

# Q residual non-parametric Distribution on Fault Detection Approach Using Unsupervised LSTM-KDE

Nur Maisarah Mohd Sobran<sup>1</sup>, Zool Hilmi Ismail<sup>2</sup>

<sup>1,2</sup> *Centre for Artificial Intelligence & Robotics, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia*

*nurmaisarah@utem.edu.my  
zool@utm.my*

<sup>1,2</sup> *Malaysia–Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia*

<sup>1</sup> *Fakulti Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka, Jalan Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.*

## ABSTRACT

It is well known among practitioner, majority collected data from industrial process plant are unlabeled. The collected historical data if utilize, able to provide vital information of process plant condition. Learning from unlabeled dataset, this study proposed Unsupervised LSTM-KDE approach as a measure to predict fault in industrial process plant. The residual based fault detection approach framework is utilized with long short-term memory (LSTM) as the main pattern learner for nonlinear and multimode condition that usually appear in process plant. Furthermore, kernel density approach (KDE) is used to determine the threshold value in non-parametric condition of unlabeled data. The LSTM-KDE approach later is evaluated with numerical data as well as Tennessee Eastman process plant dataset. The performance also was compared to Principal Component Analysis (PCA), Local outlier factor (LOF) and Auto-associative Kernel Regression (AAKR) to further examine the LSTM-KDE performance. The experimental results indicate that the LSTM-KDE fault detection approach has better learning performance and accuracy compared to other approaches.

## 1. INTRODUCTION

Process monitoring has become a crucial stage in the manufacturing industry. One of the motivations is to ensure personnel safety and optimum output by monitoring the entire engineering system. The current manufacturing plant, equipped with up-to-date data acquisition tools such as the

Internet of Things (IoT), supervisory control and data acquisition (SCADA), cloud computing, and networks, collects an abundance of informative datasets. Hence, massive, collected signals are available in order to provide vital indications of the current plant condition. Consequently, draw more attention towards data-driven fault detection in monitoring industrial processes. With this, high-volume and multivariate datasets are inevitable, given the multiple components and sensory network in the manufacturing system. Multivariate statistical process monitoring (MSPM) approaches such as principal component analysis (PCA) and partial least squares (PLS) were seen as prominent approaches to cater to the multivariate condition, but the limitation relies on linear and gaussian assumptions, which are unfit for today's industrial system (K. Wang, Zhou, and Wang 2020). Since the collected signal relies heavily on time series conditions, adding in the temporal relation would increase the reliability of the fault detection approach (Ammann, Michau, and Fink 2020). In addition, time series analysis will also enable the future prediction of faulty cases in process monitoring rather than present-day time analysis. This extends condition monitoring to preventive measures. Other than that, high numbers of pieces of evidence presented by researchers focus on supervised and semi-supervised approaches in fault detection cases (Alrifaey, Lim, and Ang 2021; Belagoune et al. 2021; Dong, Ma, and Liu 2019; Jin et al. 2020; Li et al. 2020; Monostori et al. 2016; Shadi, Ameli, and Azad 2022; Wang et al. 2018). However, less study was found for the unsupervised approach, despite the majority of industrial monitoring data being unlabeled. Therefore, a non-parametric time-series unsupervised fault detection approach will be beneficial considering the current industrial process plant environment.

Nur Maisarah Mohd Sobran et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

<https://doi.org/10.36001/IJPHM.2024.v15i2.3941>

Practically, the observation signal in process monitoring usually comes with a time series, an unknown distribution model, and parameter values. At the same time, determination threshold parameters such as the upper control limit,  $u_{CL}$  and the lower control limit,  $l_{CL}$  are important to distinguish the nominal signal condition from the faulty occurrences. Therefore, two (2) main factors that affect the efficiency of the non-parametric time series fault detection approach will be the time series model, which is based upon the choice of model,  $f$  and the unknown data distribution, which heavily impacts the determination of the threshold value.

Data-driven fault detection model,  $f$  selection has been divided into three parts: MSPM, shallow artificial intelligence (shallow AI), and deep learning approaches. The limitation on MSPM was already discussed at the first paragraph. The shallow AI approach, which is usually more towards application-specific problem-solving and hand-crafted feature extraction, shows satisfactory implementation in (Nagarajan, Kayalvizhi, and Karthikeyan 2016; Ren et al. 2018; Shao et al. 2023; Tang et al. 2018). However, considering the massive dataset, nonlinear, and temporal relations that come together in engineering systems, deep learning approaches that provide more abstraction learning are found suitable as fault detection models. Recently, among many deep learning approaches, the authors found three (3) main architectures that were selected as unsupervised approaches to fault detection: deep belief network (DBN), autoencoder (AE), and long short-term memory (LSTM).

The DBN architecture consists of multiple restricted Boltzmann machine (RBM) models. In fault detection, DBN was used as model,  $f$  in (Ren et al. 2018) and (Anaissi and Zandavi 2019) due to its capability to capture nonlinear patterns in process plants. Implicitly, DBN was seen as a good feature extractor considering the complex signal analysis in engineering systems. However, less discussion was seen on temporal relations in both studies, but similar results of unsupervised AE fault detection were seen in (K. Wang et al. 2020) and (Xiang et al. 2020). (K. Wang et al. 2020) argued that the accuracy of AE fault detection is affected by noise in the dataset, and the selection of the activation function influences the detection margin. After that, the study proposed a deviation degree penalty to overcome this. Improvements in terms of fault alarm rate were seen, but this result does not reflect the time series analysis that is needed in the fault detection monitoring system. On the other hand, (Chen et al. 2020) innovated a new AE architecture with a Gated Recurrent Unit (GRU) neural network. The notion is to combine both feature extraction and correlation analysis in AE with a time series GRU model in order to improve the reconstruction loss in the normal state so that there is an obvious difference between a normal and faulty signal. However, the AE-GRU output distribution was assumed to be normal. Likewise (Amarbayasgalan et al. 2020) in, which chooses multivariate

normal distribution for prior data distribution in training the sliding window variational autoencoder fault detection approach. Another perspective was in (Calabrese et al. 2020), which explained that the optimal subsequent time series window was generated by using autoregressive (AR), model learning with an autoencoder, and determining the threshold value using density-based spatial clustering of applications with noise (DBSCAN). However, using the clustering approach will require a detailed analysis of whether the grouped boundary is considered 'normal' or 'faulty'. Experienced personnel with the data domain are needed in this stage (Jiang et al. 2018). Furthermore, a study in (Jiang et al. 2018) proposed another fault detection framework based on sliding window AE and Kernel Density Estimation (KDE) as methods for determining threshold values.

Focusing on temporal relations or time series analysis, more attention is given to the recurrent neural network (RNN) family. Recently, LSTM, Bidirectional LSTM (BiLSTM), and GRU were among the selected architectures for the unsupervised fault detection approach (UFD). (Machado et al. 2022) proposed LSTM-AE for detecting multiple types of faulty signals at oil wells, whereas (J. Zhao et al. 2020) used BiLSTM with the same intention but in slow and fast dynamic system conditions. The same goes in (Kukkala, Thiruloga, and Pasricha 2020) using GRU to detect multiple intrusion attacks that are suitable for embedded systems. But the threshold determination for (Machado et al. 2022), (J. Zhao et al. 2020) and (Kukkala et al. 2020) is under gaussian assumption with the F1 score, Gaussian segment model, and maximum reconstruction score, respectively. Other than that, (Yu and Yan 2021) modified the LSTM architecture to capture better dynamicity in complex systems and KDE for fault detection. Furthermore, (Yu, Liu, and Ye 2021) studied lower-dimensional features representative of nonlinear complex systems using convolutional LSTM autoencoders (CLSTM-AEs) and KDE. In the perspective of early fault detection, (B. Zhao et al. 2020) solved the weak nonlinear system problem using CNN-LSTM, whereas the decision boundary nonlinear output frequency response functions (NOFRFs) indicator is specific to the application of the study area. On the other hand, (Liu et al. 2020) proposed multilayer LSTM and isolation forest (iForest) for detecting early warnings. However, the study did not mention how the selection of the iForest threshold value was made. Furthermore, (B. Wang, Liu, et al. 2020) developed a sliding window LSTM with a low-pass infinite impulse response (IIR) filter approach for large noise results, smoothing residuals for threshold determination with mean and standard deviation. The same concern was raised in (Ellefsen et al. 2019) where uncertainty in maritime components resulted in a noise dataset, and LSTM was proposed as one of the methods for learning this type of dataset. However, in the study, the threshold determination is specifically based on the velocity and acceleration of the dataset. The real-time efficiency of LSTM was proven in (B. Wang, Peng, et al.

