State-aware anomaly detection for massive sensor data in Internet of Things

Jiafan He*
jiafanhe19@ucla.edu
Dept. of Computer Science, UCLA

Lu-An Tang
ltang@nec-labs.com
NEC-Labs America

Peng Yuan
pyuan@nec-labs.com
NEC-Labs America

Yuncong Chen
yuncong@nec-labs.com
NEC-Labs America

Haifeng Chen
haifeng@nec-labs.com
NEC-Labs America

Yuji Kobayashi
y_koba@nec.com
NEC Central Research Lab

Quanquan Gu
qgu@ucla.edu
Dept. of Computer Science, UCLA

ABSTRACT

With the escalating prevalence of Internet of Things (IoTs) in critical infrastructure, the requirement for efficient and effective anomaly detection solution becomes increasingly important. Unfortunately, most prior research works have largely overlooked to adapt detection criteria for different operational states, thereby rendering them inadequate when confronted with diverse and complex work states of IoTs. In this study, we address the challenges of IoT anomaly detection across various work states by introducing a novel model called Hybrid State Encoder-Decoder (HSED). HSED employs a two-step approach, beginning with identification and construction of a hybrid state for Key Performance Indicator (KPI) sensors based on their state attributes, followed by the detection of abnormal or failure events utilizing high-dimensional sensor data. Through the evaluation on real-world datasets, we demonstrate the superiority of HSED over state-of-the-art anomaly detection models. HSED can significantly enhance the efficiency, adaptability and reliability of IoTs and avoid potential risks of economic losses by IoT failures.

KEYWORDS

Anomaly detection, Internet of Things, State-aware monitoring

ACM Reference Format:
Jiafan He, Lu-An Tang, Peng Yuan, Yuncong Chen, Haifeng Chen, Yuji Kobayashi, and Quanquan Gu. 2023. State-aware anomaly detection for massive sensor data in Internet of Things. In *Proceedings of The 3rd Workshop on Artificial Intelligence-Enabled Cybersecurity Analytics (AI4Cyber, August 7, 2023, Long Beach, California).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

AI4Cyber, August 7, 2023, Long Beach, California, XXXX

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

With the growth of cyber technologies and communication networks, Internet of Things (IoTs) have witnessed a remarkable surge in critical infrastructure domains, such as transportation networks, energy systems, water and gas distribution networks, as well as unmanned factories. The salient feature of IoTs lies in the ability to establish a tight connection between computational and physical components. By seamlessly integrating physical and cyber elements, IoTs can execute intricate real-time tasks by acquiring data from sensors, conducting data analysis, and initiating actions based on the obtained insights. Nevertheless, as the scale and complexity of IoTs continue to expand at a rapid pace, the consequences of system failures or security breaches can be severe, even posing life-threatening risks.

Therefore, it is necessary to develop data-centric anomaly detection solutions to automatically monitoring the operation of IoTs. The solution should be able to analyze large volumes of data in real-time, facilitate early detections of anomalies and prompt remedial measures. A major challenge to implement such a solution is on building the system profile for different work states. Most IoTs have multiple work states (i.e., operational modes). The normal dynamics exhibited by Key Performance Indicator (KPI) sensors may vary across different work states. Unfortunately, most existing anomaly detection methods cannot adapt their models in accordance with the work state. Consequently, they may report lots of false alerts when IoTs change the work states in a normal manner, or miss the real anomalies while testing with a wrong profile model.

Furthermore, the anomaly detection across multiple work states presents a series of challenges: First, the dynamics of work states are intricately linked to system operations or transitions. Due to a lack of domain expertise, most users cannot provide accurate and detailed information of the work states. Instead, they would like the monitoring system to learn the work states and provide them as output. Second, the data acquired from IoT sensors typically exhibits high dimensionality and is prone to noise. It is difficult to direct extract meaningful information from such data without a clear indicator. Last, the system behavior and associated dynamics can vary significantly among different states. Such distances among normal states may be magnitude larger than the difference of normal/abnormal behaviors.

