# TOXIC GAS DISPERSION MODEL BASED ON NEURAL PATTERN RECOGNITION NETWORKS

**NURFARAH ARINA BINTI ROSLAN**

**UNIVERSITI SAINS MALAYSIA**

**2022**

# TOXIC GAS DISPERSION MODEL BASED ON NEURAL PATTERN RECOGNITION NETWORKS

By

NURFARAH ARINA BINTI ROSLAN

PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR DEGREE OF BACHELOR OF CHEMICAL ENGINEERING

2022

# ACKNOWLEDGEMENT

First and foremost, praises and thanks to Allah, The Almighty for His shower of blessings throughout my research work.

Acknowledgement on behalf of the School of Chemical Engineering, Universiti Sains Malaysia for giving me this opportunity to undergo this course of Final Year Project as one of degree requirements.

High appreciation and sincere gratitude to my supervisor, Dr. Norazwan who made this work possible. His guidance and advice ensure that I can carry this project through all stages of writing for this research. It is really a great opportunity to study under his guidance. Thus, I offer my sincere appreciation for the learning opportunities during this study.

I would also be grateful for the existence of my parents, Roslan Bin Mat Yusof and Norliza Binti Yop Saludin, for their love, prayers and sacrifices for education as well as preparing me for my future. I would like to express my thanks to my brother, Muhd Firdhaus and all relatives for their support and prayers.

Special thanks to my friends as well as classmates who also showing interest during completion of this study.

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATION

| Abbreviation | Description |
|---|---|
| ANN | Artificial Neural Network |
| BR | Bayesian Regularization |
| CFD | Computational Fluid Dynamics |
| CNN | Convolutional Neural Networks |
| DNN | Deep Neural Network |
| LM | Levenberg-Marquardt |
| MSE | Mean Squared Error |
| NPR | Neural Pattern |
| ReLU | Rectified Linear Unit |
| SCG | Scale Conjugate Gradient |
| SDG | Sustainable Development Goal |
| WHO | World Health Organization |

# TABLES OF CONTENTS

**MODEL RAMALAN PENYEBARAN GAS TOKSIK BERDASARKAN RANGKAIAN PENGECAMAN CORAK NEURAL**

# ABSTRAK

Industri kejuruteraan kimia telah berkembang pesat sejak beberapa tahun kebelakangan ini yang menyebabkan banyak kejadian bencana melibatkan industri kimia. Pangkalan data ujikaji Prairie Grass digunakan sebagai data untuk membangunkan model ramalan penyebaran gas toksik berdasarkan jaringan pembelajaran mendalam. Justeru, dalam kajian ini, pembangunan jaringan pembelajaran mendalam dijalankan menggunakan MATLAB. Terdapat 14 parameter yang terdiri daripada 6583 sampel yang berkaitan dengan penyebaran gas toksik daripada eksperimen Prairie Grass digunakan. Untuk mencapai objektif, dua fasa seni bina struktur NPR dijalankan. Pertama, NPR dibangunkan menggunakan tiga algoritma berbeza iaitu Levenberg-Marquart (LM), Regularization Bayesian (BR) dan Scaled Conjugated Gradient (SCG) untuk mencadangkan algoritma rangkaian terbaik menggunakan latihan 70% dan 10-28 neuron tersembunyi. Daripada analisis, BR menunjukkan algoritma rangkaian terbaik berbanding yang lain dengan memberikan nilai R maksimum 0.95. Bedasarkan pemilihan terbaik algoritma rangkaian neural, algoritma BR dilatih lagi menggunakan latihan 50-70% dengan 10-28 neuron tersembunyi. Hasilnya, algoritma BR menggunakan latihan 70% dan 28 neuron tersembunyi memberikan prestasi terbaik dengan nilai R 0.95214. Oleh itu, model NPR ialah model yang boleh dipercayai untuk model penyebaran gas toksik.

**TOXIC GAS DISPERSION MODEL BASED ON NEURAL PATTERN RECOGNITION NETWORKS**

# ABSTRACT

The chemical engineering industry has grown steadily for the past few years that causes many catastrophic incidents involving chemical industries. Prairie Grass experiment database is used as a data to develop toxic gas dispersion prediction model based on deep learning networks. Thus, in this study, development of deep neural network is carried out using MATLAB. There are 14 parameters consist of 6583 samples related to toxic gas dispersion from Prairie Grass experiment is used. To achieve the objectives, two phases of structure architecture of NPR is carried out. First, NPR development is developed using three different algorithms which are Levenberg-Marquart (LM), Bayesian Regularization (BR) and Scaled Conjugated Gradient (SCG) to propose the best network algorithm using 70% training and 10-28 hidden neurons. From the analysis, BR shows the best network algorithm compared to others by giving maximum R-value of 0.95. Following the best selection of neural network algorithm, BR algorithm is further trained using 50-70% training with 10-28 hidden neurons. As a result, BR algorithm using 70% training and 28 hidden neurons give the best performance with R-value of 0.95214. Thus, the NPR model is a reliable model for toxic gas dispersion model.

# CHAPTER 1: INTRODUCTION

## 1.1 Introduction

The chemical engineering industry has been growing steadily which somehow led to concern about the catastrophic accidents regarding chemical processes. Toxic gas leakage accidents may occur in chemical plants due to equipment failure, poor production management, or personnel operation faults (Wang et al., 2020). The accidental release of toxic gas has threatened industrial and social security (Ni et al., 2020). This has become a concern as it can cause environmental pollution, fire explosion, and a series of chain reactions. Hence, rapid and proper aid needs to be done after any accident related to toxic gas dispersion occurs.

The toxic gas dispersion model that is mainly used by the industry is the Gaussian plume method, computational fluid dynamics (CFD) model and integrated model. However, these methods are not suitable to be used during the emergency hour as it takes a long time to simulate. By using these methods, the accuracy of the models also becomes a major concern for emergency measures.

As a solution, the deep learning method has been applied widely in the chemical engineering industry to ensure rapid simulation of toxic gas dispersion using artificial networks. A deep neural network used multiple layers of neurons to solve the complex problems for gas dispersion. However, the availability of the databases becomes a major concern of the models to ensure the accuracy of the model. Thus, the database from the CFD or other integral methods is needed to validate the model. A deep neural network also goes through optimization and activation functions to increase the capability of the model for a high-performance simulation.

In this project, a deep neural network is developed using the experiment database to ensure that the toxic gas dispersion can be simulated for better accuracy and performance. This model also applied multiple type of learning algorithm for neural pattern recognition network to increase the efficiency of the model.

## 1.2 Problem Statement

A catastrophic event which regards to the toxic gas dispersion in the chemical industry has become a major concern (Na, Jeon, and Lee, 2018). Thus, it is very crucial to determine the toxic dispersion model that could meet the accuracy and efficiency required to ensure that the model is effective during emergency times. The use of models such as the Gaussian plume and CFD model does not meet this requirement. Thus, machine learning and deep learning network is introduced for the toxic gas dispersion model. However, the availability of data also becomes a major challenge as the accuracy of the model depends on the database size. This is due to the hazardous nature of the chemical and the limited cost to develop more leak scenarios. Hence, the Prairie Grass experiment is used to evaluate and validate the machine learning and deep learning network that has been programmed.

## 1.3 Research Objectives

The objectives of this research are:

i. To develop toxic gas dispersion model based on neural pattern recognition network.
ii. To evaluate the dispersion model performance based on benchmarked Prairie Grass experiment database.
iii. To improve the performance of the toxic gas dispersion model by determine the number of hidden neurons in NPR architecture.

# CHAPTER 2: LITERATURE REVIEW

## 2.1 Literature Review

Due to the rapid growth of the chemical engineering industry, concern on catastrophic events related to the chemical engineering industry increasing. One of the risks related to the chemical engineering industry is toxic gas leakage. Toxic gas dispersion will generate toxic gas clouds that remain in the accident area that possibly affect the local population if the leakage is above a certain level. Thus, the method used to predict toxic gas dispersion is very essential as it is a very serious health hazard. However, toxic gas dispersion prediction is highly difficult to implement toxic gas dispersion prediction due to cost, risk, and techniques. Each model used also varies in accuracy and prediction time.