2020) and the scalability issue with small applications was studied in (Silva et al. 2022). Other than that, the GRU-gaussian mixture VAE approach was developed to cover multimodal systems (Guo et al. 2020). In addition, LSTM was also used in the vibration analysis of a motor, but the threshold determination is specific to the application itself (Principi et al. 2019).

From the literature, it is found that the LSTM family gives satisfactory performance as an UFD approach but less focus on the threshold determination. The study in (Guo et al. 2020, 2020; Kukkala et al. 2020, 2020; Yu and Yan 2021; J. Zhao et al. 2020) decided the decision boundary for ‘normal’ and ‘faulty’ under gaussian assumption, which in practice, under an unsupervised approach, no distribution conditions were known, whereas (Kourti and MacGregor 1995; B. Wang, Liu, et al. 2020; B. Wang, Peng, et al. 2020; B. Zhao et al. 2020) proposed application-specific threshold determination. It is also found that studies in (B. Zhao et al. 2020), and (Yu et al. 2021) use a non-parametric kernel density estimation statistical approach, but evaluation not cover different modes dataset. Thus, this study will contribute to this area of study, which includes:

1. Unsupervised non-parametric time series fault detection framework for a nonparametric distribution dataset in a complex engineering system using the LSTM model.
2. Threshold determination via Kernel Density Estimation for non-parametric residual distribution in complex engineering systems.
3. Comparison of several models with the proposed unsupervised fault estimation framework.

## 2. PRELIMINARIES

### 2.1. Statistical Process Control Chart Fault Detection

In a multivariable process control system,  $n \times p$  data matrix,  $\mathbf{X}$  consists of  $n$  number of observations and  $p$  variables are usually statistically monitored using the  $T^2$  and  $Q$  or  $SPE$  statistical process control chart. The threshold limit of the control chart upper and lower bound are calculated as follows (Cacciarelli and Kulahci 2022; Gajjar and Palazoglu 2016; Kourti and MacGregor 1995; Reinartz, Kulahci, and Ravn 2021):

$$L = \frac{p(m+1)(n-1)}{mn-m-p+1} F_{p, mn-m-p+1}(\alpha) \quad (1)$$

Where  $F_{j,k}(\alpha)$  is  $F$  distribution with  $j$  and  $k$  degree of freedom. The value of  $T^2 > L(T^2)$  and  $Q > L(Q)$  are considered as out of bounds. In Section 1, we discussed the assumption that the  $T^2$  and  $Q$  distribution in earlier studies was usually for supervised manners but in unsupervised conditions, an unknown distribution comes to attention.

The  $T^2$  and  $Q$  are, on the other hand, calculated based on Equations (2) and (3) (Cacciarelli and Kulahci 2022; Gajjar

and Palazoglu 2016; Reinartz et al. 2021) respectively. The representation of  $T^2$  and  $Q$  relies on the reconstruction error,  $e_i$  between the learned model,  $f$  from the data  $\mathbf{X}$  and the estimated value of  $\bar{x}_i$  where  $\mathbf{S}$  is the estimated covariance matrix of  $p$  variables. Thus, the selection of model,  $f$  reflect the ability of the model to learn the signal pattern and best representation of data  $X$  in normal conditions.

$$T^2 = (X_i - \bar{x}_i)^T S^{-1} (X_i - \bar{x}_i) \quad (2)$$

$$Q = e_i e_i^T = (X_i - \bar{x}_i)^T (X_i - \bar{x}_i) \quad (3)$$

The criteria that affect the selection of model,  $f$  also already been discussed in Section 1, but in summary, they are listed below:

1. A multivariable signal with various types of signals does not limit itself to a steady-state signal only. Some of the variables are in highly noisy, random, or sinusoidal conditions with an unknown distribution.
2. The faulty type covers steps, random variation, slow drift, sticking, constant position, and an unknown type of source.
3. Not only spatial relations but also temporal relations in the signal in the system.

### 2.2. Long Short-Term Memory (LSTM)

LSTM is considered part of the recurrent neural network (RNN) family that overcomes the vanishing gradient problem in the long-term dependency of temporal signals. The architecture of LSTM, as shown in Fig. 1, consists of four (4) elements which are an input gate, a cell, an output gate, and a forget gate [47]. The main aim of the listed components is to control the memory state,  $c$  either to keep the previous information or let it pass to the next state while accepting new information.

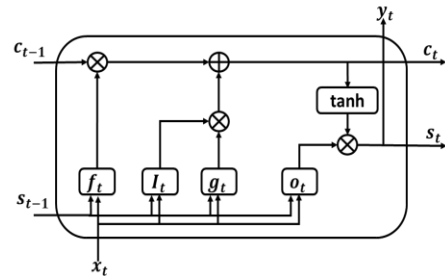


Fig. 1. The diagram of an LSTM cell of the  $l$ th layer at time  $t$ .

Initially, the normal data matrix,  $X_t$  and the previous state  $s_{t-1}$  will pass through the forget gate,  $f_t$ . At the forget gate, the information flow will be controlled by an element-wise sigmoid function as described in Equation (4). The  $W_f$  and

$U_f$  represent the weight of  $x_t$  and  $s_{t-1}$ , whereas  $b_f$  is the bias weight vector.

$$f_t = \sigma(W_f x_t + U_f s_{t-1} + b_f) \quad (4)$$

At the input gate, two steps of learning appear. The normal data matrix,  $X_t$  and the previous state  $s_{t-1}$  to be fed into the sigmoid function of  $I_t$  (see Equation (5)) for input learning as well as to  $g_t$  (see Equation (6)) to new possible cell state with the tanh activation function. After that, the new cell state  $c_t$  was updated by using Equation (7).

$$I_t = \sigma(W_i x_t + U_i s_{t-1} + b_i) \quad (5)$$

$$g_t = \tanh(W_g x_t + U_g s_{t-1} + b_g) \quad (6)$$

$$c_t = f_t \times c_{t-1} + I_t \times g_t \quad (7)$$

From Equation (7) above, the ‘‘memory’’ that will be held by  $c_t$  affected the forget gate function,  $f_t$ . The value of  $f_t$  either 1 or 0 indicates whether the previous  $c_{t-1}$  will be forgotten or not. If yes, only current memory  $I_t g_t$  will be considered for the next LSTM cell, and vice versa. Lastly, at the output gate, the output,  $o_t$  is determined by the previous state of  $c_{t-1}$ , input data matrix,  $X_t$  and previous output state,  $s_{t-1}$ , and later, the hidden state,  $s_t$  is updated accordingly.

$$o_t = \sigma(W_o x_t + U_o s_{t-1} + b_o) \quad (8)$$

$$s_t = o_t \times \tanh(c_t) \quad (9)$$

The  $W_i, W_g, W_o, U_i, U_g, U_o$  represent the weight and  $b_i, b_g, b_o$  are the bias vectors at respective input, output, and cell states.

### 2.3. Kernel Density Estimation for non-parametric threshold determination

Kernel density estimation (KDE) is a non-parametric approach to estimating an unknown distribution of dataset,  $A = \{a_1, a_2, a_3, \dots, a_i\}$ . The probability of a given point  $a_i$  in the KDE distribution is calculated by:

$$\hat{f}_h(a_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{a-a_i}{h}\right) \quad (10)$$

The kernel function  $K(\cdot)$  represents the shape of distributed data. Various types of kernel functions are used in the KDE approach, such as the Epanechnikov kernel function (Hwang, Lay, and Lippman 1994) the Gaussian kernel function (Stępień 2016; Węglarczyk 2018), the Radial Basis function (Stępień 2016) and etc (Jaya et al. 2021). In this study, the Gaussian kernel function is used as follows:

$$K(a) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{a^2}{2}\right) \quad (11)$$

In selecting the optimum value of  $h$ , in this study, the Silverman [42] bandwidth Equation is used (see Equation (12)), where IQR is the interquartile range value.

$$h = 0.9 \min\left(\sigma, \frac{IQR}{1.34}\right) \cdot n^{-1/5} \quad (12)$$

Referring to Equation (1), the threshold value was initially determined based on the percentile of the  $F_{j,k}(\alpha)$  distribution. Due to the non-parametric distribution of KDE, the upper and lower limit,  $L$  is calculated from the confidence interval of the KDE distribution (Węglarczyk 2018). The confidence interval for the upper and lower limits of KDE is given in Equation (13) below.

$$p(a_l \leq \hat{a} \leq a_u) = \int_{a_l}^{a_u} \hat{f}(a_i) da = 1 - \alpha \quad (13)$$

Where  $a_l$  and  $a_u$  are the lower and upper limits of the confidence interval within the probability of  $1 - \alpha$ . In realising the integral under the curve of  $\hat{f}(a_i)$ , a numerical approach trapezoid rule method (Yu et al. 2018) is used as stated below.

$$\widehat{F}_h(A) = \int_{a_l}^{a_u} \hat{f}(a_i) da = \sum_{i=1}^n \frac{\Delta x}{2} (\hat{f}(a_i) + \hat{f}(a_{i-1})) = 1 - \alpha \quad (14)$$

The numerical approach of the trapezoid rule in computational implementation is influenced by the selection of trapezoid size. In this study, the trapezoid size was tuned heuristically since each situation creates a different KDE distribution and a different trapezoid size for  $1 - \alpha$  condition.

