# NONLINEAR CHEMICAL PROCESS MONITORING AND FAULT DETECTION BASED ON MODIFIED LSTM MODEL

**By**

**MUHAMMAD RIDZUAN BIN ZAMBRI**

**UNIVERSITI SAINS MALAYSIA**

**2022**

# NONLINEAR CHEMICAL PROCESS MONITORING AND FAULT DETECTION BASED ON MODIFIED LSTM MODEL

**By**

**MUHAMMAD RIDZUAN BIN ZAMBRI**

**Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Chemical Engineering**

**2022**

# ACKNOWLEDGEMENT

First and foremost, all praises and thanks to Allah, the Almighty, for His showers and blessings throughout my journey to complete this thesis successfully. I would like to express my sincere thanks to Dr. Norazwan Md Nor, my final-year project supervisor, for his assistance in generating ideas and encouragement, particularly throughout the drafting of this report. His vision, sincerity, and passion have greatly influenced me, allowing me to successfully complete my project. He helped and guided me on how to complete this assignment and explain the study findings as plainly as feasible.

Next, I want to express my gratitude to all of the respected lecturers in the School of Chemical Engineering who have indirectly assisted me in preparing this report by providing guides. Highly thanks and gratitude to Prof. Dr. Mohd Roslee bin Osman for always giving the best in coordinating this final year project report as the coordinator.

Lastly, I would also like to thank my parents and friends who helped and cooperated with me in successful completion of my report.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| Abbreviation | Description |
|---|---|
| 4IR | Fourth Industrial Revolution |
| AEs | autoencoders |
| ANN | artifical neural netwok |
| BRB | belief rule base |
| CPU | Central Processing Unit |
| DAE | denoising autoencoder |
| DBN | deep belief network |
| DBNs | deep belief networks |
| DCS | Distributed Control Systems |
| DGA | dissolved gas analysis |
| FDD | Fault detection and diagnosis |
| GPU | Graphics Processing Unit |
| IIoT | Industrial Internet of Things |
| LSTM | Long Short Term Memory |
| PCA | principal component analysis |
| PLC | Programmable Logic Controllers |
| PLS | Partial least square |
| RMSE | Root Mean Square Error |
| RNN | recurrent neural networks |
| SCADA | Supervisory Control and Data Acquisition Systems |
| SDAEs | stacked denoising autoencoders |
| SPE | squared prediction error |
| TEP | Tennessee Eastman Process |

**PEMANTAUAN PROSES KIMIA BUKAN LINEAR DAN PENGESAN KESALAHAN BERDASARKAN MODEL LSTM YANG DIUBAHSUAI**

# Abstrak

Dengan perkembangan industri kimia, pengesanan kesalahan proses kimia telah menjadi cabaran yang sukar disebabkan oleh data berdimensi tinggi dan proses kimia yang kompleks serta peningkatan bilangan peralatan. Rangkaian saraf suapan ke hadapan asas tidak begitu berkesan untuk menyelesaikan isu ini. Kertas kerja ini mencadangkan model pengesanan kesalahan berdasarkan model *Long Short Term Memory* (LSTM) yang diubah suai. Percubaan simulasi proses kimia Tennessee Eastman (TE) untuk model LSTM yang diubah suai akan menggunakan perisian MATLAB. Penyiasatan prestasi antara model LSTM dengan *Artificial Neural Network* (ANN). Pengubahsuaian LSTM akan dibuat dengan membandingkan pelbagai jenis kesalahan yang akan digunakan untuk pengesanan kesalahan. Peratusan latihan dan pengesahan juga mempunyai pengaruh yang besar terhadap ketepatan pengesanan kesalahan. Pautan untuk menentukan bilangan optimum nod lapisan tersembunyi dengan memanipulasi nilai setiap lapisan tersembunyi pada rangkaian LSTM ditambah memandangkan bilangan nod lapisan tersembunyi dalam rangkaian LSTM memberi kesan kepada hasil pengesanan. Kemudian, model LSTM yang dioptimumkan akan diperolehi untuk mendapatkan ketepatan pengesanan kesalahan yang lebih tinggi dalam proses kimia. Akhir sekali, melalui simulasi dalam perisian MATLAB, keputusan menunjukkan model LSTM yang diubah suai mempunyai prestasi yang lebih baik dalam pengesanan kerosakan kimia berbanding ANN dan ketepatan yang lebih tinggi yang boleh dicapai oleh model LSTM ialah 99.69%.

**NONLINEAR CHEMICAL PROCESS MONITORING AND FAULT DETECTION BASED ON MODIFIED LSTM MODEL**

# Abstract

With the development of the chemical industry, fault detection of chemical process has become hard challenge due to the high-dimensional data and complex chemical process and increasing number of equipment. The standard feedforward neural network is not particularly effective at solving these issues. This study proposed a fault detection model based on modified Long Short-Term Memory (LSTM) model. The simulation experiment of the Tennessee Eastman (TE) chemical process for modified LSTM model will be using MATLAB software. The investigation the performance between the LSTM model with the Artificial Neural Network (ANN). The modification of the LSTM will be made by comparing different type of faults that will be used for the fault detection. The percentage of the training and validation also has a great influence towards the accuracy of the fault detection. The link to determining the optimum number of hidden layer nodes by manipulated the value of each hidden layers on the LSTM network is added since the number of hidden layer nodes in the LSTM network impacts the diagnosis outcome. Then, the optimized LSTM model will be obtained in order to get higher accuracy of the fault detection in chemical process. Finally, through the simulation in the MATLAB software, the results show that the modified LSTM model has a better performance in chemical fault detection than ANN and the higher accuracy that can be achieved by the LSTM model is 99.69%.