To ensure that toxic dispersion can be prevented and controlled rapidly, the toxic gas dispersion model that is instant and accurate is used for these emergency measures. Currently, there are a few types of toxic dispersion prediction and calculation methods that have been introduced: Gaussian plume methods, empirical methods, computational fluid dynamics (CFD) models, and integrated models.

The Gaussian plume method is a simple method that is used to simulate gas dispersion. However, by using this method, accuracy is very difficult to be achieved. For empirical models, one of the most used models is the Pasquill-Gifford and Britter-McQuaid model, which can provide rapid predictions of the downwind plume distances using pre-derived equation and computation graphs (Jiao et al., 2021).

For computational fluid dynamics (CFD) methods, it can obtain a more accurate concentration field, flow field and temperature field as it is based on the Navier-Stoke equation. CFD can conduct a full three-dimensional analysis in an accident with complicated scenarios and complicated geometry (Wang et al., 2020). However, this method is very time-consuming, thus it is not suitable for an immediate response as the simulation should take less than golden time to simulate. As the result, surrogate or meta-models has been conducted to ensure that the complex simulation can be simplified and take a shorter time to simulate. The use of integral methods that incorporate multiple dispersion models such as HEGADIS, NCAR and DRIFT can provide accurate prediction with lower computation sources but with only a limited built-in database.

The deep learning method has been applied widely in the field of the chemical engineering industry as health and safety have become major concerns in this industry for the

prediction of toxic gas dispersion. However, this method is undergoing a major challenge which is the availability of the data. This happens as the accuracy of the model depends mostly on the available database. Thus, a database from CFD or integral model is used for deep neural networks. For example, PHAST-UDM is used as a database for toxic gas dispersion, and it works by validating the neural network model to ensure accuracy. Using these methods, gives a suitable framework to the related scenario-specific parameter and chemical-specific parameter for final development as it can solve the complex problem.

A deep neural network (DNN) is derived from the regular artificial network which has multiple layers that can increase the capability to solve complex problems. The structure of a deep neural network is shown in **Figure 2.1**. The input of the deep neural network are property descriptors and outputs are dispersion distances (maximum dispersion distance, minimum dispersion distance, and maximum dispersion width) (Jiao et al., 2021). This can be programmed using PyTorch deep learning library. In a deep neural network, the activation function is very important as it can affect the efficiency of the learning method. The use of the Sigmoid function and Tanh function are common in safety-related neural network development as they give a very promising result as an activation function (Jiao et al., 2021). Rectified linear unit (ReLU) function is also used as the activation function as it is more computationally efficient as it can introduce non-linearity for backpropagation. For a deep neural network, an optimizer is also important for the optimization of the loss function. Thus, an adaptive moment estimation (Adam) optimizer is used by Jiao et al., (2021) as it has better efficiency for optimization of a loss function. Adam optimizer is performed by combining Momentum optimization and RMSProp algorithm that can develop a high-performance model using a deep learning network for data regression (Na, Jeon, and Lee, 2018).

Convolutional neural network (CNN) is also one of the neural networks used for the gas dispersion that consist of a multilayer composed of neutrons with learnable weights and biased constant (Ni et al., 2020). The structure of CNN is shown in **Figure 2.2**. This method used three-dimensional (3D) neurons which is different from other neural networks. CNN consists of two layers which is feature extraction layer, to extract

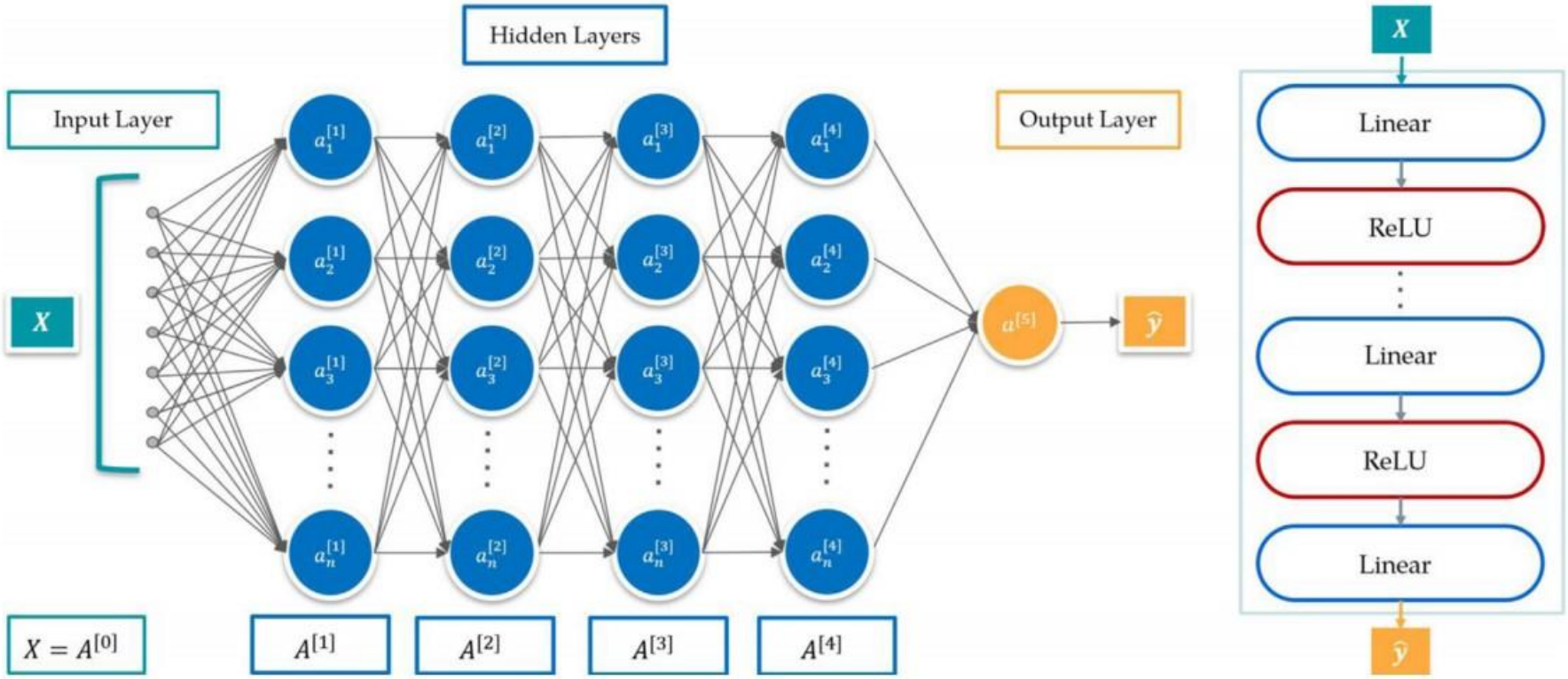


**Figure 2. 1** Main Architecture of Deep Neural Network (Jiao et al., 2021)

local characteristics of each neuron that later determined the position relationship with other characteristics, and the next layer is the feature mapping layer that contains multiple mapping planes. CNN also can use Adam optimizer to adjust and optimize the learning rate by conducting cross-validation. To ensure that the model gives the best performance when the learning rate is 0.01. According to Ni et al., (2020), CNN gives the best performance for the toxic gas dispersion model compared to another model. This model can be programmed using PyCharm using Python.