### 3. NON-PARAMETRIC LSTM FAULT DETECTION FRAMEWORK

A data-driven fault detection approach usually has two stages of process flow. The first one is the training phase, which relates to normal dataset learning. Considering the criteria in Section 2.1, this study proposed the LSTM network as the learning model,  $f$ . The accuracy of the stage is important, as the reconstruction error will reflect the final distribution of the system. Initially, a normal data matrix,  $X$  goes through normalisation pre-processing. After that, the dataset will be rearranged between input and target output in a time-series manner by looking back at a certain period of time,  $l$ . Later, the arranged dataset will be trained using a stacked LSTM network, and  $Q$  parameters of the control chart will be determined. The distribution of  $Q$  is modelled using the KDE distribution, and the trapezoid rule is used to determine upper and lower boundaries to complete the normal control chart, as shown in Fig. 2.

At the second stage, faulty data is fed into the trained LSTM network after going through pre-processing and lookback functions similar to the first stage. The  $Q$  for faulty dataset are determined, checking whether at a certain time, the condition is ‘faulty’ or not based on the upper and lower limits from the analysis at the 1<sup>st</sup> stage. Fig. 3. illustrates the fault detection phase. Table 1. and Table 2. represent the flow chart in term of algorithm for the LSTM-KDE fault detection framework.

### 3.1. Performance Measure

In determining the ability of model,  $f$  in learning the structure or pattern in a normal condition, mean squared errors are used since they represent the reconstruction error of the normal condition of the system. A lower MSE means a more accurate learning model.

$$MSE = \frac{1}{n} \sum_{i=1}^n (X_i - \hat{X}_i)^2 \quad (15)$$

Another performance measure that closely relates to fault detection is fault detection accuracy, false positive rate (FPR), and false negative rate (FNR) based on the confusion matrix. Fault detection accuracy is based on the number of accurate fault detections, whereas FPR calculates the condition where there is a 'faulty' condition but estimated as a 'non-faulty', type I error.

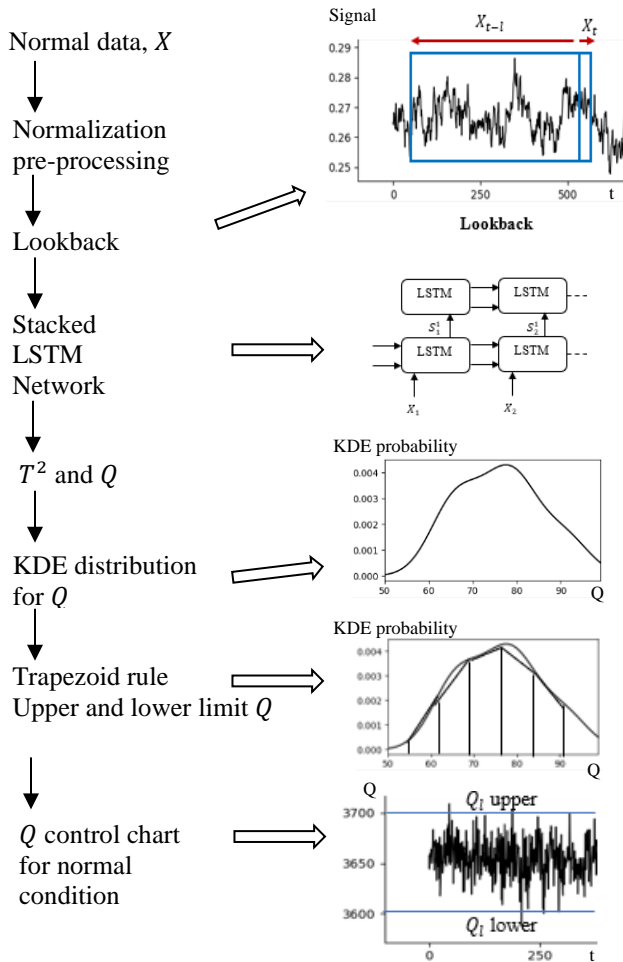


Fig. 2. Training phase

On the other hand, FNR calculates the 'non-faulty' condition as a 'faulty', type II error. FPR values are important to keep as low as possible where a high misdetection rate is open to equipment damage in the process plant.

$$Accuracy = \frac{TP+TN}{P+N} \quad (16)$$

$$FPR = \frac{FP}{N} \quad (17)$$

$$FNR = \frac{FN}{P} \quad (18)$$

The TP and TN represent true positive and true negative, whereas FP and FN indicate false positive and false negative. In addition, P and N are the number of positive and negative value.

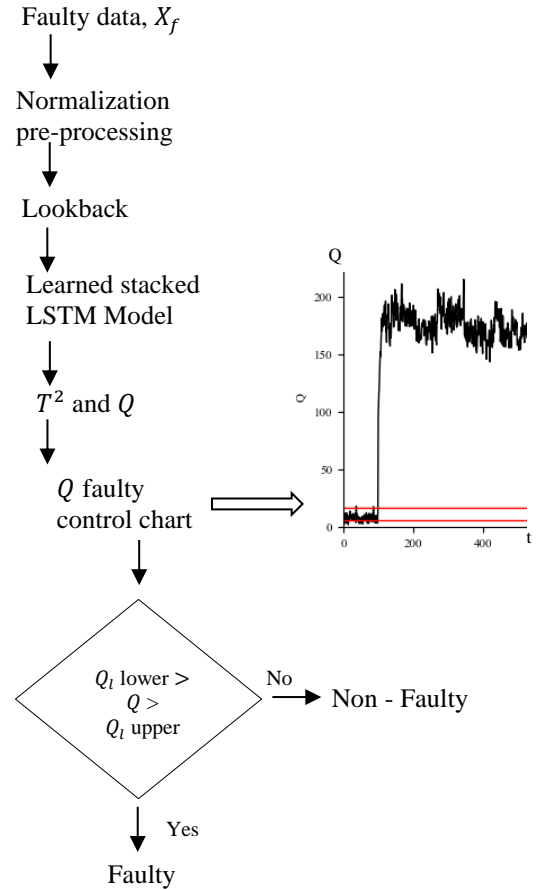


Fig. 3. Fault detection phase

#### LSTM-KDE Train algorithm

**Input:** Normal Dataset,  $X = x_1^1, x_2^1, \dots, x_i^j$

**Output:** Upper and Lower Threshold

1. Normalization pre-processing
2. Create lookback dataset for train normal dataset and test normal dataset
3. Train normal dataset:

#### LSTM model:

Stack LSTM (train normal dataset, layer, epoch = 100, optimizer = Root mean square propagation (RMSProp)), validation split = 0.2)

$\bar{x}_i = \text{predict}(\text{Stack LSTM}(\text{test normal set}))$

$Q_i = (X_i - \bar{x}_i)^T (X_i - \bar{x}_i)$

**KDE distribution:**

$q_i = \text{random choice } (Q_i, n = \text{number of sample})$   
 // n are heuristically select between 37 to 200 depends on Modes in TE process.

$$h = 0.9 \min \left( \sigma, \frac{IQR}{1.34} \right) \cdot n^{-1/5}$$

$$\text{KDE distribution} = \frac{1}{nh} \sum_{i=1}^n K \left( \frac{q_i}{h} \right)$$