# CHAPTER 1 INTRODUCTION

## 1.1 Research Background

In the era of the Fourth Industrial Revolution (4IR), the industry sector starts to transform factories into smart factories. The objective of smart factories is to meet and overcome the challenges of strong competition, highly customized goods and shorter manufacturing requirements. Furthermore, despite the fact that the chemical industry is continually improving people's lives, chemical accidents continue to occur, causing significant harm to people's lives and property. For example, there were 203 accidents and 238 fatalities in 2017, with human factors being the primary cause of these disasters in China's chemical sector (Aitao and Lingpeng, 2017). In order to minimise negative outcomes, fault detection and diagnosis must enable rapid detection and diagnosis of abnormal conditions (Park *et al.*, 2019). Failures in chemical process should be detected as early as possible to prevent critical damage on the equipment and may cause delays in operations and, consequently, tremendous economic loss (Saufi *et al.*, 2019). As a result, in the chemical sector, fault detection plays an important role. Thus the goal of a smart factories is to make the best use of the facility assets in order to achieve zero-incident and sustainable environmental, health and safety while also increasing the plant's economic operational value. In a smart factory, each asset, from the smallest equipment to a single process unit and collections of processes, not only performs its basic process function, but also provides feedback and predictive information on the current and expected performance of that asset to the plant management system through the use of real-time communication networks (Christofides *et al.*, 2007) .

The industry sector has begun to embrace new innovative technologies in the areas of instrumentation and process monitoring. Key enabling technologies such as the Industrial Internet of Things (IIoT), Cloud Computing, and Deep Learning are now presented in the

industrial processes (Gravanis *et al.*, 2022). The goal of deploying these technologies is to improve their performance, as well as to achieve optimal operation and process control. An important feature of Industry 4.0 that is considered is the quantity of data produced and recorded by various devices that contribute to the monitoring of industrial operations. Multivariate time series data from many correlated signals between sensors and actuators is dealt with by modern industrial control systems (Park *et al.*, 2019). These data gives information that may be turned to useful information about how production lines work. Fault detection is an example of such information since knowing whether a fault has occurred during production as soon as feasible may lead to actions that boost productivity and decrease downtime (Angelopoulos *et al.*, 2020).

Fault diagnosis research began in the 1960s and has continued to attract attention as technology advances. There have been numerous fault diagnosis methods developed, all of which have played an essential role in the petrochemical industry. Fault diagnosis techniques may be classified into the following groups. One of the fault diagnosis methods, such as estimation method and process parameter estimation are to establish linear or nonlinear for fault diagnosis using mathematical models. The second group includes rule-based methods, such as the decision tree algorithm, foil algorithm, and expert system, which use a significant amount of professional knowledge and expertise to identify faults in complex systems. The third group is the data driven method that has two sub-class which are statistic and deep learning (Han *et al.*, 2020).

The fault detection and diagnosis (FDD) framework is based on dynamic neural networks and is designed for nonlinear dynamic processes. The Tennessee Eastman Process (TEP) is used as a benchmark process system for evaluating the proposed framework. A systematic study of several parameters, including as sampling time, feature reduction approaches, and state-of-the-art algorithms for time series classification is performed. The

LSTM provide a robust fault detection and diagnosis framework based on state of the art dynamic deep learning models and time series analysis methods (Gravanis *et al.*, 2022).

## 1.2 Problem Statement

Despite the widespread use of multiple basic logic control systems in chemical processes, such as Distributed Control Systems (DCS), Supervisory Control and Data Acquisition Systems (SCADA), and Programmable Logic Controllers (PLC), abnormal event management still faces several difficulties (Gravanis *et al.*, 2022). The mathematical models are not very suitable for diagnosis of complex processes because the precision of the model has a direct impact on the diagnostic outcome, making diagnosis more challenging. The rules-based classification methods have the difficulty of gaining rule knowledge and conflicting rules (Han *et al.*, 2020). Traditional-model or knowledge-based techniques involving extensive human interaction are becoming too difficult to implement as the quantity of samples and complexity of industrial monitoring data has grown significantly. The complex nonlinear interactions between the signals cause very difficult to detect abnormal measurements. The excessively unbalanced samples of unusual events cause deep learning approaches are unable to tackle an overfitting problems effectively. The interference in the process field and the consequences of a failure might be severe due to robustness and reliability (Christofides *et al.*, 2007).

## 1.3 Objectives

1. To develop LSTM fault detection model using MATLAB software.
2. To diagnose process fault of a highly nonlinear complex chemical process by using deep learning method.
3. To evaluate the performance of the deep learning LSTM model based on fault detection and classification accuracy.
4. To determine the optimum number of hidden layer nodes in the memory-capable LSTM network according to various faults.

# CHAPTER 2 LITERATURE REVIEW

## 2.1 Mathematical models based methods

Form the literature review, the mathematical model that apply in fault analysis is the process parameter estimation method. The weighted least square estimation is used for the process parameter estimation (Liu *et al.*, 2016). Both of this method establish linear or nonlinear models for fault diagnosis using mathematical concepts. Radhakrishnan and Ram, (2001) also applied the mathematical model for predictive control of the bell-less top charging system of a blast furnace. The mathematical model is used to determine the surface profile from a given base line to a specified geometry. Yang et al., (2015) introduced mathematical model of hydro power units under different operating conditions to define the hydraulic-mechanical-electrical coupling system. Jochen and Allg, (2007) used structured augmented state models to diagnose nonlinear systems for single and multiple faults. Fault features are considered to be formed by dynamical exosystems switched through equality restrictions to avoid increased state observability, which limits the number of diagnosable defects. In order to diagnose a fault using mathematical models, the diagnostic method must first construct an accurate mathematical model. However, the model's accuracy has a direct impact on the diagnostic outcome, increasing the complexity of diagnosis and making it unsuitable for the diagnosis of complicated processes.

## 2.2 Rule Classification Based Methods

Decision tree algorithm and expert system are rule-based methods for diagnosing faults in complex systems. A study by Ebert, (1994) apply decision tree for rule-based fuzzy classification. Rule-based fuzzy classification as a basis for building quality models that can detect outlying software components that might cause quality issues for software process quality control. Sugumaran and Ramachandran, (2007) also applied decision tree for fuzzy

classifier in fault diagnosis of roller bearing. The use of a decision tree for automatic rule learning (a collection of 'if-then' rules) and the analysis of the effectiveness of such rules using a fuzzy classifier. He *et al.*, (2018) used belief rule base (BRB) method for fault diagnosis of wireless sensor network to address all kind of uncertain problems. Next, Lin et al., (1993) used an expert system based on the dissolved gas analysis (DGA) technology for the detection and maintenance of suspected transformer failures. The disadvantages of rules-based categorization systems include difficulties in obtaining rule knowledge and conflicting rules.