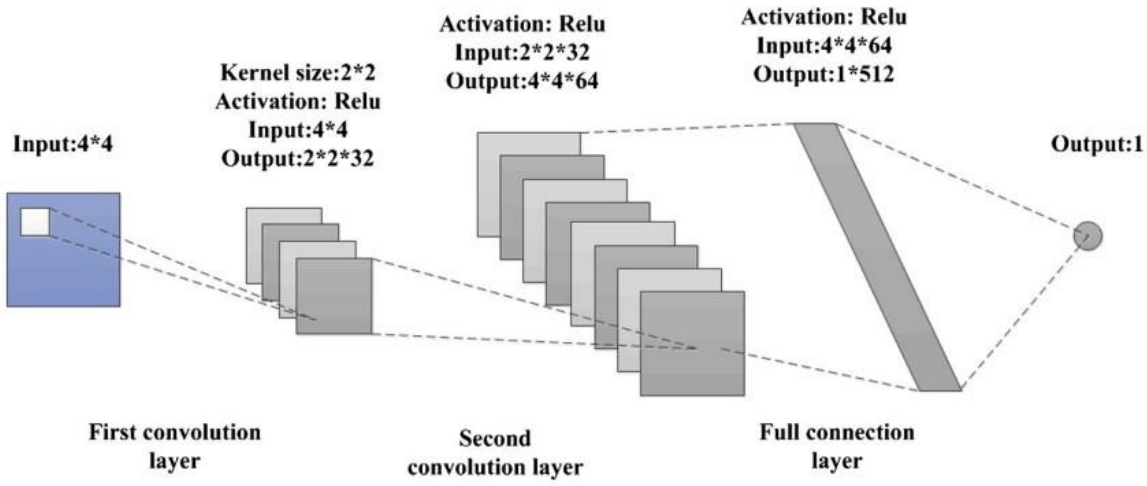


**Figure 2. 2** Detail structure of CNN (Ni et al., 2020)

The autoencoder is also one of the artificial learning systems that can learn and compress input information. This system consists of an encoder (recognition network), a decoder (generative network) and a hidden layer (internal representation) (Na et al., 2018). By using an autoencoder, it can reduce the dimensionality of the data sets even with strong nonlinearity. This method can also be combined with CNN that can ensure the improvement of the model performance as CNN can effectively extract the data even with very complex data.

# CHAPTER 3: METHODOLOGY

## 3.1 Methodology

This chapter shows an overview of the project implementation. This includes the development of a toxic gas dispersion model using benchmarked experimental data, the development of a neural pattern recognition model, and evaluation of the performance as in **Figure 3.1**.

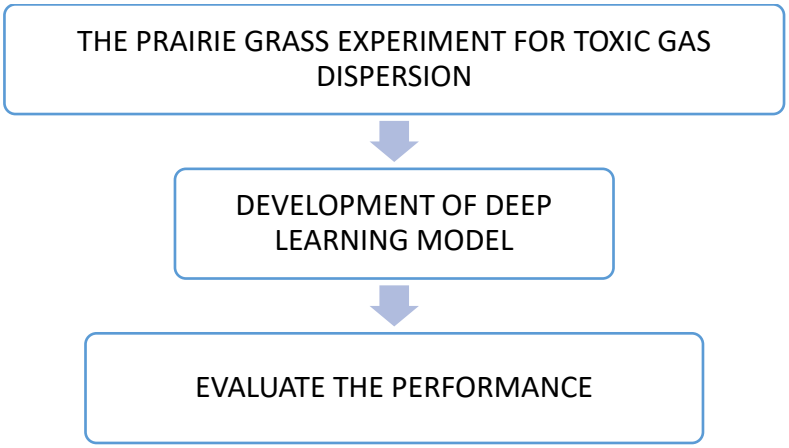


**Figure 3. 1** Methodology of Simulation

## 3.2 Phase 1: The Prairie Grass Experiment for Toxic Gas Dispersion

The database of toxic gas dispersion is retrieved from the Prairies Grass experiment which is a field experiment that referred to a typical hazardous gas emission case with flat terrain and low stack emission (Wang et al., 2018). The experiment is carried out at O'Neil, NE, USA in 1956 using sulphur dioxide that is released from various point source at height of 0.46m without buoyancy for 10 min. The released material was collected at 1.5 m height at five different distance which are 50, 100, 200, 400 and 800m in semi-circular arcs. There are 68 different releases containing tracer data and meteorological data. **Table 3.1** shows different parameters that are used to build deep learning models.

**Table 3.1** Parameter in Prairie Grass experiment

| Parameter | Symbol | Unit | Parameters | Symbol | Unit |
|---|---|---|---|---|---|
| Downwind distance | $D_x$ | $m$ | Average wind speed | $U$ | $ms^{-1}$ |
| Azimuth angle | α | ◦ | Wind direction | θ | ◦ |
| Release rate | $Q$ | $gs^{-1}$ | Friction velocity | $u^*$ | $ms^{-1}$ |
| Source release height | $H$ | m | Temperature | $T$ | ◦C |
| Roughness height | $z_0$ | m | Monin-Obukhov length | $L$ | m |

| | | | | | |
|---|---|---|---|---|---|
| Heat flux | $F_h$ | $W/m^2$ | Mixing height | $z_m$ | m |
| Friction velocity in vertical direction | w* | $ms^{-1}$ | Standard deviation of wind direction | $\sigma_d$ | ° |

## 3.2 Phase 2: Development of Neural Pattern Recognition Network for Gas Dispersion Model

The methodology of this study to develop deep learning model for toxic dispersion is presented in **Figure 3.2** that can be carried out in several steps.

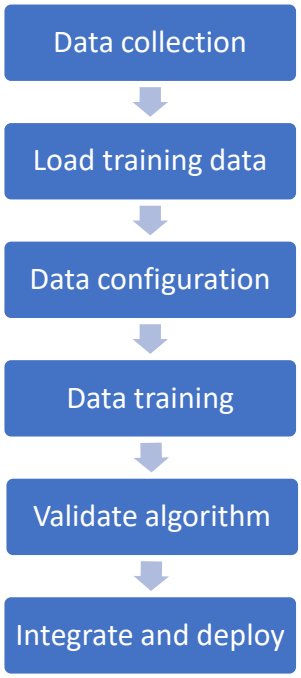


**Figure 3. 2** Development of Deep Learning Model Methodology

During the deep learning model, the simulation of toxic gas dispersion is simulated on MATLAB using Deep Learning Toolbox. Deep Learning Toolbox stores the information that defines the neural network. The input of deep neural networks are property descriptors, and the output of deep neural networks is dispersion distances. During this stage, the number of hidden layers, neurons of each layer, transfer function in each layer, training function, weight learning function and performance layer need to be specified.

After the neural network has been created, the network needs to be configured and trained. Training data is converted to rows as required by MATLAB that need all data to be presented as row vectors. The configuration is done to arrange the network to match the sample data problem. During the configuration steps, steps are consisted of examining input and target

data, setting the network input and output size to match the data, and choosing settings for processing input and output.

After the configuration steps, the training of the network is required. This can be done by performing incremental training or batch training. However, in this case, batch training is used as it is much more efficient than incremental training. During batch training, the weight and biases of the network are updated after all inputs are presented. The data for training can be randomized using the function of "randperm". During the training, the weight is adjusted to ensure that the actual outputs are close to the targeted outputs of the network. Activation function and optimization are introduced to the simulation to increase the efficiency and accuracy of the model.

### 3.3 Phase 3: Performance Evaluation of Toxic Gas Dispersion Model

The simulation that has been simulated will be evaluated to ensure the simulation have a good performance. The performance is evaluated by checking the simulation data with the actual database from ALOHA. This is done using the coefficient of determination ($R^2$), Root Mean Squared (RMSE), and Mean Absolute Error (MSE) for the training and validation of the model. $R^2$ values give the identification of the statistical relationship between measurement and predicted dispersion. The value of $R^2$ ranges from 0 to 1 where 0 means no correlation while 1 shows the perfect correlation. RMSE and MAE have the same units as dispersion that give the indication that a lower value gives better accuracy for the ANN algorithm. The value of $R^2$, RMSE, and MAE can be estimated using followed equations:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(u_i - \widehat{u_i})$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(u_i - \widehat{u_i})^2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(u_i - \widehat{u_i})^2}{\sum_{i=1}^{n}(u_i - \hat{u})^2}$$

Where $u_i$ is defined as measurement value of dispersion, $\widehat{u_i}$ is the predicted dispersion activity by ANN algorithm, $\hat{u}$ is defined mean vale of $u_i$, and $n$ is the number of samples.