Upper Threshold, Lower Threshold =

Trapezoid rule (KDE distribution,  $\alpha=0.05$ )

Table 1. LSTM-KDE Train algorithm

**LSTM-KDE Fault detection algorithm**

**Input:** Faulty Dataset,  $X = x_1^1, x_2^1, \dots, x_i^j$

**Output:** faulty or non-faulty

1. Normalization pre-processing
2. Create lookback dataset for train and testing
3. Faulty dataset:

**LSTM model:**

$\bar{x}_{fi} = \text{predict}(\text{Stack LSTM}(\text{test normal set}, \dots))$

$$Q_i = (X_{fi} - \bar{x}_{fi})^T (X_{fi} - \bar{x}_{fi})$$

**Faulty condition determination:**

for  $i = 1, \dots, n$ :

If  $Q_i > \text{Upper Threshold}$  and  $Q_i < \text{Lower Threshold}$   
 faulty

else:

non-faulty

end for

Table 2. LSTM-KDE Fault detection algorithm

**3.2. Experimental setup****NUMERICAL EXAMPLE**

The common responses that usually occur inside engineering systems are a) nonlinear conditions and b) varying operating system. In this section, these two common responses were produced and illustrated using a numerical example (Yu et al. 2018). The multivariate nonlinear process signal for normal conditions was generated based on mathematical expression (19) where  $t$  were set within the boundary  $[0.01, 2]$  and  $e_1, e_2$  and  $e_3$  are noise of each condition with a normal distribution of mean 0 and variance 0.01. As for faulty signal, two types of faulty signal were produced: a linearly increasing condition was injected into the generated data of  $x_1$  from sample 101-270 by adding  $0.01(k - 100)$  to  $x_1$  value of each sample in this range, where  $k$  is the sample number and a step (as faulty 1). In addition, another type of faulty signal, which is a step response, was inputted to  $x_2$  by 1.5 from sample 101 onwards as faulty 2.

$$x = \begin{cases} t + e_1 \\ t^2 - 3t + e_2 \\ -t^3 + 3t^2 + e_3 \end{cases} \quad (19)$$

As for the varying operating system condition, the system was described using the mathematical expression in Equation (20), where  $s_1$  and  $s_2$  follow normal distribution with a given mean and variance value as stated in Equations (21), (22), and

(23). The data was generated from 100 samples for each mode, resulting in 300 samples of normal data for the training phase. Otherwise, 200 test samples were generated due to the faulty condition. At the first faulty condition (faulty 1), the system was initially running in mode 2, and a drifting error of 0.04 ( $k - 100$ ) was applied to  $x_2$  from the 101st through the 200th samples, where  $k$  denotes the serial number of the test samples. In addition, for the second faulty condition (faulty 2), a step signal of 2 was added to  $x_1$  at Mode 1.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.3723 & 0.6815 \\ 0.4890 & 0.2954 \\ 0.9842 & 0.1793 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (20)$$

$$\text{Mode 1: } s_1 \sim N(10, 0.8), s_2 \sim N(12, 1.3) \quad (21)$$

$$\text{Mode 2: } s_1 \sim N(5, 0.6), s_2 \sim N(20, 0.7) \quad (22)$$

$$\text{Mode 3: } s_1 \sim N(16, 1.5), s_2 \sim N(30, 2.5) \quad (23)$$

**TENNESSEE EASTMAN PROCESS PLANT(TE)**

The TE benchmark process plant has been considered one of the most challenging industrial engineering problems in monitoring systems as well as in fault detection (Downs and Vogel 1993; Xiao et al. 2022). Many studies have referred to the TE process plant as the standard benchmark in the fault detection and diagnosis area, such as (Cacciarelli and Kulahci 2022; Downs and Vogel 1993; Gajjar and Palazoglu 2016). Recently, an extended version of the TE process plant has been published (Reinartz et al. 2021) and data has been shared at [https://data.dtu.dk/articles/dataset/Tennessee\\_Eastman\\_Reference\\_Data\\_for\\_Fault\\_Detection\\_and\\_Decision\\_Support\\_Systems/13385936/1](https://data.dtu.dk/articles/dataset/Tennessee_Eastman_Reference_Data_for_Fault_Detection_and_Decision_Support_Systems/13385936/1) which is in line with present plant system. Compared to the existing TE process plant dataset that focuses only on Mode 1 of the system, the extended dataset consists of Mode 1 until Mode 6 of the system. At each mode, there will be fifty-four (54) measured and manipulated variables. As for this study, only thirty-three (33) variables were used, as we referred to (Cacciarelli and Kulahci 2022) (see Table 3). Furthermore, each mode also contained twenty-eight (28) faulty conditions with steps, random variation, sticking, and unknown types of fault, as shown in Table 4.

No	Description	No	Description
pd(2)	A Feed (kscmh)	pd(19)	Stripper Temp (° C)
pd(3)	D Feed (kg/h)	pd(20)	Stripper Steam Flow (kg/h)
pd(4)	E Feed (kg/h)	pd(21)	Compressor Work (kW)
pd(5)	A and C Feed (kscmh)	pd(22)	Reactor Coolant Temp (° C)
pd(6)	Recycle Flow (kscmh)	pd(23)	Separator Coolant Temp (° C)
pd(7)	Reactor Feed Rate (kscmh)	pd(43)	D feed (%)
pd(8)	Reactor Pressure (kPa gauge)	pd(44)	E feed (%)
Pd(9)	Reactor Level (%)	pd(45)	A feed (%)
pd(10)	Reactor Temperature (° C)	pd(46)	A and C feed (%)

pd(11)	Purge Rate (kscmh)	pd(47)	Compressor recycle valve (%)
pd(12)	Product Sep Temp ( $^{\circ}$ C)	pd(48)	Purge valve (%)
pd(13)	Product Sep Level (%)	pd(49)	Separator liquid flow (%)
pd(14)	Product Sep Pressure (kPa gauge)	pd(50)	Stripper liquid flow (%)
pd(15)	Product Sep Underflow (m3/h)	pd(51)	Stripper steam valve (%)
pd(16)	Stripper Level (%)	pd(52)	Reactor Coolant (%)
pd(17)	Stripper Pressure (kPa gauge)	pd(53)	Condenser Coolant (%)
pd(18)	Stripper Underflow (m3/h)		

Table 3. Monitoring variables in TE process plant

Variable	Process variable	Faulty type
IDV 1	A/C feed ratio, B composition constant (stream 4)	Step
IDV 2	B composition, A/C ratio constant (stream 4)	Step
IDV 3	D feed temperature (stream 2)	Step
IDV 4	Water inlet temperature for reactor cooling	Step
IDV 5	Water inlet temperature for condenser cooling	Step
IDV 6	A feed loss (stream 1)	Step
IDV 7	C header pressure loss (stream 4)	Step
IDV 8	A/B/C composition of stream 4	Random variation
IDV 9	D feed (stream 2) temperature	Random variation
IDV 10	C feed (stream 4) temperature	Random variation
IDV 11	Cooling water inlet temperature of reactor	Random variation
IDV 12	Cooling water inlet temperature of separator	Random variation
IDV 13	Reaction kinetics	Random variation
IDV 14	Cooling water outlet valve of reactor	Sticking
IDV 15	Cooling water outlet valve of separator	Sticking
IDV 16	Variation coefficient of the steam supply of the heat exchange of the stripper	Random variation
IDV 17	Variation coefficient of heat transfer in reactor	Random variation
IDV 18	Variation coefficient of heat transfer in condenser	Random variation
IDV 19	Unknown	Unknown
IDV 20	Unknown	Random variation
IDV 21	A feed (stream 1) temperature	Random variation
IDV 22	E feed (stream 3) temperature	Random variation
IDV 23	A feed flow (stream 1)	Random variation
IDV 24	D feed flow (stream 2)	Random variation
IDV 25	E feed flow (stream 3)	Random variation
IDV 26	A and C feed flow (stream 4)	Random variation
IDV 27	Reactor cooling water flow	Random variation
IDV 28	Condenser cooling water flow	Random variation

Table 4. Faulty condition in TE process plant

To evaluate the proposed LSTM fault detection framework, three other fault detection approaches were used. The details of each approach are mentioned below:

1. The PCA+KDE approach was inspired by (Jaya et al. 2021) where the study argues that the fault detection of multivariable condition control charts is better approximated using the KDE approach to reduce high

false alarms in nonparametric data distribution. Furthermore, PCA was considered the most prominent approach and is still, to date, being chosen as the fault detection approach, making it an appropriate comparison model to evaluate the proposed fault detection framework.