In this study, we address the above challenges of IoT anomaly detection and propose a novel framework called Hybrid State Encoder-Decoder (HSED). The major contributions are listed as follows:

- HSED employs state encoders model to recover multiple hidden states for the IoT. The system extracts embedding features from individual state and constructs a hybrid model to address dynamic work states.
- HSED reconstructs the sensor signals based on the hybrid hidden state of IoT. The system then conducts anomaly detection by comparing the reconstructed signals with original ones.
- HSED is deployed and evaluated on real world dataset. HSED achieves higher accuracy in anomaly detection tasks comparing to other state-of-the-art methods.

2 RELATED WORK

In recent years, there has been a notable surge of interest on applying deep learning-based approaches for anomaly detection in IoT data [8]. Most methods focus on anomaly detection by reconstructing the sensor data with deep models. Recurrent Neural Networks (RNNs) have gained widespread adoption for capturing temporal dependencies in IoTs [4, 13], and auto-encoders are commonly employed to model correlations among diverse sensors [1, 2].

Recurrent Variational Auto-Encoder (VAE) architectures have been used to construct non-linear state space models [6] and facilitate the learning of disentangled representations [3]. RNN-based deep state space models have been proposed Rangapuram et al. [10], Salinas et al. [11] to accomplish probabilistic sensor data forecasting. However, the majority of these approaches are limited to constructing a single model and are not well-suited for addressing the diverse work states in IoT applications. Moreover, these methods require the users to input prior domain-specific knowledge. Their applicability is also restricted and confined to some specific domains, such as power management.

One of the seminal works in contextual anomaly detection is Conditional Anomaly Detection (CAD) [12]. This method requires users to partition features into contextual and behavioral categories. The method uses Gaussian Mixture Models (GMM) to fit the distributions of the contextual and behavioral feature spaces and learns dependencies between them. Another method, the Robust Contextual Outlier Detection (ROCOD) method [7], uses both local and global models to describe the relationships between contextual and behavioral features. The main problem of above methods is on the assumption. They rely on the assumption that features adhere to specific statistical distributions, such as GMM. However, this assumption may prove to be too restrictive in real-world scenarios. It can potentially hinder the accuracy of the results obtained from these methods.

3 PROBLEM DEFINITION

Let $\mathcal{X} = \{\mathbf{X}_1, ..., \mathbf{X}_n\}$ be the historical data-set, which consists of n different sensor data. Each sensor data $\mathbf{X}_t \in \mathbb{R}^{d \times T}$ is a multivariate time series with length T and for each time step $1 \le t \le T$, $\mathbf{X}_{i,t} = [\mathbf{C}_{i,t}, \mathbf{S}_{i,t}] \in \mathbb{R}^{d_1+d_2}$ is the corresponding multi-variate vector with dimension $d = d_1 + d_2$. Here, $\mathbf{S}_{i,t} \in \mathbb{R}^{d_2}$ is the data collected from KPI sensors.

Note that, the value of KPI sensors can vary significantly across different work states. To incorporate these information, we introduce contextual variables $\mathbf{C}_{i,t} \in \mathbb{R}^{d_1}$ to denote the state attributes, which indicates the dynamic of the work state.

In general, anomaly detection can be divided into two categories: point anomalies and contextual anomalies. Point anomalies refer to data points that deviate significantly from the expected behavior observed throughout the entire trajectory. On the other hand, contextual anomalies pertain to data points that exhibit significant deviations from the expected behavior within a particular context or environment. The identification of contextual anomalies necessitates considering the specific environment or context in which the data point is observed. Contextual anomalies are more difficult to detect than point anomalies, and we focus on the detection of contextual anomalies in this study.

Problem Definition (Anomaly Detection): Let $\{\mathbf{S}_1, ..., \mathbf{S}_n\}$ be the historical dataset of normal operation, and $\{\mathbf{C}_1, ..., \mathbf{C}_n\}$ be the state information, the task is to learn a model $\mathcal{M}$, such that when new streaming data $\mathbf{X}$ comes in, the model $\mathcal{M}$ can detect whether $\mathbf{X}$ is abnormal or not.