## 2.3 Data driven method

The data driven method is divided into two categories which are statistics and deep learning. The principal components analysis and partial least square is the statistics data driven method. Jackson, (1991) describe the principal component analysis (PCA) is a widely used statistical method for simultaneously monitoring many variables. It captures the correlations between linear combinations of variables rather than the variables themselves. PCA frequently generates linear combinations of variables that can be used as descriptions or predictions of certain process occurrences. The signal averaging properties of PCA cause these sets of variables are frequently more reliable indicators of process conditions than individual variables (Wise and Gallagher, 1996). According to Newhart *et al.*, (2019) partial least square (PLS) is likely the same as PCA that use $T^2$ and squared prediction error (SPE) statistic to identifies independent linear combination of the measured variables and outliers. PLS distinguishes between input and output variables and conducts dimension reduction individually for each set of variables. PLS is an example of supervised dimension reduction that only monitors output variables that are impacted by input variables while PCA will monitor all the variable in the process simultaneously.

Next, deep learning is another type of data driven method. A study by Yu et al., (2019), the capability of deep learning to handle huge amounts of data and learn high-level

representations by several deep neural networks such as deep belief networks (DBNs), autoencoders (AEs), and stacked denoising autoencoders (SDAEs), have made their way into machine health monitoring and fault detection. Denoising and rebuilding input data from artificial corruption allows AE models, specifically the denoising autoencoder (DAE), to learn useful representations from raw inputs. The deep belief network (DBN) model is a nonparametric process monitoring approach that may be utilized to monitor any process without making any assumptions about distribution. The proposed DBN model's structure is dynamic and can be changed depending on the reference dataset's sample size, which is important in real-world applications with limited and unbalanced reference datasets (Liu *et al.*, 2019). As a result, deep learning algorithms perform better when dealing with nonlinear systems and timing-related defect diagnostics.

## 2.4 The LSTM

The LSTM neural network was created using recurrent neural networks (RNN). A RNN is a neural network model for modelling time series that was first developed in the 1980s. The network's structure is identical to that of a standard multilayer perceptron, with the exception that we enable connections between hidden units with a time delay. The model can retain knowledge about the past due to these connections, allowing it to identify temporal correlations between occurrences that are far apart in the data RNN (Razvan et al.,2013). In contrast to the general artifical neural netwok (ANN), the concept of time is added when RNN is learning. The hidden layer and the output layer are completely connected in traditional neural networks, but the neurons in the hidden layer are not connected such as back propagation neural networks. The neurons in the hidden layer of the RNN contain a feedback mechanism, which forms a closed-loop structure in the RNN's hidden layer. A time series is established when it is stretched to realise the transmission of information before and after. The structure of the RNN is shown in Fig. 2.1.

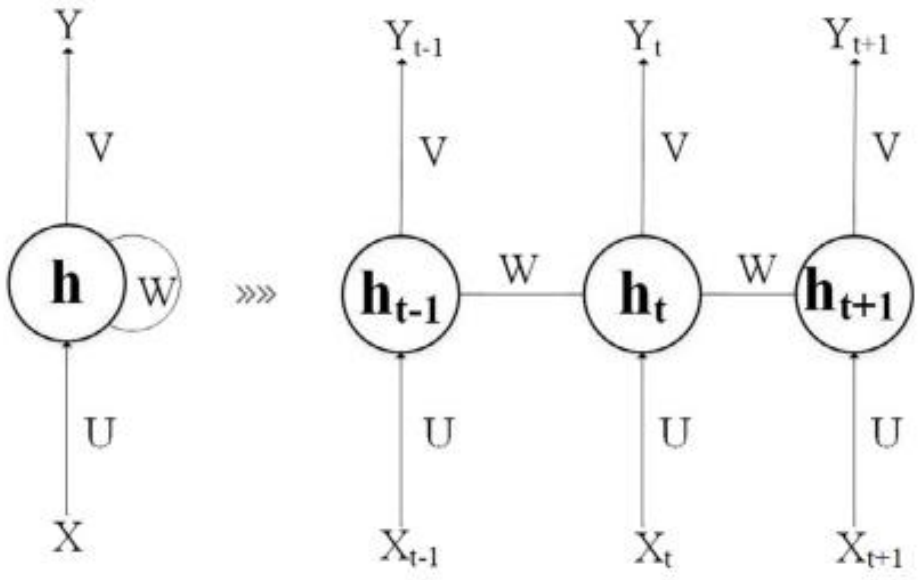


Figure 2. 1 The structure of RNN

In Fig. 2.1, ht denotes for the hidden state of time t, yt for the output of time t, and U denotes for the weight from the input layer to the hidden layer, which abstracts the original input as the hidden layer's input. The memory controller of the network, responsible for memory scheduling, assigns a weight W to hidden layer to hidden layer transmission. f is the activation function and b is the offset. The weight from the hidden layer to the output layer is V, implying that the hidden layer's representation will be abstracted and utilised as the final output. The relationship between them is shown in Eqs.(1) and (2).

$$h_t = f\ (x_t U + \ h_{t-1} W + \ b_h) \tag{1}$$

$$y_t = f\ (h_t W + \ b_y) \tag{2}$$

The RNN network's feature is to load previous information into the current task, as seen in the equation. This enables the neural network to learn and process new information. This enables the neural network to learn and perform tasks involving time series so it can infer the current understanding based on the previous content. However, as the distance between the past knowledge and the present assignment increases, the RNN's capacity to learn decreases.

The RNN's learning capacity will be lost when the distance exceeds a certain value. The LSTM network is presented as a solution to this issue.