2. The LOF approach was referred to in (Benkő, Bábel, and Somogyvári 2022) which defines a faulty condition in terms of distance instead of distribution. The study also emphasises unsupervised as well as nonlinear conditions, which is in line with our study.
3. Another approach chosen as a comparison method in this study is auto-associative kernel regression (AAKR) with KDE fault detection approach (Yu et al. 2018). This study focuses on the multimode approach from a regression perspective. The research also explains that the non-parametric distribution relies on a multimode dataset, which confirms the importance of this study.

## 4. RESULTS AND DISCUSSION

### 4.1. Evaluation of model, $f$

The selection of model,  $f$  reflects the ability of the model to learn the signal. The aim is to analyse the MSE value for reconstruction error approach that has been proposed in Section 3. Based on Section 2, there are a number of deep learning approaches within the scope of this study. However, considering the nonlinear and multimode characteristics of engineering systems, this section further analyses the impact of three types of model,  $f$ , which are our proposed stacked LSTM, LSTM-AE (Machado et al. 2022), CNN-LSTM (B. Zhao et al. 2020), and BiLSTM (J. Zhao et al. 2020). The numerical multimode dataset was used to evaluate the performance of each selected model,  $f$ . The dataset was pre-processed using min-max scaling, which brings the value between 0 and 1. After that, the data were split between train input and output by the lookback function, using  $x_{t=0}$  until  $x_{t=l}$  to predict the next output  $x_t$ . The value  $l$  in this study was set as eight, epoch equal to 100, number of neurons nine and the optimizer used in this study is RMSprop using the TensorFlow Python library.

Table 5 shows the average accuracy and losses for each selected model,  $f$  for 100 epochs. From the observation of accuracy at Table 5, CNN-LSTM showed the lowest accuracy among all with 0.8913 followed by LSTM-AE with 0.9289 which leaves the option between BiLSTM and stacked LSTM with 0.9513 and 0.9397 respectively. When compared to the two-left approach in term of  $\Delta Acc$  which difference between train accuracy and validation accuracy, stacked LSTM shows better result with 0.1646 compared to BiLSTM. BiLSTM was interpreted as overfit since high different between train and validation accuracy results. Thus, this experiment concludes that a stacked LSTM architecture

will be used as main the model,  $f$  approach for unsupervised fault detection.

#### 4.2 Evaluation of threshold determination based on KDE distribution.

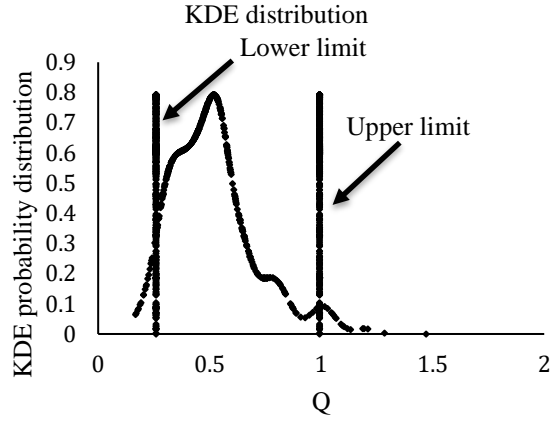


Fig. 4. KDE distribution for Tennessee Eastman process plant mode 1

This section describes KDE distribution shape of the proposed framework for non-parametric fault detection LSTM-KDE approach. Fig. 4 shows The Tennessee Eastman process plant mode 1 that were selected as visual representation of KDE distribution in this study. Refer to the algorithm Table. 1, the number of kernels for this distribution was set as  $n = 150$  with  $\alpha = 0.05$ . From the algorithm the upper and lower threshold limit are, upper limit = 0.9935 and lower limit = 0.2604 that represent in vertical line at Fig.4.

#### 4.3 Evaluation of LSTM fault detection performance based on KDE threshold approach.

This section describes the performance of the proposed framework for non-parametric fault detection LSTM-KDE approach. The generated dataset, as mentioned in Section 3.2 numerical example, was used to evaluate the accuracy of the LSTM-KDE approach compared to the selected model in Section 3.3. Tables 7. and 6. show the performance indices of each method. For both multimode and nonlinear conditions, fault 1 illustrates amplitude increment fault, whereas fault 2 indicates a linear increment fault.

For multimode fault 1 at Table 7, the lowest accuracy is at AAKR+KDE with only 0.395, followed by LSTM+KDE with 0.4136, LOF with 0.5 and the highest is PCA+KDE with 0.605. Similar to the nonlinear condition of fault 1, LOF again has the lowest accuracy of 0.33, followed by LSTM+KDE at 0.48, PCA+KDE at 0.74, and AAKR+KDE at 0.787.

As for multimode fault 2, PCA+KDE, LOF, and AAKR+KDE have nearly the same values of 0.53, 0.5, and 0.54, whereas LSTM+KDE shows the overall highest value with 0.6701. In nonlinear fault 2, the lowest accuracy is

PCA+KDE with 0.65, AAKR+KDE with 0.747, LOF with 0.76, and the highest is LSTM+KDE at 0.804. This indicates the LSTM+KDE framework is able to detect linear increment faults better than other methods. However, different performance in terms of multimode and amplitude increment faults. The overlapped learning between multimode conditions and steady-state increments makes it difficult to detect faulty conditions. This will be further discussed based on Figure 5.

Fig. 5. illustrates the Q residual value for the LSTM-KDE approach for multimode and nonlinear conditions. The figures show a clear detection for a linear increment fault (fault 2) starting at 100 and above based on the calculated threshold value, whereas not for the fault 1 condition. The multimode condition fault 1 Q residual signal response does not represent the steady-state increment at all, thus there is no faulty signal indication, which is in line with the low detection rate. In the nonlinear case, the steady-state was observed as seen in the figure fault 1 nonlinear; however, the threshold value is not accurate for fault detection in this area. Thus, further improvement is needed for the LSTM-KDE framework to be able to work for both nonlinear and multimode conditions in all fault signals.

Model, $f$	Train Acc	Train MSE	Val Acc	$\Delta$ Acc
Stacked LSTM	0.9397	0.0074	0.7751	0.1646
LSTM-AE	0.9289	0.0065	0.7888	0.1401
CNN-LSTM	0.8913	0.0109	0.7833	0.1080
BiLSTM	0.9513	0.0061	0.7622	0.1891

Table 5. Average accuracy and loss for 100 epochs

Fault	NONLINEAR SIMULATED DATA					
	PCA+KDE (upper= 1.355, lower= 0.364)			LOF		
	ACC	FPR	FNR	ACC	FPR	FNR
1	0.74	0.758	0	0.33	0.0	1.0
2	0.65	0.758	0.121	0.76	0.0	0.927

Fault	NONLINEAR SIMULATED DATA					
	AAKR+KDE (upper = 0.0392, lower = 0.00861)			LSTM+KDE (Upper = 0.99182, Lower = 0.02793)		
	ACC	FPR	FNR	ACC	FPR	FNR
1	0.787	0.63	0.0	0.48	0.162	0.695
2	0.747	0.59	0.076	0.804	0.161	0.197

Table 6. Performance indices for nonlinear condition

Fault	MULTIMODE SIMULATED DATA					
	PCA+KDE (upper= 3.696, lower= 0.357)			LOF		
	ACC	FPR	FNR	ACC	FPR	FNR
1	0.605	0.08	0.707	0.5	0.0	1.0
2	0.53	0.02	0.919	0.5	0.0	1.0

Fault	MULTIMODE SIMULATED DATA					
	AAKR+KDE (upper = 0.18, lower = 0.0115)			LSTM+KDE (Upper = 0.01189, Lower = 0.0010)		
	ACC	FPR	FNR	ACC	FPR	FNR
1	0.395	0.32	0.89	0.4136	0.37	0.8222
2	0.54	0.53	0.38	0.6701	0.4	0.2