# CHAPTER 4: RESULTS AND DISCUSSION

This chapter presents the result obtained based on proposed methodology in Chapter 3 to meet the research objectives. The results are simulated in the MATLAB platform, tabulated and visualised in forms of figures and tables for further analysis.

## 4.1 Development of Neural Pattern Recognition (NPR) Network for Toxic Gas Dispersion Model

The structure of ANN training, validation and testing for neural pattern recognition model is listed in Table 4.1. Three different training algorithms which are Levenberg-Marquardt (LM), Bayesian Regularization (BR) and Scale Conjugate Gradient (SCG) used for analysis. The training algorithm uses the maximum number of epochs which is 1000 if the early stopping is not triggered. The early stopping is usually triggered if the validation shows no improvement or the MSE value of the training set is lower than the goal (Wang. R. et al., 2018). The number of hidden neurons is chosen between 10 to 28 with concentration as the output nodes. Using an appropriate number of hidden neurons in between of the input layer and output layer, the network can perform well with accuracy and convergence speed. The determination of the number of the hidden neuron was done to choose the best training algorithm that can be used for further study. MATLAB deep learning toolbox is used to compute the inputs and target.

**Table 4. 1** Structure of NPR

| | |
|---|---|
| Input | 14 parameters |
| Output/ Target | Concentration |
| No of hidden neurons | 10-28 |
| Training algorithm | Levenberg-Marquardt (LM) |
| | Bayesian Regularization (BR) |
| | Scale Conjugate Gradient (SCG) |
| Training (%) | 70 |
| Validation (%) | 15 |
| Testing (%) | 15 |

By applying the proposed structure according to **Table 4.1**, based on **Figure 4.1**, the SCG algorithm show the lowest range of R value (0.42-0.60), followed by LM (0.72-0.89) and BR give the highest range of R (0.85-0.95). For the MSE value, as in **Figure 4.2,** the BR

algorithm shows the lowest range of MSE (544-1624), followed by LM (991-2865) and SCG (3523-4672). Based on performance of each algorithm, BR algorithm give the best performance using 28 hidden neurons, LM algorithm with 24 hidden neurons while SCG algorithm give the best performance using 28 hidden neurons.

Thus, BR algorithm gives the best overall performance as the R-value approaches to 1 and the lowest MSE value compared to the other two algorithms. This expressed the high accuracy of the algorithm that gives predicted value that is close to the experimental.

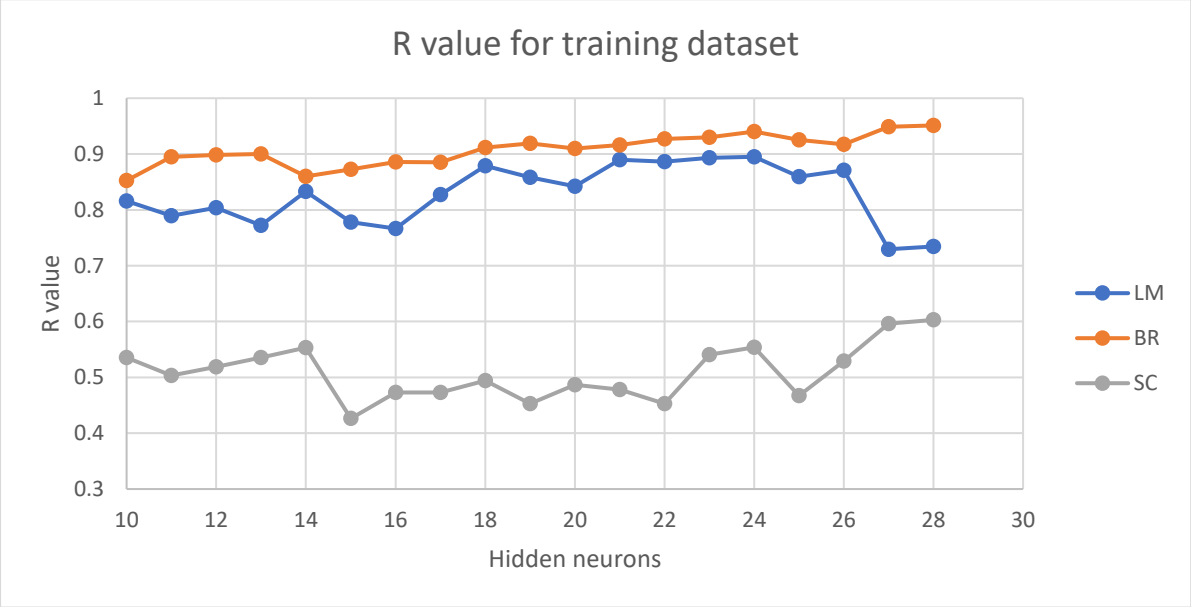


**Figure 4. 1** R values for training dataset

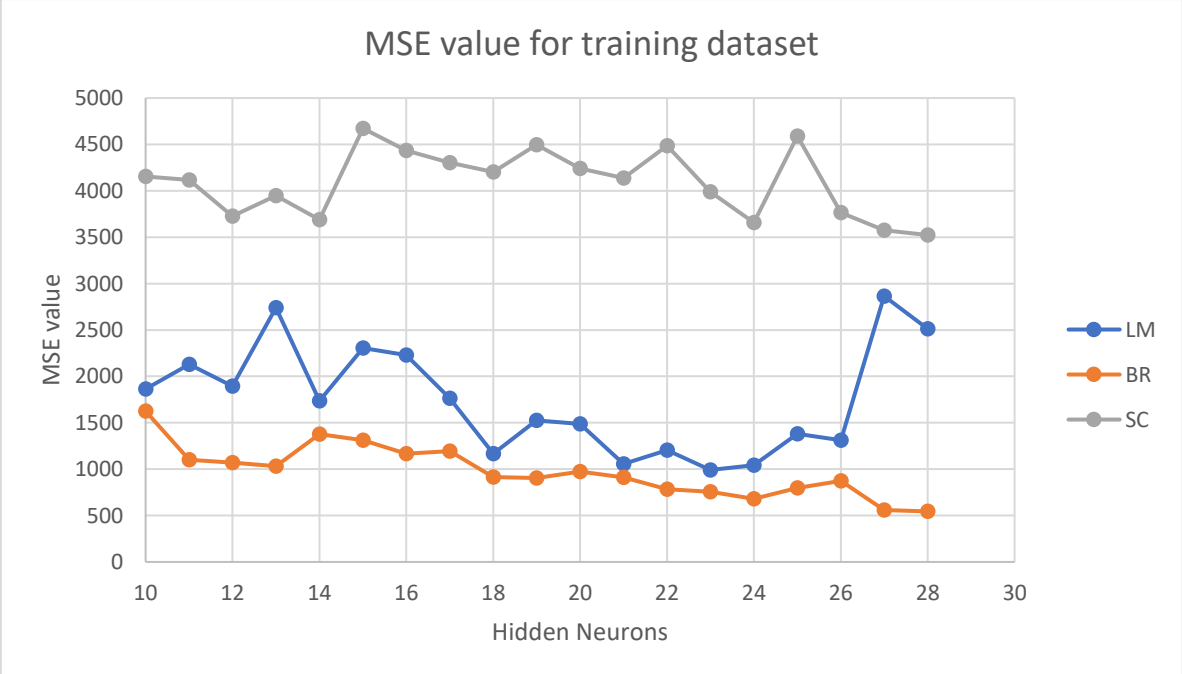


**Figure 4. 2** MSE value for training dataset

From **Figure 4.3** and **Figure 4.4,** LM shows better performance than SCG for the validation dataset. For BR algorithm, it does not require any validation dataset as it is unnecessary. BR algorithm avoids overfitting because the regularization pushes an unnecessary weight towards zero, effectively eliminating them (Burden and Winkler, 2008). Other than that, they are difficult to overtrain as it provides an objective criterion for stopping the training and removes the need for a separate validation set.