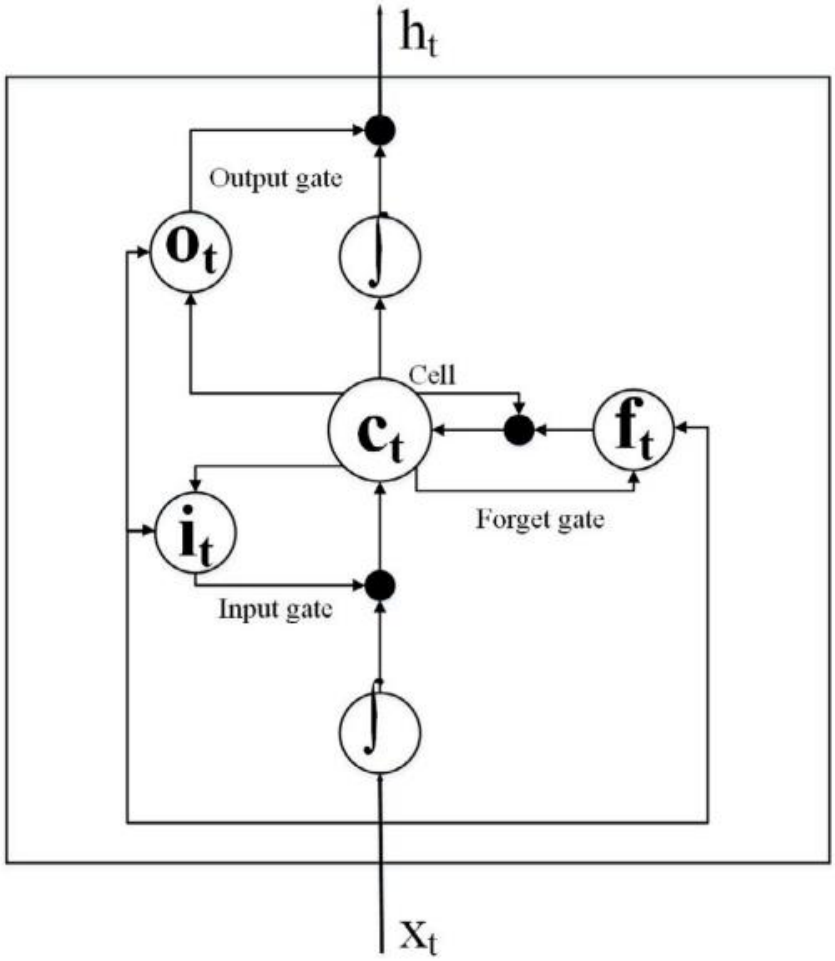


Figure 2. 2 The LSTM structure

In comparison to the typical RNN network, the LSTM network has a new structural cell that aids in the screening of the information learned by the neural network, allowing useful information to remain while rejecting useless information. The LSTM network adds an output cell, which records the current unit's output state, according to the time dimension. The input gate, forget gate, and output gate are the three gates that the cell uses to accomplish this. Figure 2.2 illustrates the specific structure. The forget gate regulates how much of the previous time's unit state $c_{t-1}$ is stored to the present time $c_t$, while the input gate determines how much of the

network's input $x_t$ is saved to the current time's unit state $c_t$, and the output gate control the output from $c_t$ to $h_t$. The TE process is more complex in the time dimension so the update of state layer in the LSTM is improved, resulting in a more balanced information update of the forget gate and input gate in the time dimension. Additionally, the LSTM can learn the fault feature information in the time dimension more effectively while conducting fault diagnosis of the TE process. Table 2.1 shows the definitions of the parameters. The specific method is shown in Eqs. (3) - (8).

Table 2. 1 Parameter meaning table.

| Parameter | Meaning | Parameter | Meaning |
|---|---|---|---|
| σ | Activation function | W | Weight matrix |
| h | Node state of the hidden layer | x | Input |
| b | Bias | tanh | Activation function |
| E | Loss function | δ | Error |

$$f_t = \sigma\left(\mathrm{W_f}.\left[h_{t-1},x_t\right] + b_f\right) \tag{3}$$

$$i_t = \sigma\left(\mathrm{W_i}.\left[h_{t-1},x_t\right] + b_i\right) \tag{4}$$

$$\tilde{C}_t = 1 - \mathrm{f_t} \tag{5}$$

$$C_t = \mathrm{f_t} * \mathrm{C_{t-1}} + \mathrm{i_t} * \tilde{C}_t \tag{6}$$

$$o_t = \sigma\left(\mathrm{W_o}.\left[h_{t-1},x_t\right] + b_o\right) \tag{7}$$

$$h_t = o_t * \tanh(C_t) \tag{8}$$

The weight matrices and offsets of the forget gate, input gate, output gate, and unit state are updated by back propagation error in the reverberation propagation of the LSTM network.

The error at time t − 1 can be calculated as follows

$$\delta_{t-1}^{T} = \frac{\partial E}{\partial h_{t-1}} = \frac{\partial E}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}} = \delta_t^T \frac{\partial h_t}{\partial h_{t-1}} \tag{9}$$

Calculation of weight gradient:

$$\frac{\partial E}{\partial W_{oh}} = \sum_{j=1}^{t} \delta_{o,j} h_{j-1}^{T} \tag{10}$$

$$\frac{\partial E}{\partial W_{fh}} = \sum_{j=1}^{t} \delta_{f,j} h_{j-1}^{T} \tag{11}$$

$$\frac{\partial E}{\partial W_{ih}} = \sum_{j=1}^{t} \delta_{i,j} h_{j-1}^{T} \tag{12}$$

$$\frac{\partial E}{\partial W_{ch}} = \sum_{j=1}^{t} \delta_{\tilde{c},j} h_{j-1}^{T} \tag{13}$$

$$\frac{\partial E}{\partial b_o} = \sum_{j=1}^{t} \delta_{o,j} \tag{14}$$

$$\frac{\partial E}{\partial b_i} = \sum_{j=1}^{t} \delta_{i,j} \tag{15}$$

$$\frac{\partial E}{\partial b_f} = \sum_{j=1}^{t} \delta_{f,j} \tag{16}$$

$$\frac{\partial E}{\partial b_c} = \sum_{j=1}^{t} \delta_{\tilde{c},j} \tag{17}$$

$$\frac{\partial E}{\partial W_{ox}} = \frac{\partial E}{\partial net_{o,t}}\frac{\partial net_{o,t}}{\partial W_{ox}} = \delta_{o,t} X_t^T \tag{18}$$