Table 7. Performance indices for multimode condition

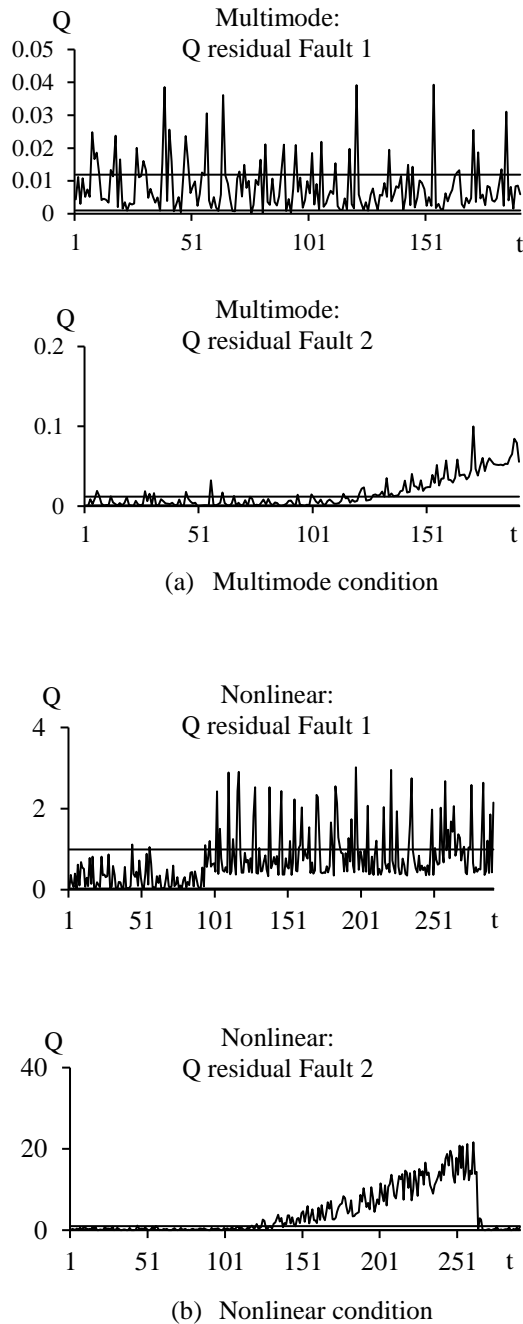


Fig. 4. Q residual control chart with faulty condition and threshold value obtained from fault detection LSTM framework: (a) Multimode condition and (b) Non-linear condition.

#### 4.4 Evaluation of fault detection performance in large scale plant: Tennessee Eastman process plant

In the training phase, at each mode there are 2001 point of data. The normal signal at each faulty data point from time 1 to time 500 were grouped as normal datasets, and pre-processing was done using a minimum-maximum scaling of 0 to 1. After that, the data were fed into a non-parametric framework similar to Section 3. However, the number of stacked LSTM neurons at each layer was changed to thirty-three (33) neurons, value  $l$  in this study was set as twenty (20) and other hyperparameters remain the same as in section 4.1. The remaining data were used in the testing phase, which is data from 501 to 600 that are normal and data from 600 to 2001 that are faulty. The performance indices of accuracy, FPR, and FNR were calculated at the last stage. The indices are shown in Table 8. until Table 11. We would like to note that in Mode 2, IDV 17, 18, and 28 were unable to be extracted, thus implicating the simulation with no results for these three types of faults and not included in the average calculation.

In the section 3.2 described the faulty condition consist of IDV 1 to IDV 28 with different type of faults. At the initial observation of accuracy results, author found two types of group appear, which is the detection accuracy between 80% until 100% named as Group 1 (IDV 1, IDV 2, IDV 4, IDV 6, IDV 7, IDV 8, IDV 10, IDV 11, IDV 13, IDV 14, IDV 17, IDV 18, IDV 19, IDV 20, IDV 25, IDV 26, and IDV 27) whereas outside this interval as Group 2 (IDV 3, IDV 5, IDV 9, IDV 12, IDV 15, IDV 16, IDV 21, IDV 22, and IDV 23). Group 2 was also considered as difficult faulty event to detect in past literature of (Cacciarelli and Kulahci 2022; Gajjar and Palazoglu 2016; Reinartz et al. 2021).

Table 8 for faulty condition under Group 1 shows that LSTM have the highest accuracy in average with 0.94 follow by PCA and AAKR with 0.93 whereas LOF become the lowest with 0.54. Similar performance can be seen at Table 9 for Group 2 faulty condition, LSTM also shows superior results compare to other method but with the percentage of 0.52, LOF with 0.51, PCA with 0.35 and the least accuracy is AAKR with 0.31. Despite of the results, with LSTM performance in Group 2 (0.52) still consider low accuracy for fault detection approach.

Further analysis was made on the performance indices of false positive rate (FPR) and false negative rate (FNR). In FPR situation, when the signals are supposedly indicated as faulty, the real output shows a normal condition. This is dangerous when considering interconnected process plants, which promote damage to machines. The FNR, on the other hand, expected a normal condition but a faulty signal to appear, thus wasting resources when the faulty signal does not actually appear. Thus, it is the best to keep these indexes as low as possible. Table 10 and Table 11 show the results for these indices. For FPR index, lowest rate goes to AAKR with 0.20, after that PCA with 0.23, next LOF with 0.25 and

the highest is on LSTM which is 0.52. As for FNR index, LSTM shows the best performance with 0.19, follow by PCA is 0.28, AAKR is 0.30 and lowest performance on LOF with 0.49.

Refer back to the study of the TE process plant in (Reinartz et al. 2021) shows fault detection based on PCA and explains that the  $ARL_0$  is not always in a nominal state, which impacts the  $T^2$  chart. The main reason is that the assumptions of observation are uncorrelated in terms of time. At the same time, it is also mentioned in (Gajjar and Palazoglu 2016) that the  $T^2$  control chart is not reliable and modifies the loss function in the study. This study proposes a stacked LSTM framework to cover the time-correlated control chart while focusing on SPE or Q residual response for fault detection. Using SPE measurements LSTM shows significant improvement compared to PCA in all modes except mode 5, in which PCA shows better performance, whereas AAKR always results in the middle between the other two approaches. From the 28 faulty conditions labelled as IDV 1-28, faults 1, 2, 4, 6, 7, 8, 10, 11, 13, 14, 17, 18, 19, 20, 25, 26, and 27 show a good result of more than 90%, but concern is raised at faults 3, 5, 9, 12, 15, 16, 21, 22, and 23. This is in line with the discussion in (Reinartz et al. 2021), which imposed that the faults in 3 and 15 are difficult to catch, as well as the multiple fault type of random variation and sticking in 9, 15, 20, and 23. However, the proposed LSTM framework is able to improve the fault in 13 and 18, which in (Reinartz et al. 2021) are also labelled as difficult to detect faults. Consider the SPE Equation (3), which relates to the matrix decomposition of highly correlated and multivariable variables might jeopardize the fault detection accuracy. For example, IDV 1, in which significant individual variables are out of control, shows a highly visible area between faulty and non-faulty conditions (Cacciarelli and Kulahci 2022), making the determined threshold value ample for fault detection, whereas other faulty conditions consist of fewer variables, shadowing the boundary between faulty and non-faulty detection. This individual signal condition as well as the correlation of variables are not included in this study and might become the next focus to improve the LSTM framework.

Method	PCA	LOF	AAKR	LSTM
Mode 1	0.94	0.53	0.96	<b>0.96</b>
Mode 2	0.94	0.55	0.93	<b>0.93</b>
Mode 3	0.89	0.55	0.93	<b>0.94</b>
Mode 4	0.97	0.55	0.96	<b>0.96</b>
Mode 5	0.93	0.54	0.89	<b>0.94</b>
Mode 6	0.90	0.52	0.89	<b>0.93</b>
Average	0.93	0.54	0.93	<b>0.94</b>

Table 8. Performance index – accuracy for Group 1 faulty condition

Method	PCA	LOF	AAKR	LSTM
Mode 1	0.29	0.51	0.24	<b>0.41</b>
Mode 2	0.24	0.51	0.23	<b>0.28</b>
Mode 3	0.57	0.51	0.51	<b>0.64</b>
Mode 4	0.34	0.50	0.38	<b>0.74</b>
Mode 5	0.25	0.51	0.15	<b>0.25</b>
Mode 6	0.39	0.53	0.38	<b>0.80</b>
Average	0.35	0.51	0.31	<b>0.52</b>