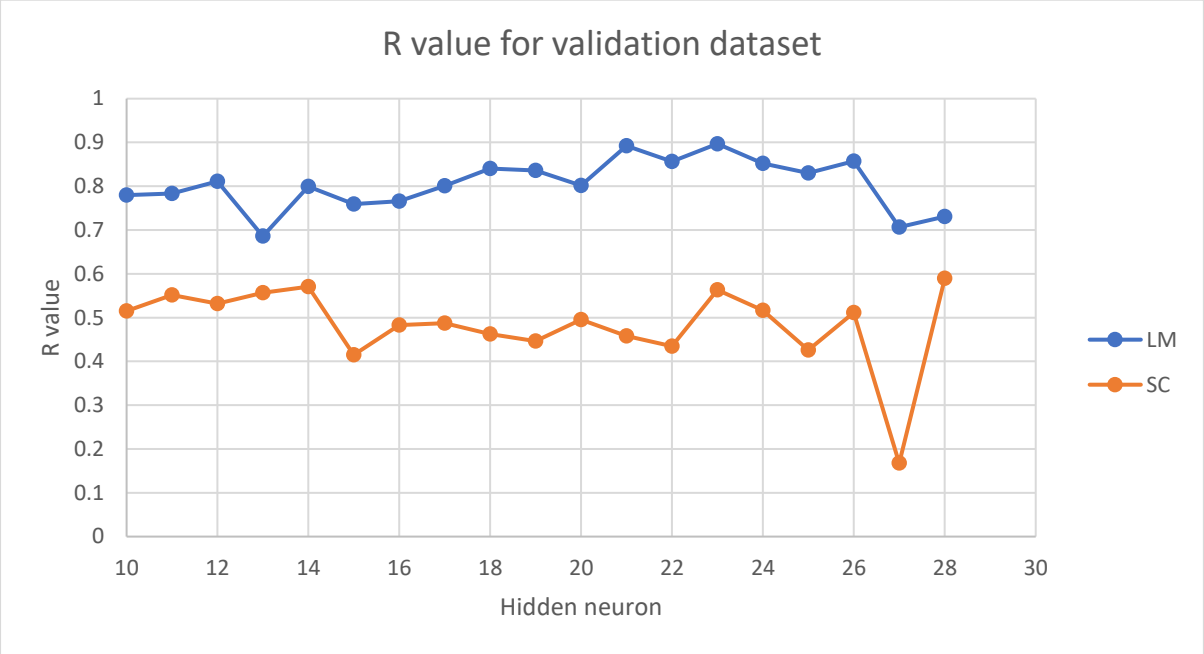


**Figure 4. 3** R value for validation dataset

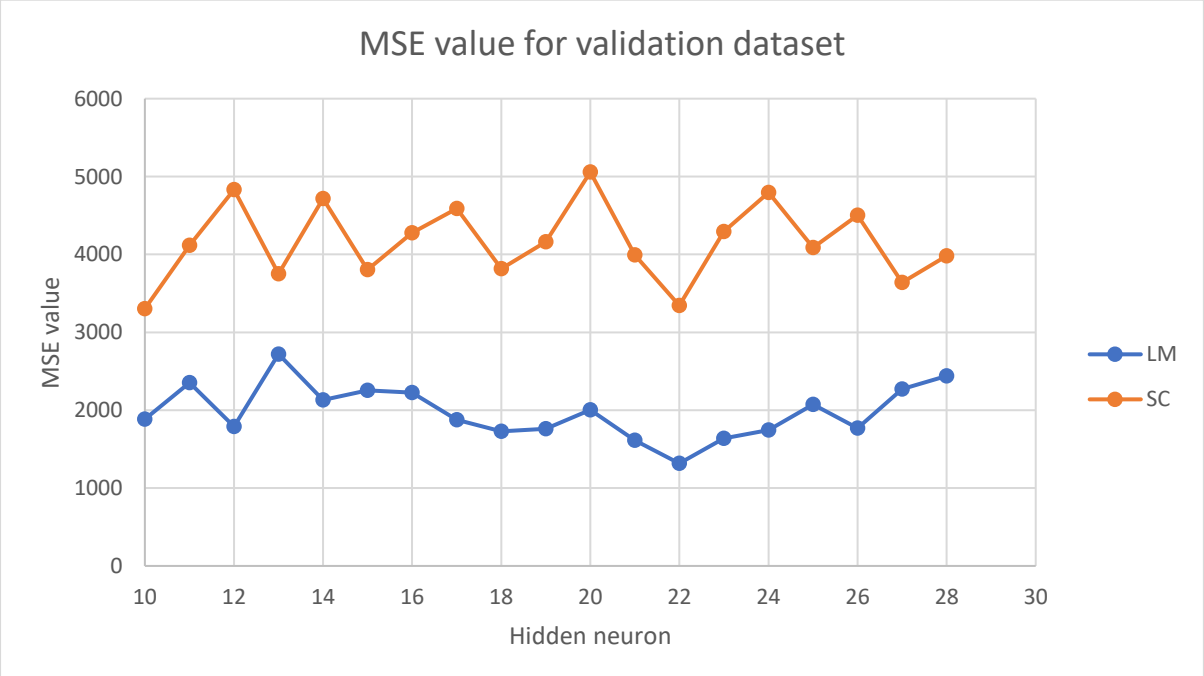


**Figure 4. 4** MSE value for validation dataset

After the model undergo training and validation, it will be analysed for testing dataset. Based on **Figure 4.5** and **Figure 4.6,** BR algorithm give the best performance followed by LM algorithm and lastly SCG algorithm. This can be seen from the higher R value and low MSE value compared to the other two algorithm. In term of number of hidden neurons, BR algorithm and SCG algorithm give the best performance when using 28 hidden neurons, while LM algorithm gives the best performance using 24 hidden neurons. The testing is done to compare the output against the target from the neural network in an independent set. If the testing is done correctly, the neural network can proceed to the deployment phase.

