we attempt to remove the attack 23 from test set and repeat the experiments to verify the hypothesis. As shown in Table II, we find that as the anomaly rate becomes much lower, the F1 score drops drastically. The removal of attack 23 makes the anomaly detection task become more difficult, but it also makes the comparisons become more discriminative. This illustrates that rectifying function of the encoder leads to better performance. In this scenario, F1 score drops from 0.5826 to 0.5551 without the Dictionary Memory, which changes greater than that in entire attack set.

*2) 2nd MEM (Second-stage Memory Module):*

Removing the 2nd MEM makes the model performance drops 0.006 in SWaT with attack 23, 0.0214 in SWaT without attack 23, and 0.0195 in the WADI compared to **MCCED** in terms of F1 score (see **MCCEDw/o2nd-MEM**). Although the model guarantees the rectification of the source sequence through *1st MEM*, high-quality prediction for the pure normal sequence cannot be satisfied. The *2nd MEM* stores the prototypical patterns in normal operational process which is used specially for decoding process. In each step of the decoding process, it can preserve temporal patterns that are ignored by the latent representations, thus helping improve the decoding process. The reduction of the prediction error in the normal sequence further improves the discrimination of the model.

*3) 2nd MEM&Atten (2nd MEM with Attention Module):*

Removing the *2nd MEM* with Attention Module makes model performance drop 0.0086 in SWaT with attack 23, 0.0255 in SWaT without attack 23, and 0.0231 in the WADI compared to **MCCED** in terms of F1 score (see **MCCEDw/o2nd-MEM&Atten**). The drop is more drastic than that of **MCCEDw/o2nd-MEM**, which demonstrates the attention module's efficiency and necessity. Attention module utilizes a dynamic representation of the input sequence, which outperforms pure encoder-decoder model in representation. Compared to the *2nd MEM*, the attention module just provides a temporary and unbounded memory, which only represents the information contained in the current sequence. However, *2nd MEM* supplies a stabilized and bounded memory for accesses at any time. The two module collaboratively helps the prediction decoder predict better and generate more distinguishable prediction error between normal and abnormal data.

Fig. 7 shows the changing curve of prediction value and groundtruth value of specific attacked sensor of different models on detection in attack 3 and attack 20 of SWaT. The model baseline model **CCED** predicts both normal and anomalous sequences too well, resulting in low prediction errors no matter normal sequences or anomalies (see Fig. 7a and 7b). When we add *1st MEM* to baseline model, that is the **MCCEDw/o2nd-MEM&Atten**, we observed that the model predicts anomalous sequences badly and discrimination between the normal and anomalous sequences increases. However, we found that at the same time the normal sequence is also predicted unsteadily, which may result from the sparsity of the further characterization of latent representation in encodings. (see Fig. 7c and 7d). After combining *2nd MEM* and Attention Module, the normal sequences can be fitted more closely and steadily and the abnormal sequences still produce large fitting errors (see Fig. 7e and 7f), which further enlarges the discrimination between the normality and abnormality. It is the result we expect to see the fact that model enhancing the normal memory exclusively (only improve the prediction of normal sequences, instead of both normal and abnormal ones).

*4) Reconstruction Decoder:*

The performance of the model without the reconstruction decoder decreases 0.0237 in the SWaT with attack 23, 0.0252 in the SWaT without attack 23, and 0.0219 in WADI compared to **MCCED** in terms of F1 score. In fact, reconstruction and prediction are similar tasks with the same input and different targets. Reconstruction task forces the encoder to memorize more input information from the previous moment used in prediction task [40]. Therefore, the combination of

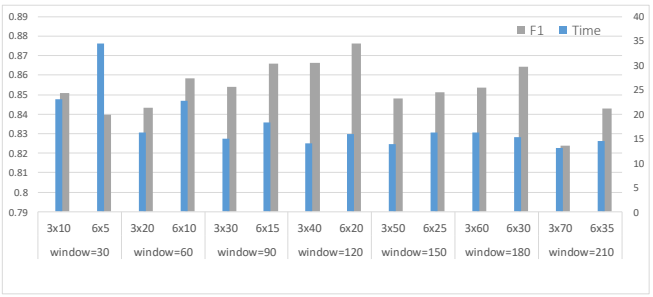



Fig. 8. Effects of Different Window Size, Subsegment size and Subsegment Number.

reconstruction and prediction can improve the performance.

### F. Effects of Fragmenting Hyperparameter

Window (of reconstruction or prediction) size $L$ is an important preprocessing parameter, which determines the throughput of the network and the number of samples in the iteration. We perform a grid search on the window size $L$, subsegment number $T$, subsegment size $l$. For window size, we choose $L$ from $\{30, 60, 90, 120, 150, 180, 210\}$, whose $T$ and $l$ are set respectively according to $L = l \times T$. Here we set $T = 3$ or 6. We conduct various experiments to observe the impacts of three variables settings on F1 measure. The results are shown in Fig. 8, from which We have drawn the conclusions as follows:

1) We can see that the F1 value achieve the best when the window size is 120, specifically $T \times l = 6 \times 20$.
2) Generally, the effect of $T$=6 is better than another that of $T$=3, except for the window length $l = 30$. This may be because relatively larger the number of segments are suitable showing the property of Conv-LSTM module to capture relatively longer temporal dependency.
3) $T = 6$ generally takes more time than $T = 3$ in every epoch. The more time steps demands more sequential operations other than spatial operations (convolution), which is more time-consuming.