Table 9. Performance index – accuracy for Group 2 faulty condition

Mode	PCA	LOF	AAKR	LSTM
	FPR	FPR	FPR	FPR
Mode 1	0.29	0.32	0.12	<b>0.33</b>
Mode 2	0.09	0.18	0.12	<b>0.22</b>
Mode 3	0.58	0.19	0.34	<b>0.48</b>
Mode 4	0.14	0.20	0.32	<b>0.59</b>
Mode 5	0.16	0.22	0.06	<b>0.13</b>
Mode 6	0.22	0.26	0.24	<b>0.70</b>
Average	0.25	0.23	0.20	<b>0.41</b>

Table 10. Performance index – False positive rate (Type I error)

Mode	PCA	LOF	AAKR	LSTM
	FNR	FNR	FNR	FNR
Mode 1	0.29	0.49	0.31	<b>0.23</b>
Mode 2	0.33	0.48	0.34	<b>0.31</b>
Mode 3	0.20	0.48	0.21	<b>0.14</b>
Mode 4	0.27	0.49	0.25	<b>0.09</b>
Mode 5	0.32	0.49	0.40	<b>0.32</b>
Mode 6	0.28	0.49	0.30	<b>0.07</b>
Average	0.28	0.49	0.30	<b>0.19</b>

Table 11. Performance index – False negative rate (Type II error)

## 5. CONCLUSION

A time-series fault detection study based on the LSTM framework was proposed in this section. The main reason is to overcome the non-stationary assumption in previous studies, such as in (Cacciarelli and Kulahci 2022; Reinartz et al. 2021). In the LSTM framework, KDE threshold

determination was introduced since the majority of the gathered datasets have an unknown distribution and most of the studies assume the collected signal is a gaussian assumption (Guo et al. 2020; Kukkala et al. 2020; Machado et al. 2022; Silva et al. 2022; B. Wang, Liu, et al. 2020; J. Zhao et al. 2020). The study compared four types of LSTM variants as the main learning model and came up with staked LSTM as the best model for fault detection. The study also analyses the numerical representation of multimode and nonlinear conditions, as well as the real representation of the Tennessee Eastman process plant for the proposed LSTM fault detection framework in comparison to the PCA+KDE, LOF, and AAKR+KDE approaches. The majority of results show the dominance of the LSTM approach, but

improvements must be made to cater to the high correlation and individual signal contribution towards the accuracy of fault detection as well as the challenge of multimode conditions in the LSTM fault detection framework for future work.

#### ACKNOWLEDGEMENT

Thanks to Universiti Teknologi Malaysia and Universiti Teknikal Malaysia Melaka for providing their available software platform to conduct this research work. We also extend our appreciation to the Ministry of Higher Education Malaysia in providing student scholarship to author in completing the study.

#### REFERENCES

- Alrifayy, M., Lim, W. H., and Ang, C. K. (2021). "A Novel Deep Learning Framework Based RNN-SAE for Fault Detection of Electrical Gas Generator." *IEEE Access* 9:21433–42. doi: 10.1109/ACCESS.2021.3055427.
- Amarbayasgalan, T, Van, H. P., Nipon T., and Keun, H R. (2020). "Unsupervised Anomaly Detection Approach for Time-Series in Multi-Domains Using Deep Reconstruction Error." *Symmetry* 12(8):1251. doi: 10.3390/sym12081251.
- Ammann, O., Gabriel M., and Olga F. (2020). "Anomaly Detection And Classification In Time Series With Kervolutional Neural Networks."
- Anaissi, A., and Seid M. Z. (2019). "Multi-Objective Autoencoder for Fault Detection and Diagnosis in Higher-Order Data." Pp. 1–8 in *2019 International Joint Conference on Neural Networks (IJCNN)*. Budapest, Hungary: IEEE.
- Belagoune, S., Nouredine B., Azzeddine B., Bousaadia B., and Karim A. (2021). "Deep Learning through LSTM Classification and Regression for Transmission Line Fault Detection, Diagnosis and Location in Large-Scale Multi-Machine Power Systems." *Measurement* 177:109330. doi: 10.1016/j.measurement.2021.109330.
- Benkó, Z., Tamás B., and Zoltán S. (2022). "Model-Free Detection of Unique Events in Time Series." *Scientific Reports* 12(1):227. doi: 10.1038/s41598-021-03526-y.
- Cacciarelli, D., and Murat, K. (2022). "A Novel Fault Detection and Diagnosis Approach Based on Orthogonal Autoencoders." *Computers & Chemical Engineering* 163:107853. doi: 10.1016/j.compchemeng.2022.107853.
- Calabrese, F., Alberto, R., Lucia, B., Cristina, M., and Francesco, G. G. (2020). "Unsupervised Fault Detection and Prediction of Remaining Useful Life for Online Prognostic Health Management of Mechanical Systems." *Applied Sciences* 10(12):4120. doi: 10.3390/app10124120.
- Chen, T., Liu, X., Xia, B., Wang, W., and Lai, Y. (2020). "Unsupervised Anomaly Detection of Industrial Robots Using Sliding-Window Convolutional Variational Autoencoder." *IEEE Access* 8:47072–81. doi: 10.1109/ACCESS.2020.2977892.
- Dong, J., Ma, Y., and Liu, D. (2019). "Deep Learning Based Multiple Sensors Monitoring and Abnormal Discovery for Satellite Power System." Pp. 638–43 in *2019 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*.
- Downs, J. J., and Vogel. E. F. (1993). "A Plant-Wide Industrial Process Control Problem." *Computers & Chemical Engineering* 17(3):245–55. doi: 10.1016/0098-1354(93)80018-I.
- Ellefsen, A. L., Bjørlykhaug, E., Æsøy, V., and Zhang, H. (2019). "An Unsupervised Reconstruction-Based Fault Detection Algorithm for Maritime Components." *IEEE Access* 7:16101–9. doi: 10.1109/ACCESS.2019.2895394.
- Gajjar, S., and Palazoglu, A. (2016). "A Data-Driven Multidimensional Visualization Technique for Process Fault Detection and Diagnosis." *Chemometrics and Intelligent Laboratory Systems* 154:122–36. doi: 10.1016/j.chemolab.2016.03.027.
- Guo, Y., Ji, T., Wang, Q., Yu, L., Min, G., and Li, P. (2020). "Unsupervised Anomaly Detection in IoT Systems for Smart Cities." *IEEE Transactions on Network Science and Engineering* 7(4):2231–42. doi: 10.1109/TNSE.2020.3027543.
- Hwang, J., Lay, S., and Lippman., A. (1994). "Nonparametric Multivariate Density Estimation: A Comparative Study." *IEEE Transactions on Signal Processing* 42(10):2795–2810. doi: 10.1109/78.324744.
- Jaya, A. I., Aulia, T. R., Putri, F. D. and Rakhmawati, T. (2021). "T<sup>2</sup> Control Chart Based on PCA with KDE Control Limit for Monitoring Intrusion."