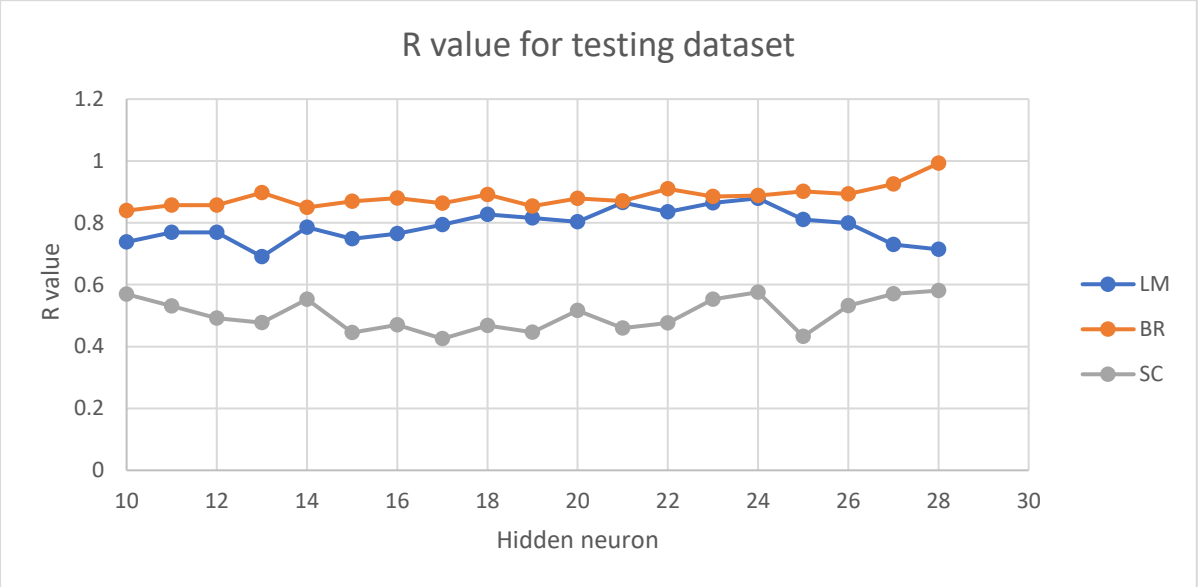


**Figure 4. 5** R value for testing dataset

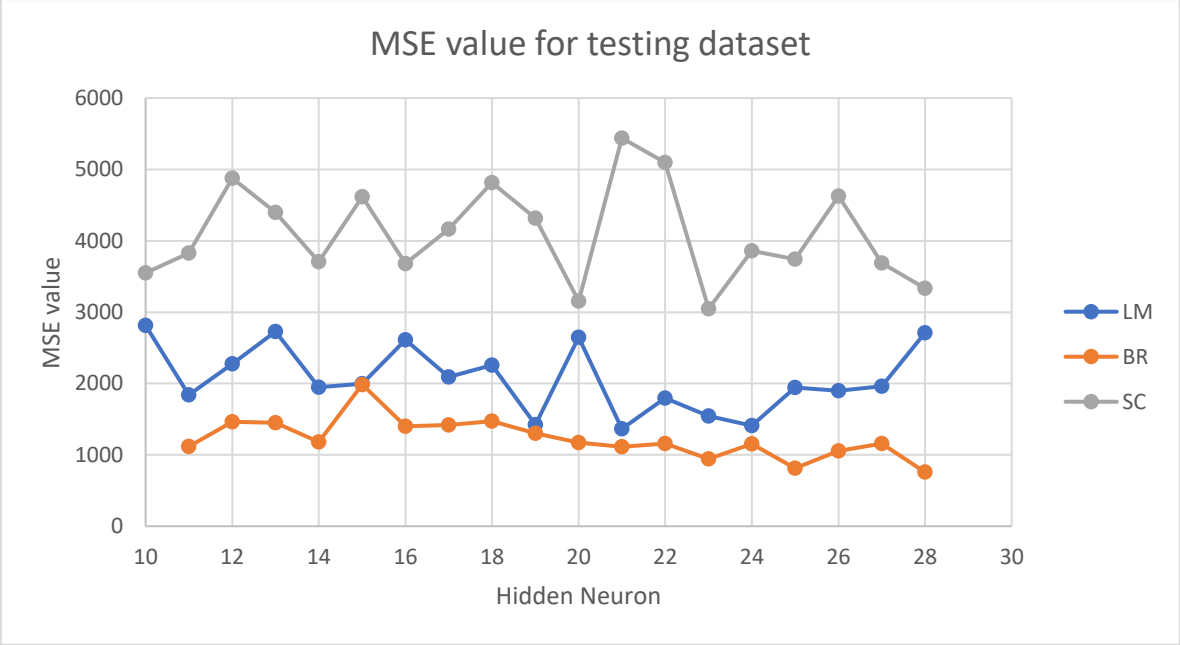


**Figure 4. 6** MSE value for testing dataset

## 4.2 Performance for NPR-based Toxic Gas Dispersion Model with BR Training Algorithm

As BR algorithm is chosen as the best algorithm for the model, **Table 4.2** shows the structure NPR for the optimization for further analysed.

**Table 4. 2** Structure of NPR used for optimization

| | | | |
|---|---|---|---|
| Input | 14 parameters | | |
| Output | concentration | | |
| Training algorithm | Bayesian Regularization (BR) | | |
| No hidden neuron | 10-28 | | |
| Training (%) | 70 | 60 | 50 |
| Validation (%) | 15 | 20 | 25 |
| Testing (%) | 15 | 20 | 25 |

Based on **Figure 4.7** and **Figure 4.8,** R and MSE value is obtained from 50-70% training with hidden neuron of 10-28 using BR algorithm. By comparing the R values with different hidden neuron, the best performance shows when training with 70% with giving the highest R-value of 0.95124, followed by 50% training (0.94993) and lastly 60% training with an R-value of 0.94565. From the analysis, the highest R-value is obtained when 70% training is used with 28 hidden neurons. The performance of 70% training with 28 hidden neurons also shows the lowest MSE value which supports the performance of the training. Thus, 70% training with 28 hidden neurons is chosen as the best structure for PRN analysis.