Algorithm 1 HSED

Require: Number of different states m

1: Extracting the feature embedding by
$\text{Enc}(\cdot), \{\text{Center}_j\}_{j=1}^{m} \leftarrow$ **Cond-Embedding**(C)
2: Set the similarity matrix by
$S(\mathbf{C}_i, \text{Center}_j) = \|\text{Enc}(\mathbf{C}_i) - \text{Center}_j\|_2$
3: Set the hybrid hidden-states as
$\mathbf{P}_{i,j} \sim \left(1 + S(\mathbf{C}_i, \text{Center}_j)/\alpha\right)^{-(1+\alpha)/2}$
4: **for** each main state $\mathbf{C}_j$ and KPI attributes $\mathbf{S}_i$ **do**
5: Extracting the embedding for KPI attributes $\mathbf{S}_i$ within state $\mathbf{C}_j$
$\text{Embedding}_j(\cdot) \leftarrow$ **Sensor-Embedding**$(\mathbf{S}_i, \mathbf{P}_{i,j})$
6: **end for**
7: For each sequence $\mathbf{S}_i$, combining the embedding and recovering the original sequence
$\mathbf{S}'_i, \mathbf{S}'_{i,b} \leftarrow$ **Rec**$\left(\{\text{Embedding}_j(\mathbf{S}_i), \mathbf{P}_{i,j}\}_{j=1}^{m}\right)$

4 THE HSED FRAMEWORK

In this section, we propose a noval framework called Hybrid State Encoder-Decoder (HSED) to detect the anomalies in sensor data of IoTs. The overall system structure is shown in Figure 1. And the main steps are illustrated in Algorithm 1.

4.1 Distinguish Hidden States

Since the dynamic of KPI sensor $\mathbf{S}$ highly relied on the hidden state, it becomes imperative to first uncover the hidden state before proceeding with anomaly detection. Hence, in HSED algorithm (Line 1), we employ an auto-encoder-based deep temporal clustering mechanism **Cond-Embedding** [5] to distinguish the hidden state from the state attributes $C = \{\mathbf{C}_1, .., \mathbf{C}_n\}$, where each clustering Center_j corresponding to a different state. The whole process is

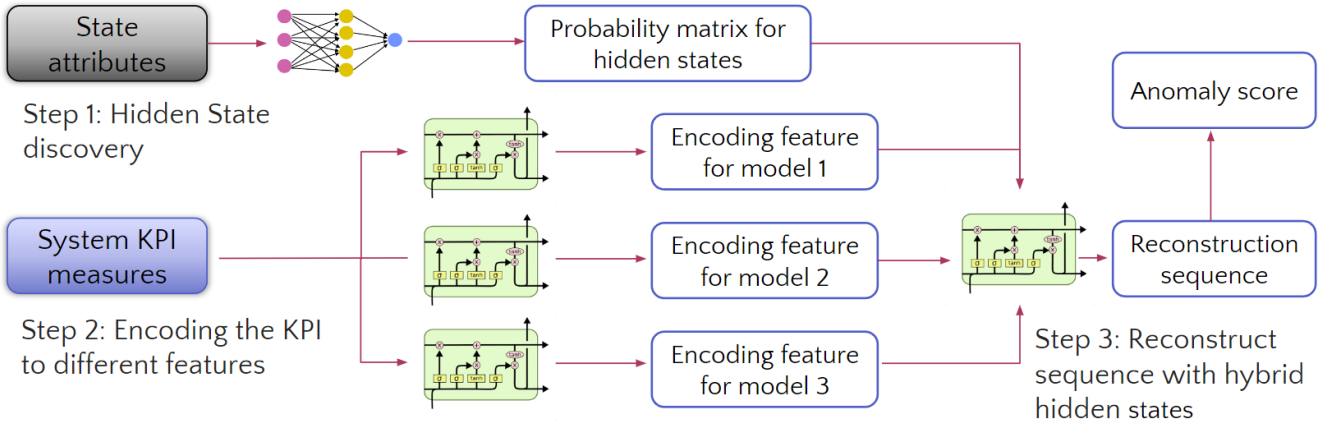


Figure 1: The overall framework for the HSED and details for each steps can be found in Section 4.1 to 4.5.