- Journal of Physics: Conference Series* 2123(1):012017. doi: 10.1088/1742-6596/2123/1/012017.
- Jiang, G., Xie, P., He, H., X., and Yan, J. (2018). “Wind Turbine Fault Detection Using a Denoising Autoencoder With Temporal Information.” *IEEE/ASME Transactions on Mechatronics* 23(1):89–100. doi: 10.1109/TMECH.2017.2759301.
- Jin, S., Zhang, Z., Chakrabarty, K., and Gu., X. (2020). “Self-Learning and Efficient Health-Status Analysis for a Core Router System.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39(9):1935–48. doi: 10.1109/TCAD.2019.2926506.
- Kourti, T., and MacGregor, J., F. (1995). “Process Analysis, Monitoring and Diagnosis, Using Multivariate Projection Methods.” *Chemometrics and Intelligent Laboratory Systems* 28(1):3–21. doi: 10.1016/0169-7439(95)80036-9.
- Kukkala, V. K., Thiruloga, S. V., and Pasricha, S. (2020). “INDRA: Intrusion Detection Using Recurrent Autoencoders in Automotive Embedded Systems.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39(11):3698–3710. doi: 10.1109/TCAD.2020.3012749.
- Li, W., Liu, Y., Li, Y., and Guo, F. (2020). “Series Arc Fault Diagnosis and Line Selection Method Based on Recurrent Neural Network.” *IEEE Access* 8:177815–22. doi: 10.1109/ACCESS.2020.3027002.
- Liu, L., Song, D., Geng, Z., and Zheng, Z. (2020). “A Real-Time Fault Early Warning Method for a High-Speed EMU Axle Box Bearing.” *Sensors* 20(3):823. doi: 10.3390/s20030823.
- Machado, A. P. F., Vargas, R. E. V., Ciarelli, P. M., and Munaro, C. J. (2022). “Improving Performance of One-Class Classifiers Applied to Anomaly Detection in Oil Wells.” *Journal of Petroleum Science and Engineering* 218:110983. doi: 10.1016/j.petrol.2022.110983.
- Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., and Ueda, K. (2016). “Cyber-Physical Systems in Manufacturing.” *CIRP Annals* 65(2):621–41. doi: 10.1016/j.cirp.2016.06.005.
- Nagarajan, S., Kayalvizhi, S., and Karthikeyan, B. (2016). “Neural Network Based Intelligent Sensor Fault Detection in a Three Tanks Interacting Level Process.” Pp. 2429–34 in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*.
- Principi, E., Rossetti D., Squartini, S. and Piazza, F. (2019). “Unsupervised Electric Motor Fault Detection by Using Deep Autoencoders.” *IEEE/CAA Journal of Automatica Sinica* 6(2):441–51. doi: 10.1109/JAS.2019.1911393.
- Reinartz, C., Kulahci, M., and Ravn, O. (2021). “An Extended Tennessee Eastman Simulation Dataset for Fault-Detection and Decision Support Systems.” *Computers & Chemical Engineering* 149:107281. doi: 10.1016/j.compchemeng.2021.107281.
- Ren, H., Chai, Y., Qu, J., Ye, X., and Tang, Q., (2018). “A Novel Adaptive Fault Detection Methodology for Complex System Using Deep Belief Networks and Multiple Models: A Case Study on Cryogenic Propellant Loading System.” *Neurocomputing* 275:2111–25. doi: 10.1016/j.neucom.2017.10.063.
- Shadi, M. R., Ameli, M., and Azad, S. (2022). “A Real-Time Hierarchical Framework for Fault Detection, Classification, and Location in Power Systems Using PMUs Data and Deep Learning.” *International Journal of Electrical Power & Energy Systems* 134:107399. doi: 10.1016/j.ijepes.2021.107399.
- Shao, K., He, Y., Xing, Z., and Du, B. (2023). “Detecting Wind Turbine Anomalies Using Nonlinear Dynamic Parameters-Assisted Machine Learning with Normal Samples.” *Reliability Engineering & System Safety* 233:109092. doi: 10.1016/j.res.2023.109092.
- Silva, M. F., Pacini, A., Sgambelluri, A. and Valcarengi, L. (2022). “Learning Long-and Short-Term Temporal Patterns for ML-Driven Fault Management in Optical Communication Networks.” *IEEE Transactions on Network and Service Management* 1–1. doi: 10.1109/TNSM.2022.3146869.
- Stępień, B. (2016). “Confidence Intervals for the Long-Term Noise Indicators Using the Kernel Density Estimator.” *Archives of Acoustics* 41(3):517–25. doi: 10.1515/aoa-2016-0050.
- Tang, P., Peng, K., Zhang, K., Chen, Z., Yang, X. and Li, L. (2018). “A Deep Belief Network-Based Fault Detection Method for Nonlinear Processes.” *IFAC-PapersOnLine* 51(24):9–14. doi: 10.1016/j.ifacol.2018.09.522.
- Wang, B., Liu, D., Peng, Y., and Peng, X. (2020). “Multivariate Regression-Based Fault Detection and Recovery of UAV Flight Data.” *IEEE Transactions on Instrumentation and Measurement* 69(6):3527–37. doi: 10.1109/TIM.2019.2935576.
- Wang, B., Peng, X., Jiang, M., and Liu, D. (2020). “Real-Time Fault Detection for UAV Based on Model Acceleration Engine.” *IEEE Transactions on Instrumentation and Measurement* 69(12):9505–16. doi: 10.1109/TIM.2020.3001659.

- Wang, K., Zhou, L. and Wang, Y. (2020). "Enhanced Fault Detection Using Deviation Degree Penalty with Stacked Autoencoder in Industry Process." Pp. 1084–89 in *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*.
- Wang, Z., Wang, L., Liang, K., and Tan, Y., (2018). "Enhanced Chiller Fault Detection Using Bayesian Network and Principal Component Analysis." *Applied Thermal Engineering* 141:898–905. doi: 10.1016/j.applthermaleng.2018.06.037.
- Węglarczyk, S. (2018). "Kernel Density Estimation and Its Application." *ITM Web of Conferences* 23:00037. doi: 10.1051/itmconf/20182300037.
- Xiang, G., Tao, R., Peng, Y., Tian, K., and Qu, C. (2020). "Unsupervised Deep Learning for Fault Detection on Spacecraft Using Improved Variational Autoencoder." Pp. 5527–31 in *2020 Chinese Automation Congress (CAC)*.
- Xiao, Y., Shi, H., Wang, B., Tao, Y., Tan, S., and Song, B. (2022). "Weighted Conditional Discriminant Analysis for Unseen Operating Modes Fault Diagnosis in Chemical Processes." *IEEE Transactions on Instrumentation and Measurement* 71:1–14. doi: 10.1109/TIM.2022.3152235.
- Yu, J., Liu, X., and Ye, L. (2021). "Convolutional Long Short-Term Memory Autoencoder-Based Feature Learning for Fault Detection in Industrial Processes." *IEEE Transactions on Instrumentation and Measurement* 70:1–15. doi: 10.1109/TIM.2020.3039614.
- Yu, J., and Yan, X. (2021). "Deep unLSTM Network: Features with Memory Information Extracted from Unlabeled Data and Their Application on Industrial Unsupervised Industrial Fault Detection." *Applied Soft Computing* 108:107382. doi: 10.1016/j.asoc.2021.107382.
- Yu, J., Yoo, J., Jang, J., Park, J. H., and Kim, S., (2018). "A Novel Hybrid of Auto-Associative Kernel Regression and Dynamic Independent Component Analysis for Fault Detection in Nonlinear Multimode Processes." *Journal of Process Control* 68:129–44. doi: 10.1016/j.jprocont.2018.05.004.
- Zhao, B., Cheng, C., Peng, Z., Dong, X., and Meng, G. (2020). "Detecting the Early Damages in Structures With Nonlinear Output Frequency Response Functions and the CNN-LSTM Model." *IEEE Transactions on Instrumentation and Measurement* 69(12):9557–67. doi: 10.1109/TIM.2020.3005113.
- Zhao, J., Li, Y., He, H., and Deng, F. (2020). "One-Step Predictive Encoder - Gaussian Segment Model for Time Series Anomaly Detection." Pp. 1–7 in *2020 International Joint Conference on Neural Networks (IJCNN)*.

## BIOGRAPHIES



**Nur Maisarah Mohd Sobran** is currently a PHD student from Universiti Teknologi Malaysia. In 2009 and 2011, she received his B. Eng and M. Eng. degrees in mechatronics engineering from from Universiti Teknologi Malaysia, Skudai, Johor, Malaysia, respectively. She was appointed as a lecturer at Universiti Teknikal Malaysia Melaka in 2011.

**Zool Hilmi Ismail** obtained his Ph.D. degree in Electrical Engineering from Heriot-Watt University, Edinburgh, United Kingdom in 2011. In 2005 and 2007, he received his B. Eng and M. Eng. degrees in mechatronics engineering from from Universiti Teknologi Malaysia, Skudai, Johor, Malaysia, respectively. He was appointed as a senior lecturer at Universiti Teknologi Malaysia in 2011 and currently a research member of Malaysia-Japan International Institute of Technology and Center for Artificial Intelligence and Robotics at UTM Kuala Lumpur. He was also appointed as a visiting researcher at Kyoto University and Jordan University of Science & Technology in 2014 and 2016, respectively. Zool Hilmi Ismail is a registered professional engineer under the Board of Engineers Malaysia, a member of Society for Underwater Technology, The Institution of Engineering and Technology, Institute of Electrical and Electronics Engineers - Oceanic Engineering Society, Asian Control Association, member and a registered chartered marine engineer of the Institute of Marine Engineering, Science and Technology. His current research interest is in the area of Edge Computing, Model-Predictive Control, Path-Planning and Task Allocation based on Deep-Reinforcement Learning. He has received a number of awards from International RoboCup Competitions (Service Robot Category). In 2017, he was appointed as one of committee member for International RoboCup@Home Education.