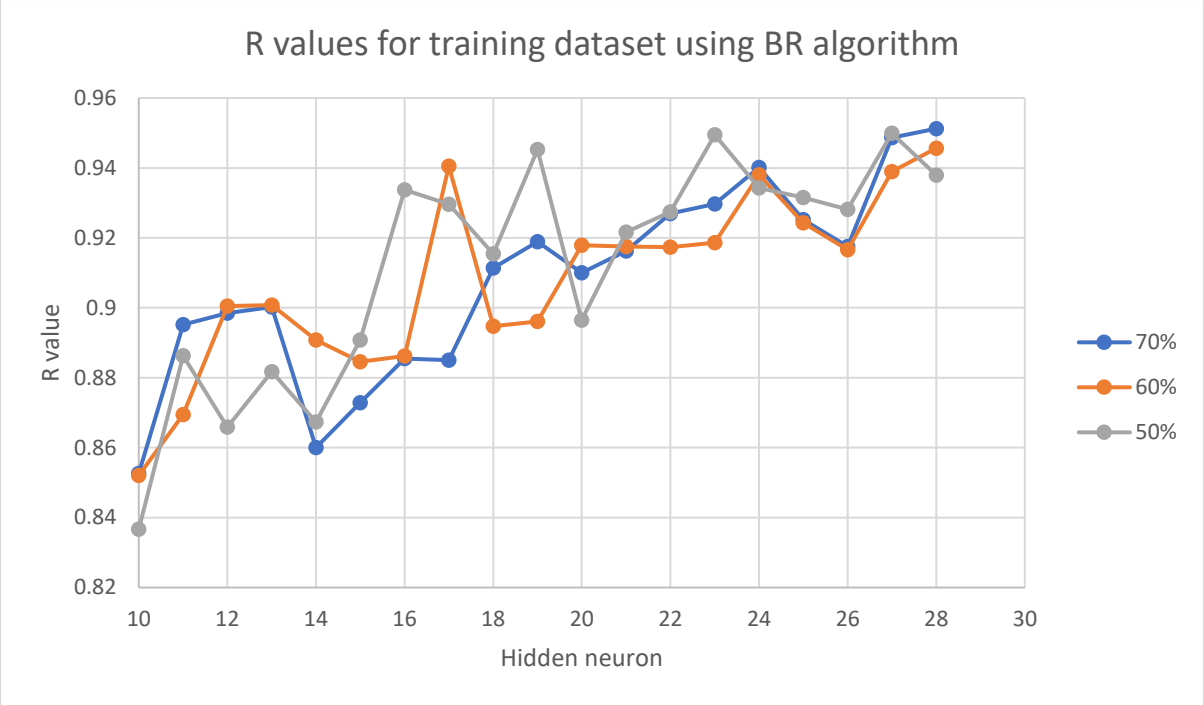


**Figure 4. 7** R values for training dataset using BR algorithm

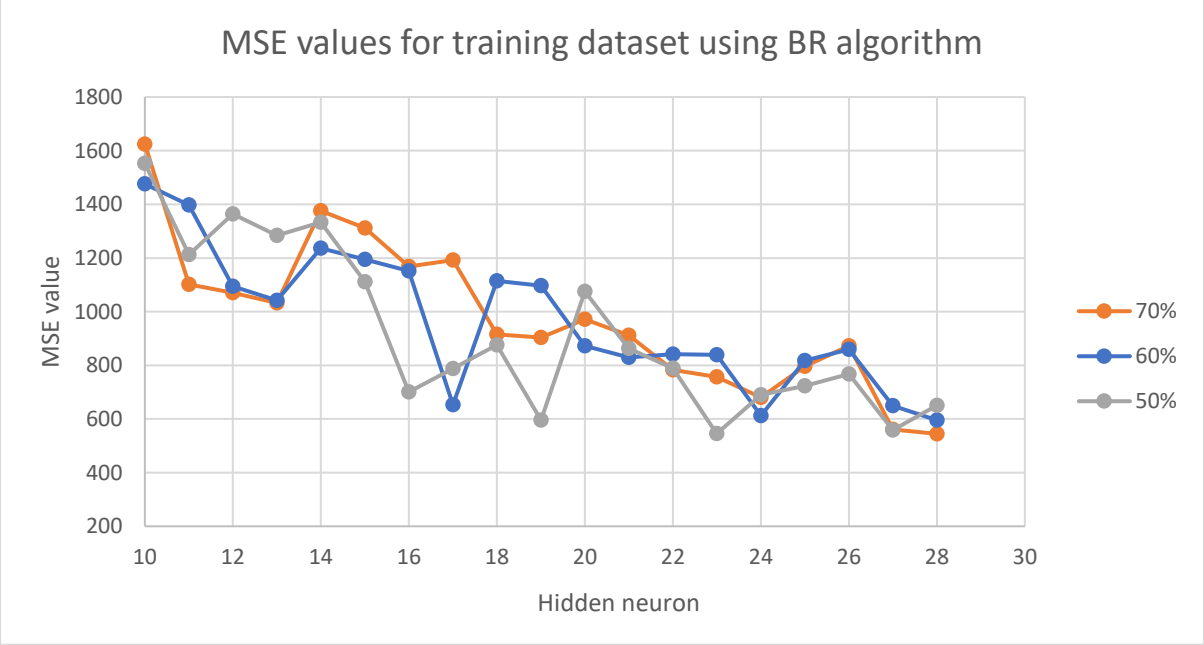


**Figure 4. 8** MSE value for training dataset using BR algorithm

## 4.3 Sustainability

Based on the 2030 Agenda for Sustainable Development that is adopted by all United Nations Member States in 2015, it provides a shared blueprint for peace and prosperity for people and the planet, now into the future. 17 Sustainable Development Goals (SDG) is developed as an urgent call for all countries whether developed or developing in a global partnership. This goals results from a process that is more inclusive than ever, with governments involving business, civil society and citizens. The goals recognized to ensure ending poverty and other deprivations must go with strategies that improve health and education, reduce inequality and spur economic growth while ensuring that climate change can be tackled and working to preserve our ocean and forests. The 17 Goals of SDG involve no poverty, zero hunger, good health and well-being, quality education, gender equality, clean water and sanitation, affordable and clean energy, decent work and economic growth, industry, innovation and infrastructure, reduced inequalities, sustainable cities and communities, responsible consumption and production, climate action, life below water, life on land, peace, justice and strong institutions and lastly partnership for the goals.

As this work involved toxic gas dispersion, SDG 3 which is good health and well-being is relatively close. Exposure to various chemicals occurs every day which is very concerning. According to World Health Organization (WHO) (2021), death due to exposure to hazardous chemicals worldwide increase by 29% in 2019 compared to 2016. This is about 2 million fatalities in 2019 compared to 1.56 million in 2016. Toxic gas dispersion prediction can be a fast alternative to ensure a safe environment if there are any concerns regarding toxic gas dispersion which are hazardous as it can act as an early warning. Thus, the global target to reduce the number of death and illnesses from hazardous chemicals by 2030 can be achieved.

Other than that, SDG 11 which is sustainable cities and communities is relatively close to this research. This goal ensured that cities and human settlements were inclusive, safe, resilient and sustainable. According to the Federation of Malaysian Manufacturers (FMM), 3000 manufacturing and industrial companies of varying sizes have been developed in Malaysia. This rapid development increased the potential of pollution that have the potential of catastrophic events. Thus, the implementation of ANN could aid in ensuring safe and sustainable cities by providing early warning.

# CHAPTER 5: CONCLUSION AND RECOMMENDATION

## 5.1 Conclusion

In conclusion, this study on toxic gas dispersion model that developed using neural pattern recognition network architecture using MATLAB toolbox. This work used 14 parameter of toxic gas dispersion that obtained from Prairies Grass experiment data with total of 6583 dataset that is analysed to achieve the objectives.

The study focused on using NPR model development that involved three different scenarios based on three different algorithm which are LM, BR and SCG algorithm. Random division of 70% training, 15% validation and 15% testing is carried out using 10-28 hidden neurons. Based on the performance of each algorithm, BR algorithm is chosen as the best algorithm as it shows better R and MSE value.

For further optimization, BR algorithm is trained with various number of hidden neurons between 10-28 with 50-70% training. Based on the performance, 70% training with 28 hidden neurons shows the best network structure of NPR model using BR algorithm with R-value of 0.95124.

This summarised that all objectives on development of toxic gas dispersion model using deep learning network were achieved through the study.

## 5.2 Recommendation

The proposed toxic gas dispersion modelling that developed using MATLAB is considered successful as it gives high accuracy in fitting the hazardous gas dispersion database in field cases. However, it can be further improved to ensure that it can also function well in complex terrain environment as terrain of Prairie Grass experiment is considered simple compared to urban area. This model only works for single gas point source emission prediction in certain scenario. Thus, future work using complex terrain environment is considered to enhance the accuracy of the model. Other than that, to ensure the model can operates well with different gas dispersion, it is required to prepared various sample of different kinds of gas or more scenarios related.

# REFERENCES


Burden, F. and Winkler, D. (2008). Bayesian Regularization of Neural Networks. *Methods in Molecular Biology™*, pp.23–42. doi:10.1007/978-1-60327-101-1_3.

Dadkani, P., Noorzai, E., Ghanbari, A. H., & Gharib, A. (2021). Risk analysis of gas leakage in gas pressure reduction station and its consequences: A case study for Zahedan. *Heliyon*, *7*(5). https://doi.org/10.1016/j.heliyon.2021.e06911

Jiao, Z., Ji, C., Sun, Y., Hong, Y., & Wang, Q. (2021). Deep learning based quantitative property-consequence relationship (QPCR) models for toxic dispersion prediction. *Process Safety and Environmental Protection*, *152*, 352–360. https://doi.org/10.1016/j.psep.2021.06.019

Na, J., Jeon, K., & Lee, W. B. (2018). Toxic gas release modeling for real-time analysis using variational autoencoder with convolutional neural networks. *Chemical Engineering Science*, *181*, 68–78. https://doi.org/10.1016/j.ces.2018.02.008

Ni, J., Yang, H., Yao, J., Li, Z., & Qin, P. (2020). Toxic gas dispersion prediction for point source emission using deep learning method. *Human and Ecological Risk Assessment*, *26*(2), 557–570. https://doi.org/10.1080/10807039.2018.1526632

United Nations (2015). Envision2030: 17 goals to transform the world for persons with disabilities | United Nations Enable. [online] Un.org. Available at: https://www.un.org/development/desa/disabilities/envision2030.html.

Wang, J., Yu, X., & Zong, R. (2020). A dynamic approach for evaluating the consequences of toxic gas dispersion in the chemical plants using CFD and evacuation modelling. *Journal of Loss Prevention in the Process Industries*, *65*. https://doi.org/10.1016/j.jlp.2020.104156

Wang, R., Chen, B., Qiu, S., Zhu, Z., Wang, Y., Wang, Y., & Qiu, X. (2018). Comparison of machine learning models for hazardous gas dispersion prediction in field cases. *International Journal of Environmental Research and Public Health*, *15*(7). https://doi.org/10.3390/ijerph15071450

WHO (2021). The public health impact of chemicals: knowns and unknowns. [online] www.who.int. Available at: https://www.who.int/publications-detail-redirect/WHO-FWC-PHE-EPE-16-01 [Accessed 17 Jun. 2022].