shown as Algorithm 2: Given the input sequence $\mathbf{C}_i \in \mathbb{R}^{d_1 \times T}$, we use sequential GRU-cells as encoder to extract the time-series information from both the forward sequence $\mathbf{C}_i$ and the corresponding inverse sequence $\mathbf{C}_{i,b}$. After receiving the feature embedding matrices $\mathbf{H}_f, \mathbf{H}_b \in \mathbb{R}^{T \times l}$, we introduce the attention mechanism (Lines 4 to 6) to aggregate the feature across different timestamps. Based on the final embedding feature $\text{Enc}(\mathbf{C}_i)$, we reconstruct the original sequence with a GRU-based decoder (Line 7) and also perform K-Means algorithm with the embedding feature to generate different clusters (Line 8). Based on the reconstruction error and clustering error, the training loss of algorithm 2 consists of the following three parts:

$$\text{Loss} = \alpha \cdot \underbrace{\sum_{i=1}^{n} \|\mathbf{C}_i - \mathbf{C}'_i\|_2^2}_{\text{Auto-encoder forward loss}} + \alpha \cdot \underbrace{\sum_{i=1}^{n} \|\mathbf{C}_{i,b} - \mathbf{C}'_{i,b}\|_2^2}_{\text{Auto-encoder backward loss}} + \beta \cdot \underbrace{\sum_{i=1}^{n} \|\text{Enc}(\mathbf{C}_i) - \text{Center}(\text{Enc}(\mathbf{C}_i))\|_2^2}_{\text{K-mean loss}},$$

where $\text{Center}(\text{Enc}(\mathbf{C}_i))$ correspond to the closest center for the feature embedding $\text{Enc}(\mathbf{C}_i)$. In Algorithm 2, we aim to minimize the training loss and update the parameters $\boldsymbol{\theta}_f, \boldsymbol{\theta}_b$ with gradient descent.

4.2 Hybrid State Construction

Now HSED has distinguished the hidden states Center_j. However, time series $\mathbf{X}_i$ may not fall into each state, the smoothly dynamic of the hidden state and the existence of some intermediate state makes it more difficulty to detect the abnormal events. To capture the smoothly switching of hidden states and corresponding intermediate states, we develop a similarity-based mechanism to discover the hybrid states and decomposes each time series $\mathbf{X}_i$ to the hybrid of multiple basic hidden states. More specifically, for each time series $\mathbf{X}_i$, we focus on the state attributes $\mathbf{C}_i$ and first extract the feature embedding $\text{Enc}(\mathbf{C}_i)$ from Algorithm 2. Then each state attributes $\mathbf{C}_i$ and the hidden state Center_j, we compute the similarity,

Algorithm 2 Cond-Embedding

Require: Number of different states m, Forward and backward parameters $\boldsymbol{\theta}_f, \boldsymbol{\theta}_b$

1: **Input**: state attributes $\mathbf{C}_i \in \mathbb{R}^{d_1 \times T} = [\mathbf{x}_1, .., \mathbf{x}_T]$
2: Create the inverse sequence $\mathbf{C}_{i,b} = [\mathbf{x}_T, .., \mathbf{x}_1]$
3: Let $\mathbf{H}_f \leftarrow \mathbf{GRU}_{\boldsymbol{\theta}_f}(\mathbf{C}_i) = [\mathbf{h}_1, .., \mathbf{h}_T]$, $\mathbf{H}_b \leftarrow \mathbf{GRU}_{\boldsymbol{\theta}_b}(\mathbf{C}_{i,b}) = [\mathbf{h}_{1,b}, .., \mathbf{h}_{T,b}]$
4: $\mathbf{u}_f = \tanh(H_f \cdot \mathbf{W}_f + \boldsymbol{b}_f)$, $\mathbf{u}_b = \tanh(H_b \cdot \mathbf{W}_b + \boldsymbol{b}_b)$
5: $\mathbf{w}_{f,t} = \text{Soft-Max}(\mathbf{u}_f \circ \mathbf{h}_t)$, $\mathbf{w}_{b,t} = \text{Soft-Max}(\mathbf{u}_b \circ \mathbf{h}_{t,b})$
6: $\mathbf{h}_f^{\textbf{attention}} = \sum_{t=1}^{T} \mathbf{w}_{f,t}\mathbf{h}_t$, $\mathbf{h}_b^{\textbf{attention}} = \sum_{t=1}^{T} \mathbf{w}_{b,t}\mathbf{h}_{t,b}$
7: Set embedding feature $\text{Enc}(\mathbf{C}_i) = \mathbf{h}_f^{\textbf{attention}} + \mathbf{h}_b^{\textbf{attention}}$
8: Reconstruct the sequences $\mathbf{C}'_i \leftarrow \mathbf{GRU}_{\boldsymbol{\theta}_f}(\text{Enc}(\mathbf{C}_i))$, $\mathbf{C}'_{i,b} \leftarrow \mathbf{GRU}_{\boldsymbol{\theta}_b}(\text{Enc}(\mathbf{C}_i))$
9: Perform K-mean algorithm for the embedding feature $\text{Enc}(\mathbf{C}_i)$ and return the centers: $\{\text{Center}_j\}_{j=1}^{m}$