In formula (18),

$$net_{o,t} = W_{oh}h_{t-1} + W_{ox}X_t + b_o \tag{19}$$

And:

$$\frac{\partial E}{\partial W_{fx}} = \delta_{f,t}X_t^T \tag{20}$$

$$\frac{\partial E}{\partial W_{ix}} = \delta_{i,t}X_t^T \tag{21}$$

$$\frac{\partial E}{\partial W_{cx}} = \delta_{\tilde{c},t}X_t^T \tag{22}$$

# CHAPTER 3 RESEARCH METHODOLOGY

## 3.1 Research Flow

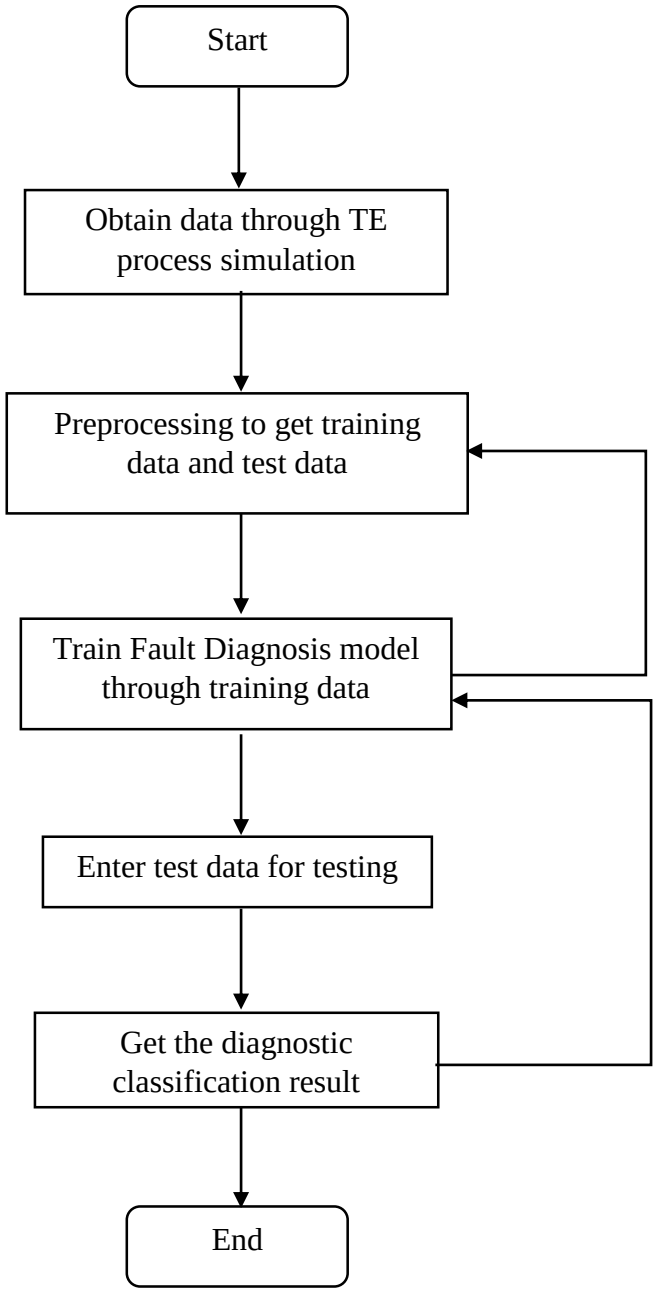


Figure 3. 1 Flow diagram on research project

## 3.2 Data from Tennessee Eastman Process (TEP)

### 3.2.1 Process flowsheet

The Tennessee Eastman Process (TEP) is a simulation model of a real-world industrial process. The simulation model is well suited to process control technology research, but it may also be used to analyze other types of control difficulties. The TEP is shown in Figure 3.2 as a flowsheet. The reactor, product condenser, vapor-liquid separator, recycling compressor, and product stripper are the five major components of the TEP (Yin *et al.*, 2014). The TEP is a closed-loop control mechanism that covers the whole facility. The simulated process may create simulated data at a sampling period of 3 minutes and can replicate normal functioning situations as well as 20 faulty conditions. Two sets of data, training datasets and testing datasets, are generated for each instance (whether normal or faulty). The training datasets are used to build a statistical prediction model, while the testing datasets are used to measure the classifier's accuracy.

The inert B, together with the gaseous reactants A, C, D, and E, is fed into the reactor, which produces the liquid products G and H. In the reactor, there are the following reactions equations (23) – (26).