## V. Conclusion

In this paper, a Memory-enhanced Composite Conv-LSTM Encoder-Decoder for detecting anomaly in industrial process is proposed. Through performing unsupervised end-to-end learning on the normal training data, the proposed model can concurrently perform reconstruction analysis and prediction analysis on captured process data so that a composite anomaly score can be generated. The proposed model utilizes Conv-LSTM unit to enhance its capability of describing spatiotemporal correlations contained in the normal industrial process. In addition, to deal with the inherent disturbances within the process data, a novel two-stage memory enhancing mechanism is introduced into the proposed model so that the trivial patterns in disturbance data will not be learnt during training. The experimental results on two benchmark datasets proves that the proposed model outperforms the existing baseline and state-of-the-art models, and the designate components within the proposed model indeed promote detection performance of the model.

Our future work is to further improve the detection performance of the proposed model by making it adaptive to concept drifts of the normal industrial process. Concept drift is a phenomenon that constantly appears in process control system due to configuration change, variation of raw material, variation of target product, etc. The occurrence of concept drift makes process patterns deviate from the normal process pattern learnt by baseline model during offline training, hence result in degradation of the detection performance. The concept drift occurred in the testing data of the two benchmark datasets is a major cause of false detections made by our proposed approach. For instance, the processes for recovering the ICS from a succeeded attack is labeled a kind of normal data in testing dataset, but is not included in the training dataset. Moreover, by observing the two benchmark datasets, we found that some of the sensor readings in testing dataset are significantly unstable compared with the normal dataset. A possible research direction is to introduce concept drift detection and incremental learning techniques into the anomaly detection approach.

## Acknowledgment

This work is supported by CCF-NSFOCUS KunPeng Research Fund (2018013), and the research on “Safety Protection technologies for Process Industry based on Adaptive Heterogeneous Neural Network”.

## References

[1] A. Humayed, J. Lin, F. Li, and B. Luo, “Cyber-physical systems security—a survey,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[2] Y. Wang, M. M. Amin, J. Fu, and H. B. Moussa, “A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids,” *IEEE Access*, vol. 5, pp. 26 022–26 033, 2017.

[3] K. Stouffer, J. Falco, and K. Scarfone, “Guide to industrial control systems (ics) security,” *NIST special publication*, vol. 800, no. 82, pp. 16–16, 2011.

[4] K. Zetter, “Inside the cunning, unprecedented hack of ukraine’s power grid,” [Online]. Available: https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/.

[5] T. Ball, “Top 5 critical infrastructure cyber attacks,” [Online]. Available: https://www.cbronline.com/cybersecurity/top-5-infrastructure-hacks.

[6] Y. Mo, R. Chabukswar, and B. Sinopoli, “Detecting integrity attacks on scada systems,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1396–1407, 2013.

[7] Z. Drias, A. Serhrouchni, and O. Vogel, “Analysis of cyber security for industrial control systems,” in *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*. IEEE, 2015, pp. 1–8.

[8] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, “A survey of intrusion detection on industrial control systems,”

*International Journal of Distributed Sensor Networks*, vol. 14, no. 8, p. 1550147718794615, 2018.

[9] O. Bar and L. Lev, "Stealthy deception attacks against scada systems," in *Computer Security: ESORICS 2017 International Workshops, CyberICPS 2017 and SECPRE 2017, Oslo, Norway, September 14-15, 2017, Revised Selected Papers*, vol. 10683. Springer, 2018, p. 93.

[10] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 55, 2014.

[11] P. Wang, M. Govindarasu, A. Ashok, S. Sridhar, and D. McKinnon, "Data-driven anomaly detection for power system generation control," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 1082–1089.

[12] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, "Anomaly detection for a water treatment system using unsupervised machine learning," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 1058–1065.

[13] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *International Conference on Critical Information Infrastructures Security*. Springer, 2016, pp. 88–99.

[14] A. P. Mathur and N. O. Tippenhauer, "Swat: a water treatment testbed for research and training on ics security," in *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. IEEE, 2016, pp. 31–36.

[15] S. Adepu, V. R. Palleti, G. Mishra, and A. Mathur, "Investigation of cyber attacks on a water distribution system," [Online]. Available: http://arxiv.org/abs/1906.02279, 2019.

[16] "Gasoil heating loop dataset," [Online]. Available: https:// kas.pr/ics-research/dataset_ghl_1., 2016.

[17] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, and H. Wang, "Rule-based intrusion detection system for scada networks," 2013.

[18] D. Myers, S. Suriadi, K. Radke, and E. Foo, "Anomaly detection for industrial control systems using process mining," *Computers & Security*, vol. 78, pp. 103–125, 2018.