which is denote by the euclidean distance between $\text{Enc}(\mathbf{C}_i)$ and Center_j: $S(\mathbf{C}_i, \text{Center}_j) = \|\text{Enc}(\mathbf{C}_i) - \text{Center}_j\|_2$ (Line 2). At last, we construct the hybrid state with the Student's t distribution and generate the weight matrix $\mathbf{P}_{i,j} \sim \left(1 + S(\mathbf{C}_i, \text{Center}_j)/\alpha\right)^{-(1+\alpha)/2}$ (Line 3), where $w_{i,j}$ represents the probability to assign state series $\mathbf{C}_i$ to main state Center_j. With the help of this weighted matrix $\mathbf{P}$, the hidden state for series Xi can be decomposed as a hybrid of main states.

4.3 Extract Embedding Features

After constructing the hybrid states, HSED employs a LSTM-based encoder to extract the features from the KPI sensors $\boldsymbol{S}_i$. Since the sensor data of different states can be totally different, HSED needs to collect the features of various information such as period or magnitude across multiple states. Therefore, HSED constructs individual encoders and decoders for each main state Center_j separately and the detailed algorithm is shown in Algorithm 3. Specifically, the encoder consists two different LSTM-based units to process the KPI sensor data: the first one is used to aggregate the time-series

Algorithm 3 Sensor-Embedding

Require: Number of different states m, Forward and backward parameters $\boldsymbol{\theta}_{f,j}, \boldsymbol{\theta}_{b,j}$

1: **Input**: KPI attributes $\boldsymbol{S}_i \in \mathbb{R}^{d_2 \times T} = [\mathbf{y}_1, .., \mathbf{y}_T]$, Hybrid-state matrix $\mathbf{P}_{i,j}$
2: Create the inverse sequence $\boldsymbol{S}_{i,b} = [\mathbf{y}_T, .., \mathbf{y}_1]$
3: Let $\mathbf{h}_f \leftarrow \mathbf{LSTM}_{\boldsymbol{\theta}_{f,j}}(\boldsymbol{S}_i)$, $\mathbf{h}_b \leftarrow \mathbf{LSTM}_{\boldsymbol{\theta}_{b,j}}(\boldsymbol{S}_{i,b})$
4: Set embedding feature $\text{Embedding}_j(\boldsymbol{S}_i) = \text{Concatenate}(\mathbf{h}_f, \mathbf{h}_b)$
5: Reconstruct the sequences $\boldsymbol{S}'_i \leftarrow \mathbf{LSTM}_{\boldsymbol{\theta}_{f,j}}(\text{Embedding}_j(\boldsymbol{S}_i))$, $\boldsymbol{S}'_{i,b} \leftarrow \mathbf{LSTM}_{\boldsymbol{\theta}_{b,j}}(\text{Embedding}_j(\boldsymbol{S}))$

Algorithm 4 Rec

Require: Number of different states m, Forward and backward parameters $\boldsymbol{\theta}_f, \boldsymbol{\theta}_b$