$$A(g) + C(g) + D(g) \rightarrow G(liq), \quad (23)$$

$$A(g) + C(g) + E(g) \rightarrow H(liq), \quad (24)$$

$$A(g) + E(g) \rightarrow F(liq), \quad (25)$$

$$3D\,(g) \rightarrow 2F(liq). \quad (26)$$

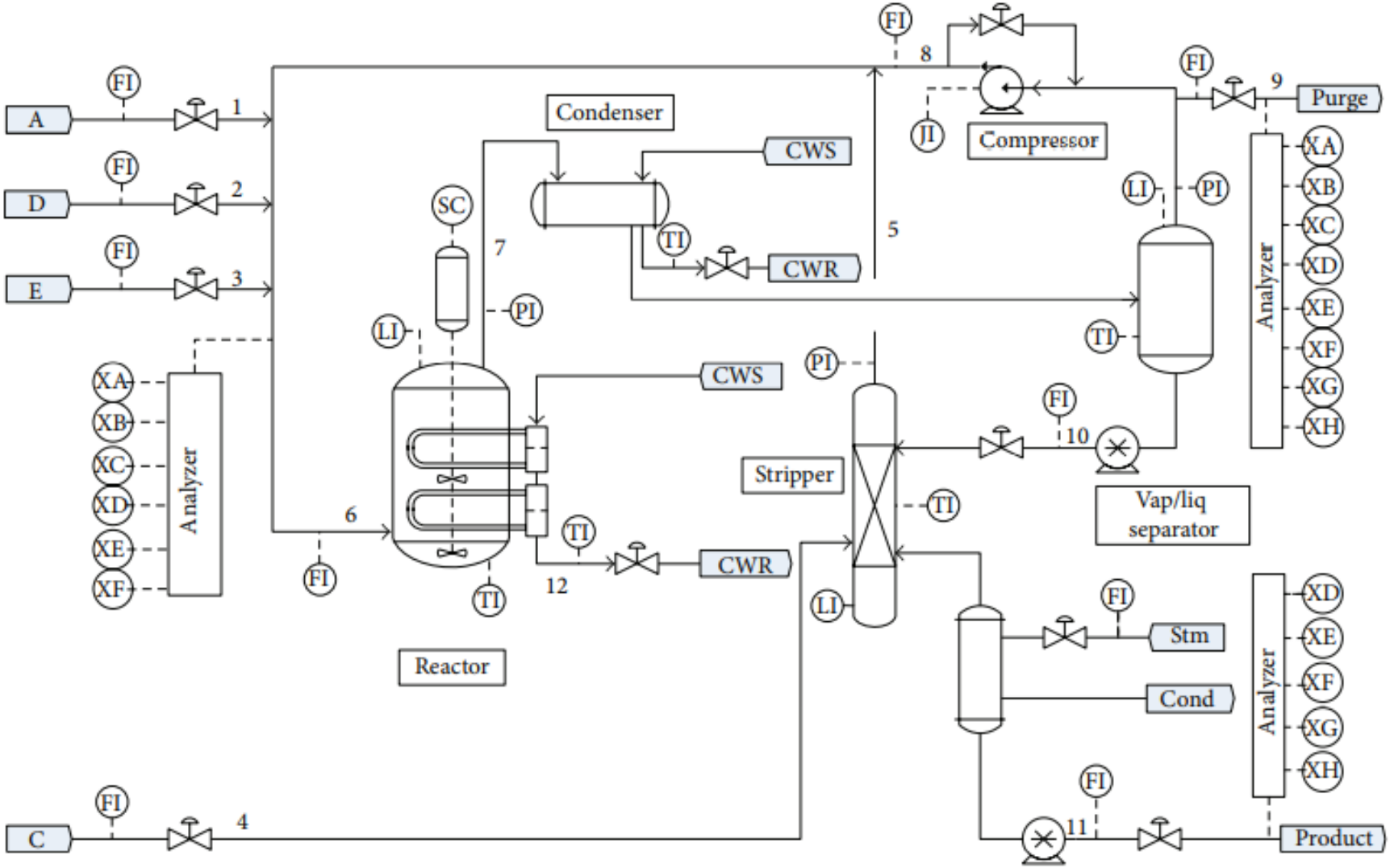


Figure 3. 2 the flowsheet for industrial plant of TE process

The reaction produces the species F as a by-product. In terms of reactant concentrations, the reactions are irreversible, exothermic, and approximately first-order. The reaction rates are Arrhenius functions of temperature, with the G reaction having a larger activation energy than the H reaction, resulting in greater temperature sensitivity. A condenser cools the reactor product stream before it is supplied to a vapor-liquid separator. A compressor recycles the vapour from the separator back into the reactor feed. To prevent inert and byproduct accumulation in the recycling process, a part of the recycle stream is purged. The separator's condensed components (Stream 10) are pumped to a stripper. The remaining reactants from Stream 10 are stripped from Stream 4 and mixed with the recycling stream through Stream 5. The products G and H leaving the stripper's base are routed to a downstream process not shown in the Figure 3.2.

### 3.2.2 Process Variables

There are 41 variables that can be measured and 12 variables that can be manipulated in this procedure. The manipulated variables are listed in Table 3.1. Table 3.2 lists the 22 measured variables, XMEAS (l) through XMEAS (22), that are sampled every 3 minutes. Table 3.3 lists the 19 composition measurements (XMEAS (23) through XMEAS (41). Streams 6, 9, and 11 were used to collect the composition data. Streams 6 and 9 have a sample interval and time delay of 6 minutes, whereas Stream 11 has a sampling period and time delay of 15 minutes. Gaussian noise is present in all process measurements.

Table 3. 1 Manipulated Variables

| Variable | Description |
|---|---|
| XMV(1) | D Feed Flow (Stream 2) |
| XMV(2) | E Feed Flow (Stream 3) |
| XMV(3) | A Feed Flow (Stream 1) |
| XMV(4) | Total Feed Flow (Stream 4) |
| XMV(5) | Compressor Recycle Valve |
| XMV(6) | Purge Valve (Stream 9) |
| XMV(7) | Separator Pot Liquid Flow (Stream 10) |
| XMV(8) | Stripper Liquid Product Flow (Stream 11) |
| XMV(9) | Stripper Steam Valve |
| XMV(10) | Reactor Cooling Water Flow |
| XMV(11) | Condenser Cooling Water Flow |
| XMV(12) | Agitator Speed |

Table 3. 2 Process Measurement (3 minutes sampling interval)

| Variable | Description | Units |
|---|---|---|
| XMEAS (1) | A Feed (Stream 1) | kscmh |
| XMEAS (2) | D Feed (Stream 2) | kg/hr |
| XMEAS (3) | E Feed (Stream 3) | kg/hr |
| XMEAS (4) | Total Feed (Stream 4) | kscmh |
| XMEAS (5) | Recycle Flow (Stream 8) | kscmh |
| XMEAS (6) | Reactor Feed Rate (Stream 6) | kscmh |
| XMEAS (7) | Reactor Pressure | kPa gauge |
| XMEAS (8) | Reactor Level | % |
| XMEAS (9) | Reactor Temperature | °C |
| XMEAS (10) | Purge Rate (Stream 9) | kscmh |
| XMEAS (11) | Product Separation Temperature | °C |
| XMEAS (12) | Product Separation Level | % |
| XMEAS (13) | Product Separation Pressure | kPa gauge |
| XMEAS (14) | Product Separation Underflow (Stream 10) | $m^3$/hr |
| XMEAS (15) | Stripper Level | % |
| XMEAS (16) | Stripper Pressure | kPa gauge |
| XMEAS (17) | Stripper Underflow (Stream 11) | m° /hr |
| XMEAS (18) | Stripper Temperature | °C |
| XMEAS (19) | Stripper Steam Flow | kg/hr |
| XMEAS (20) | Compressor Work | kW |
| XMEAS (21) | Reactor Cooling Water Outlet Temperature | °C |
| XMEAS (22) | Separator Cooling Water Outlet Temperature | °C |