[19] W. Jiang, Y. Hong, B. Zhou, X. He, and C. Cheng, "A gan-based anomaly detection approach for imbalanced industrial time series," *IEEE Access*, vol. 7, pp. 143 608–143 619, 2019.

[20] W. Yan, L. K. Mestha, and M. Abbaszadeh, "Attack detection for securing cyber physical systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8471–8481, 2019.

[21] L. Martí, N. Sanchez-Pi, J. Molina, and A. Garcia, "Anomaly detection based on sensor data in petroleum industry applications," *Sensors*, vol. 15, no. 2, pp. 2774–2797, 2015.

[22] 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.

[23] L. Tomlin, M. R. Farnam, and S. Pan, "A clustering approach to industrial network intrusion detection," in *Proceedings of the 2016 Information Security Research and Education (INSuRE) Conference (INSuRECon-16)*, 2016.

[24] S. E. Chandy, A. Rasekh, Z. A. Barker, and M. E. Shafiee, "Cyberattack detection using deep generative models with variational inference," *Journal of Water Resources Planning and Management*, vol. 145, no. 2, p. 04018093, 2018.

[25] M. Kravchik and A. Shabtai, "Detecting cyber attacks in industrial control systems using convolutional neural networks," in *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*. ACM, 2018, pp. 72–83.

[26] 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, 2017, pp. 140–145.

[27] P. Filonov, F. Kitashov, and A. Lavrentyev, "Rnn-based early cyber-attack detection for the tennessee eastman process," [Onine]. Available: https://arxiv.org/abs/1709.02232, 2017.

[28] P. Filonov, A. Lavrentyev, and A. Vorontsov, "Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model," [Online]. Available: https://arxiv.org/abs/1612.06676, 2016.

[29] N. Gugulothu, P. Malhotra, L. Vig, and G. Shroff, "Sparse neural networks for anomaly detection in high-dimensional time series."

[30] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 703–716.

[31] D. Li, D. Chen, J. Goh, and S.-k. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," [Online] Available: https://arxiv.org/abs/1809.04758, 2018.

[32] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[33] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1409–1416.

[34] S. Tariq, S. Lee, Y. Shin, M. S. Lee, O. Jung, D. Chung, and S. S. Woo, "Detecting anomalies in space using multivariate convolutional lstm with mixtures of probabilistic pca," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &*

*Data Mining*. ACM, 2019, pp. 2123–2133.
[35] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," [Online]. Available: http://arxiv.org/abs/1904.02639, 2019.
[36] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," 2018.
[37] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014.
[38] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," [Online]. Available: https://arxiv.org/abs/1508.04025, 2015.
[39] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 95–104.
[40] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*, 2015, pp. 843–852.
[41] F. C. et al., "Keras," [Online]. Available: https://github.com/fchollet/keras, 2015.
[42] "Pyod," [Online]. Available: https://github.com/yzhao062/pyod.
[43] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.
[44] D. Shalyga, P. Filonov, and A. Lavrentyev, "Anomaly detection for water treatment system based on neural network with automatic architecture optimization," [Online] Available: https://arxiv.org/abs/1807.07282, 2018.
[45] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in neural information processing systems*, 2007, pp. 153–160.

**Boyang Xia** is currently pursuing B.S. degree with School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. He is a master candidate of the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). His research interests include Anomaly detection, Time series Analysis, and Computer vision.

**Yuxin Zhang** received the B.S. degree in Digital Media Technology from North China University of Technology, Beijing, China, in 2011 and 2015. She is currently a ph.D candidate of the Research Center for Ubiquitous Computing Systems (CUbiCS) at the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). Her research interests include Anomaly Detection, Speech Recognition and Activity Recognition.

**Zhiwen Pan** received the B.S. degree from the Purdue University Calumet, in 2012, and the M.S. and Ph.D. degrees from the University of Arizona, in 2014 and 2017. He is currently an Assistant Research Fellow in Research Center for Ubiquitous Computing System, Institute of Computing Technology, Chinese Academy of Science. His current research focuses on Anomaly Detection, Internet of Things, Industrial Control Systems, and Context Computing.

**Rui Yao** is research assistant in the Pervasive Computing Research Center at the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). She received the B.S. degrees in Electronic Information Engineering from Dalian University of Technology, Dalian, China, in June 2019. Her research interests include Artificial Intelligence, Anomaly detection, and Computer Vision.

**Yiqiang Chen** received the B.S. and M.S. degrees in computer science from Xiangtan University, Xiangtan, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003. In 2004, he was a Visiting Scholar Researcher with the Department of Computer Science, Hong Kong University of Science and Technology (HKUST), Hong Kong. He is currently a professor and the director of the Pervasive Computing Research Center at the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). His research interests include Artificial Intelligence, Pervasive Computing, and Human-Computer Interaction.

**Yuting He** is now studying in Chongqing University for Bachelor's degree. She is a master candidate of the research Center for Ubiquitous Computing Systems (CUbiCS) at the institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). Her research interests include Artificial Intelligence, Anomaly Detection, and Sign Language Recognition.