1: **Input**: Embedding feature $\{\text{Embedding}_j(\boldsymbol{S}_i)\}_{j=1}^m$, Hybrid-state matrix $\mathbf{P}_{i,j}$
2: Set embedding feature $\text{Embedding}(\boldsymbol{S}_i) = \text{Concatenate}(\mathbf{P}_{i,j} \cdot \text{Embedding}_j(\boldsymbol{S}_i))$
3: Reconstruct the sequences::
$\boldsymbol{S}'_i \leftarrow \mathbf{LSTM}_{\boldsymbol{\theta}_f}(\text{Embedding}(\boldsymbol{S}_i))$,
$\boldsymbol{S}'_{i,b} \leftarrow \mathbf{LSTM}_{\boldsymbol{\theta}_b}(\text{Embedding}(\boldsymbol{S}))$

information from the forward sequence and the other focused on the backward sequence (Line 3). After concatenating the output of two different units $\mathbf{h}_f, \mathbf{h}_b$ (Line 4), HSED obtains the encoding feature $\text{Embedding}_j(\boldsymbol{S}_i)$ and feed it into the decoder to reconstruct the original sequence (Line 5). Finally, since each time series $\boldsymbol{S}_i$ is not purely belong to one main state Center_j, the reconstruction error for each main state Center_j is also weighted by the hybrid-state matrix $\mathbf{P}_{i,j}$ as following:

$$\text{Loss}_j = \underbrace{\sum_{i=1}^{n} \mathbf{P}_{i,j} \|\boldsymbol{S}_i - \boldsymbol{S}'_i\|_2^2}_{\text{Hybrid Auto-encoder forward loss}} + \underbrace{\sum_{i=1}^{n} \mathbf{P}_{i,j} \|\boldsymbol{S}i, b - \boldsymbol{S}'_{i,b}\|_2^2}_{\text{Hybrid Auto-encoder backward loss}}, \tag{4.1}$$

and HSED uses gradient descent method to update the parameters $\boldsymbol{\theta}_{f,j}, \boldsymbol{\theta}_{b,j}$ for main state Center_j and minimize the loss for each state.

4.4 Reconstruction with Hybrid State

So far, HSED retrieves the encoding features $\text{Embedding}_j(\boldsymbol{S}_i)$ for each main state and KPI sensor $\boldsymbol{S}_i$. For each KPI sensor $\boldsymbol{S}_i$, we concatenate the feature $\text{Embedding}_j(\boldsymbol{S}_i)$ with the hybrid-state matrix and feed it in to a LSTM-based decoder to reconstruct the original sequence (See Algorithm 4). In this step, we update the parameter with gradient descent to minimize the following reconstruction loss:

$$\text{Loss} = \underbrace{\sum_{i=1}^{n} \mathbf{P}_{i,j} \|\boldsymbol{S}_i - \boldsymbol{S}'_i\|_2^2}_{\text{Forward Reconstruction Error}} + \underbrace{\sum_{i=1}^{n} \mathbf{P}_{i,j} \|\boldsymbol{S}_{i,b} - \boldsymbol{S}'_{i,b}\|_2^2}_{\text{Backward Reconstruction Error}}.$$

4.5 Anomaly Detection

After training the model by four steps, HSED combines the state-aware models and detects the anomalies for online sequence $\mathbf{X} = [\mathbf{C}, \boldsymbol{S}]$. Similar to the training process, the anomaly detection also consists of four steps. In the first step, HSED implies the deep temporal clustering method (Algorithm 2) with the state series $\mathbf{C}$ and generate the state-embedding vector $\text{Enc}(\mathbf{C})$. Next, HSED uses the similarity matrix to compute the euclidean distance between $\text{Enc}(\mathbf{C})$ and each main stateCenter_j, and discover the hybrid hidden states with Student's t distribution (Line 3 in Algorithm 1). In the third step, HSED encodes the KPI sensor $\boldsymbol{S}$ for each main state (Algorithm 2) and generates embedding features $\{\text{Embedding}_j(\boldsymbol{S})\}_{j=1}^m$. Finally, HSED reconstructs the original KPI sensor data $\boldsymbol{S}'$ and $\boldsymbol{S}'_b$ with hybrid hidden models (Algorithm 4). Based on the reconstruction error between the original series $\boldsymbol{S}$ and reconstructed data $\boldsymbol{S}', \boldsymbol{S}'_b$, HSED computes the reconstruction error as the anomaly score and report an alert if the anomaly score is higher than a threshold. In addition, HSED can also compute a confidence score based on the distribution of the hybrid hidden states.