Table 3. 3 Composition measurement

| Variables | Description | Stream | Sampling Interval (min) |
|---|---|---|---|
| XMEAS (23) | Component A | 6 | 6 |
| XMEAS (24) | Component B | 6 | 6 |
| XMEAS (25) | Component C | 6 | 6 |
| XMEAS (26) | Component D | 6 | 6 |
| XMEAS (27) | Component E | 6 | 6 |
| XMEAS (28) | Component F | 6 | 6 |
| XMEAS (29) | Component A | 9 | 6 |
| XMEAS (30) | Component B | 9 | 6 |
| XMEAS (31) | Component C | 9 | 6 |
| XMEAS (32) | Component D | 9 | 6 |
| XMEAS (33) | Component E | 9 | 6 |
| XMEAS (34) | Component F | 9 | 6 |
| XMEAS (35) | Component G | 9 | 6 |
| XMEAS (36) | Componènt H | 9 | 6 |
| XMEAS (37) | Component D | 11 | 15 |
| XMEAS (38) | Component E | 11 | 15 |
| XMEAS (39) | Component F | 11 | 15 |
| XMEAS (40) | Component G | 11 | 15 |
| XMEAS (41) | Component H | 11 | 15 |

### 3.2.3 Process Faults

There are 20 faults preprogrammed in the TEP simulation as tabulated Table 3.4. Sixteen of these deviations are well-known, while the other five are unknown. Fault 1–7 are linked to a step change in a process variable, such as the temperature of the cooling water input or the composition of the feed. The rise in the unpredictability of various process variables is linked to faults 8-12. Fault 13 is caused by a gradual drift in reaction kinetics, whereas fault 14 and 15 are caused by stuck valves. The errors may be simulated singly or in conjunction with one another using the simulation software (Chiang L.H et al., 2001)

The TEP simulator can simulate 22 different conditions, including the normal state and 20 different forms of programmed defects generated by various known process disruptions. All variables will be impacted after the defect is inserted, and some modifications will appear.

Table 3. 4 Process faults

| Variable | Description | 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) | Reactor Cooling Water Inlet Temperature | Step |
| IDV(5) | Condenser Cooling Water Inlet Temperature | Step |
| IDV(6) | A Feed Loss (Stream 1) | Step |
| IDV(7) | C Header Pressure Loss - Reduced Availability (Stream 4) | Step |
| IDV(8) | A, B, C Feed Composition (Stream 4) | Random Variation |
| IDV(9) | D Feed Temperature (Stream 2) | Random Variation |
| IDV(10) | C Feed Temperature (Stream 4) | Random Variation |
| IDV(11) | Reactor Cooling Water Inlet Temperature | Random Variation |
| IDV(12) | Condenser Cooling Water Inlet Temperature | Random Variation |
| IDV(13) | Reaction Kinetics | Slow Drift |
| IDV(14) | Reactor Cooling Water Valve | Sticking |
| IDV(15) | Condenser Cooling Water Valve | Sticking |
| IDV(16) | Unknown | |
| IDV(17) | Unknown | |
| IDV(18) | Unknown | |
| IDV(19) | Unknown | |
| IDV(20) | Unknown | |

### 3.3 Development of LSTM-based Fault Detection Model

Various hidden layer nodes correlate to different diagnostic mistakes when neural networks are trained for different faults (for example, 20 faults in the TE process). The LSTM network based on the original LSTM neural network is optimized in this research by adding a link to the traditional LSTM to identify the ideal number of hidden layer nodes. The testing data is input to get the diagnostic result once the ideal number of hidden layer nodes has been identified. The diagnostic error is greatly decreased after the modification, showing that it has a very excellent impact, according to the results in the experimental section. A vector with a

time length is used as the fault diagnostic model's input. The model produces a two-dimensional array as an output, allowing for defect categorization diagnosis.

The percentage of training data and validation data can be manipulating to maximize the accuracy. The training data is imported into the model once the testing data is collected, and training is carried out after comparing the errors to identify the ideal number of hidden layer nodes. The testing data is loaded into the trained model, and the classification diagnosis result is predicted. The fault diagnosis accuracy rate will increase greatly if the appropriate hidden layer node number is determined. The algorithm fault detection for LSTM using MATLAB can be seen in the appendix A.

### 3.4 Performance Evaluation of LSTM and ANN Model for Fault Detection.

The LSTM and ANN model are the suitable model to perform the fault detection on the nonlinear chemical process monitoring. The data that obtain from the TEP simulation will be used to perform for the fault detection. Although the data from the TEP is complex and nonlinear but both model can handle and detect the fault detection.

There are 2 different performance or assessment measures for both neural networks that are based on accuracy and speed. The evaluation measures based on accuracy include loss. The performance evaluation for LSTM model is based on accuracy. The accuracy is the number of true labels in the test data that match the classifications from classify divided by the number of images in the test data. The equation of the accuracy LSTM model can be seen in equation (27).

$$acc = \frac{sum(Ypred == Ytest).}{numel(Ypred)} \tag{27}$$

Where acc, sum, Ypred, Ytest, and numel respectively the accuracy, summation, output predicition, output test and number of elements. A neural network that can identify the fault type of unseen signals with minimal error is said to have high accuracy. Therefore, the network

is better the greater the accuracy. The performance evaluation that will be used for ANN model is Root Mean Square Error (RMSE). The equation of the RMSE can be seen in the equation (28).

$$RMSE = \sqrt{\left[\frac{1}{n}\sum_{i=1}^{n}(|x_i - y_i|)^2\right]} \quad (28)$$

Where $x_i, y_i$ and n respectively the input value, the measured value, and the total number of samples. The RMSE measures how far the data points are apart from the regression line (da Silva *et al.*, 2022). The value of the RMSE and the percentage of the accuracy that obtain from the MATLAB software will be tabulate for both model. The elapsed time take for both model also will be compare to measure the speed of the fault detection.

The comparison of both model will be compared based on the accuracy of the LSTM model and RMSE for the ANN model. The accuracy for both model will be compared to determine which method will have better accuracy for the fault detection. The elapsed time also will be compare to determine the speed of the fault detection. The shorter elapsed time will show the faster response for the fault detection.