5 PERFOMANCE EVALUATION

5.1 Experiment Setting

Engine dataset: This dataset contains 17 sensors collected from an IoT on vehicle engine and includes over 10000 timestamps. There are multiple work states of the eninge, and we focus on detect the following anomalies related to engine failure:

- High temperature: the engine temperature is too high.
- Misfiring: One or more engine cylinders are misfiring.
- Acceleration problem: Engine load has fluctuations or irregularities.

Compared method: We compare the HSED with the EncDec-AD method [9], which directly reconstruct the original time-series behavior with LSTM-Based auto-encoder.

For the engine dataset, we first use max-min operation to normalize each data and then divide the data-set to sliced window with length 150. After using the HSED algorithm, the average sqaure-reconstructed error is 0.1348. In comparison, the reconstructed error of EncDec-AD is 0.1532, which has a worse performance than our novel method.

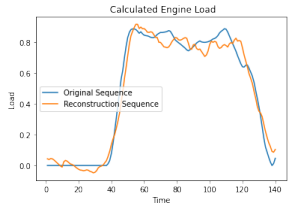
6 CONCLUSION

In this work, we proposed a new algorithm, called HSED, for IoT anomaly detection across various work state. HSED first introduce a state encoders to discover the hybrid model for the dynamic work state and later, employs a LSTM-based auto-encoder for reconstructing the original sensor signal. The evaluation on real-world datasets also suggests the superiority of HSED over state-of-the-art anomaly detection models.

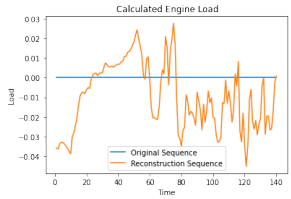
REFERENCES

[1] Abdulrahman Al-Abassi, Jacob Sakhnini, and Hadis Karimipour. 2020. Unsupervised Stacked Autoencoders for Anomaly Detection on Smart Cyber-physical Grids. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 3123–3129.

[2] Mikel Canizo, Isaac Triguero, Angel Conde, and Enrique Onieva. 2019. Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study. *Neurocomputing* 363 (2019), 246–260.



(a)



(b)

Figure 2: This Figure depicts the reconstruction sequence for the attribute "Calculated Engine Load". For the second KPI sensor, there exist a large deviation between the original sequence and the reconstructed sequence. Under this situation, it may be caused from acceleration problems and will report anomaly events to the agent.

[3] Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. 2017. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *Advances in neural information processing systems* 30 (2017).

[4] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 387–395.

[5] Dino Ienco and Roberto Interdonato. 2020. Deep multivariate time series embedding clustering via attentive-gated autoencoder. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 318–329.

[6] Rahul Krishnan, Uri Shalit, and David Sontag. 2017. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[7] Jiongqian Liang and Srinivasan Parthasarathy. 2016. Robust contextual outlier detection: Where context meets sparsity. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. 2167–2172.

[8] Yuan Luo, Ya Xiao, Long Cheng, Guojun Peng, and Danfeng Yao. 2021. Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–36.

[9] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148* (2016).

[10] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. 2018. Deep state space models for time series forecasting. *Advances in neural information processing systems* 31 (2018).

[11] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.

[12] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. 2007. Conditional anomaly detection. *IEEE Transactions on knowledge and Data Engineering* 19, 5 (2007), 631–645.

[13] Shahroz Tariq, Sangyup Lee, Youjin Shin, Myeong Shin Lee, Okchul Jung, Daewon Chung, and Simon S Woo. 2019. 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*. 2123–2133.