# CHAPTER 4 RESULT AND DISCUSSION

## 4.1 Fault Detection Model Development based on LSTM Model

The proposed LSTM model for fault detection is using MATLAB software. The results of the LSTM model will show the accuracy of the fault detection. The accuracy of the fault detection will determine the effectiveness of the model. LSTM neural networks is used in order to obtained the accuracy of the LSTM model. In the LSTM model, the data needs to separate into two part which are training and validation. The results of training network using LSTM model for the first run for fault detection is shown in the Table 4.1.

Table 4. 1 The summary of training network using LSTM model

| Summary of Training Network using LSTM model | |
|---|---|
| Results | |
| Validation accuracy (%) | 85.18 |
| Training finished | Max epochs completed |
| Training Time | |
| Elapsed | 23min 53 sec |
| Training Cycle | |
| Epoch | 30 of 30 |
| Iteration | 4710 of 4710 |
| Iteration per epoch | 157 |
| Maximum iterations | 4710 |
| Validation | |
| Frequency | 50 iterations |
| Other information | |
| Hardware resource | Single GPU |
| Learning rate schedule | Constant |
| Learning rate | 0.001 |

The summary of the training network shown that the validation accuracy is 85.18%. The training network also completed the max epochs and the training time took 23 minutes and 53 seconds. The epoch and iteration of the training cycle is 30 of 30 and 4710 of 4710

respectively. The iteration per epoch is 157 and the maximum iterations are 4710. The validation of the frequency is 50. The other information that obtain from the training network is computational effort for this iterative calculation is done based on single Graphical Processing Unit (GPU) CUDA-core capacity. GPU has speed up the computational process as compared to the previous method based on the Central Processing Unit (CPU) core capacity as discussed in Chapter 4.2. The learning rate schedule is constant and learning rate at 0.001.

In the Figure 4.1, it shows the graph for training network using LSTM method. Based on this graph, we can see that there are two types of graph which are accuracy graph and loss graph. In the accuracy and loss graph, both graph has three type of lines. In the accuracy graph, the darker blue line shows the smoothed training line, the light blue line show training line and black dotted line with the circle is the validation line. In the loss graph, the darker red line shows the smoothed training line, the light red show training line and black dotted line with the circle is the validation line. The darker line is the smoothed line from the normal training to make the graph look better and nice. When the accuracy graph is increase, the loss graph will be decrease. At the iteration 2000 in both graph, we can see that both graph is already stable and the validation accuracy become more accurate. Thus, both graph is related with each other.

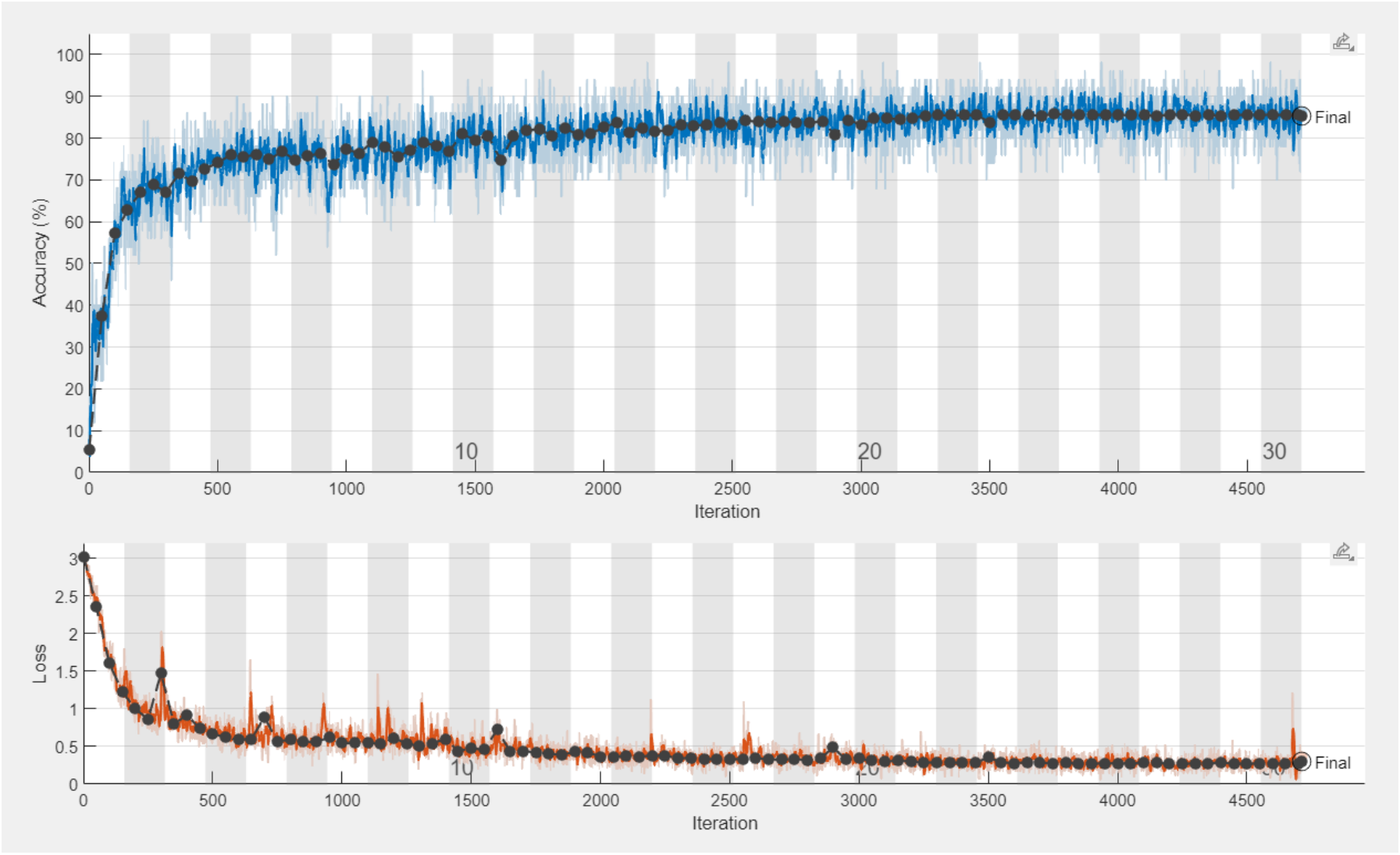


Figure 4. 1 Accuracy graph using LSTM method