# APPENDIX

## Appendix A: Data Tabulation

### Neural Pattern (NPR) Model

| TRAINING ALGORITHM | NUMBER OF HIDDEN NEURON | TRAINING (70%) | | VALIDATION (15%) | | TESTING (15%) | |
|---|---|---|---|---|---|---|---|
| | | R VALUE | MSE VALUES | R VALUE | MSE VALUES | R VALUE | MSE VALUES |
| LM | 28 | 0.734404 | 2511.76194 | 0.7308 | 2440.948 | 0.714598 | 2709.36422 |
| | 27 | 0.729507 | 2865.36983 | 0.706481 | 2269.60945 | 0.730241 | 1960.13121 |
| | 26 | 0.871032 | 1311.3159 | 0.856852 | 1768.50552 | 0.799114 | 1900.71409 |
| | 25 | 0.859651 | 1380.65691 | 0.830061 | 2074.06452 | 0.810132 | 1943.25454 |
| | 24 | 0.895043 | 1041.03094 | 0.852345 | 1745.46519 | 0.879338 | 1407.65525 |
| | 23 | 0.893481 | 991.36373 | 0.896899 | 1636.74352 | 0.864621 | 1546.39935 |
| | 22 | 0.886122 | 1203.78635 | 0.856814 | 1316.43859 | 0.835121 | 1796.6143 |
| | 21 | 0.889599 | 1055.07673 | 0.892511 | 1611.09919 | 0.865377 | 1364.77149 |
| | 20 | 0.842255 | 1489.2675 | 0.80144 | 2004.56261 | 0.802961 | 2648.82246 |
| | 19 | 0.858205 | 1527.24817 | 0.835645 | 1758.99686 | 0.815378 | 1422.43876 |
| | 18 | 0.878839 | 1166.9205 | 0.840743 | 1726.57002 | 0.827148 | 2257.63523 |
| | 17 | 0.827318 | 1763.10862 | 0.801184 | 1875.1168 | 0.79405 | 2091.27814 |
| | 16 | 0.766624 | 2229.42132 | 0.765798 | 2227.38024 | 0.765088 | 2612.29731 |
| | 15 | 0.778066 | 2306.422331 | 0.759368 | 2253.98232 | 0.748875 | 1995.90902 |
| | 14 | 0.83299 | 1734.72548 | 0.79912 | 2130.57846 | 0.786079 | 1949.18873 |
| | 13 | 0.7721276 | 2739.21494 | 0.685828 | 2720.28181 | 0.690603 | 2728.22059 |
| | 12 | 0.803749 | 1894.0467 | 0.811256 | 1788.33941 | 0.769484 | 2274.33213 |
| | 11 | 0.789523 | 2130.41243 | 0.783384 | 2352.56008 | 0.768947 | 1838.26702 |
| | 10 | 0.815847 | 1863.10287 | 0.779837 | 1884.65814 | 0.73769 | 2813.10415 |
| BR | 28 | 0.95124 | 544.54194 | 0 | 0 | 0.99236 | 681.32271 |
| | 27 | 0.948688 | 560.68508 | 0 | 0 | 0.925522 | 757.14227 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 26 | 0.917549 | 872.88816 | 0 | 0 | 0.893651 | 1159.69885 |
| | 25 | 0.925204 | 796.33269 | 0 | 0 | 0.901293 | 1055.90824 |
| | 24 | 0.940173 | 680.75698 | 0 | 0 | 0.887757 | 811.00527 |
| | 23 | 0.929732 | 756.91764 | 0 | 0 | 0.884943 | 1154.22485 |
| | 22 | 0.92692 | 782.68712 | 0 | 0 | 0.910278 | 941.40149 |
| | 21 | 0.916237 | 911.46277 | 0 | 0 | 0.870078 | 1158.74295 |
| | 20 | 0.909978 | 972.00307 | 0 | 0 | 0.878802 | 1112.15753 |
| | 19 | 0.918941 | 903.94527 | 0 | 0 | 0.853783 | 1173.28736 |
| | 18 | 0.911369 | 915.14466 | 0 | 0 | 0.890912 | 1303.51518 |
| | 17 | 0.885051 | 1192.84213 | 0 | 0 | 0.863004 | 1470.37955 |
| | 16 | 0.885524 | 1168.44123 | 0 | 0 | 0.879397 | 1417.15989 |
| | 15 | 0.872812 | 1311.08292 | 0 | 0 | 0.869364 | 1401.18755 |
| | 14 | 0.860046 | 1376.06429 | 0 | 0 | 0.85037 | 1985.78901 |
| | 13 | 0.900248 | 1032.06659 | 0 | 0 | 0.89733 | 1183.10315 |
| | 12 | 0.898504 | 1070.08699 | 0 | 0 | 0.85694 | 1451.76712 |
| | 11 | 0.895213 | 1101.45255 | 0 | 0 | 0.857239 | 1464.51257 |
| | 10 | 0.852611 | 1624.72939 | 0 | 0 | 0.839322 | 1120.31755 |
| SC | 28 | 0.602951 | 3523.19103 | 0.58979 | 3979.41649 | 0.580881 | 3335.43479 |
| | 27 | 0.5959764 | 3575.86231 | 0.16752529 | 3637.8497 | 0.570737 | 3687.64549 |
| | 26 | 0.529445 | 3765.63832 | 0.511551 | 4504.629 | 0.531721 | 4626.01622 |
| | 25 | 0.467196 | 4588.15917 | 0.425589 | 4089.17446 | 0.432802 | 3743.77695 |
| | 24 | 0.55377 | 3659.24981 | 0.51674 | 4794.53043 | 0.575501 | 3860.33357 |
| | 23 | 0.540855 | 3987.54681 | 0.563621 | 4294.77015 | 0.553232 | 3048.28503 |
| | 22 | 0.45263 | 4485.01601 | 0.43441 | 3344.32039 | 0.476434 | 5096.63463 |
| | 21 | 0.478324 | 4135.89244 | 0.458145 | 3993.00058 | 0.460222 | 5439.03654 |
| | 20 | 0.486482 | 4242.204 | 0.495609 | 5056.844 | 0.516151 | 3152.55226 |
| | 19 | 0.452727 | 4495.68524 | 0.446538 | 4162.85913 | 0.446295 | 4315.1379 |
| | 18 | 0.494246 | 4202.07188 | 0.462273 | 3815.08831 | 0.467684 | 4813.06597 |
| | 17 | 0.473249 | 4301.88777 | 0.487129 | 4588.49766 | 0.425152 | 4162.79714 |
| | 16 | 0.473211 | 4433.89292 | 0.483256 | 4276.69497 | 0.470243 | 3680.89734 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 15 | 0.426349 | 4672.62 | 0.414625 | 3802.36417 | 0.445126 | 4617.1382 |
| | 14 | 0.553056 | 3688.01263 | 0.57062 | 4714.85972 | 0.553213 | 3708.4804 |
| | 13 | 0.535538 | 3948.92593 | 0.556794 | 3749.37219 | 0.47763 | 4399.0828 |
| | 12 | 0.518629 | 3725.94747 | 0.531909 | 4831.64927 | 0.491449 | 4879.02921 |
| | 11 | 0.503379 | 4115.4331 | 0.551518 | 4115.87465 | 0.531434 | 3828.75076 |
| | 10 | 0.535582 | 4156.05142 | 0.515263 | 3303.50115 | 0.569198 | 3548.85873 |

**NPR Model with BR Training Algorithm**

| TRAINING ALGORITHM | NUMBER OF HIDDEN NEURON | TRAINING (70%) | | VALIDATION (15%) | | TESTING (15%) | |
|---|---|---|---|---|---|---|---|
| | | R VALUE | MSE VALUES | R VALUE | MSE VALUES | R VALUE | MSE VALUES |
| BR | 28 | 0.95124 | 544.5419 | 0 | 0 | 0.99236 | 681.32271 |
| | 27 | 0.948688 | 560.6851 | 0 | 0 | 0.925522 | 757.14227 |
| | 26 | 0.917549 | 872.8882 | 0 | 0 | 0.893651 | 1159.69885 |
| | 25 | 0.925204 | 796.3327 | 0 | 0 | 0.901293 | 1055.90824 |
| | 24 | 0.940173 | 680.757 | 0 | 0 | 0.887757 | 811.00527 |
| | 23 | 0.929732 | 756.9176 | 0 | 0 | 0.884943 | 1154.22485 |
| | 22 | 0.92692 | 782.6871 | 0 | 0 | 0.910278 | 941.40149 |
| | 21 | 0.916237 | 911.4628 | 0 | 0 | 0.870078 | 1158.74295 |
| | 20 | 0.909978 | 972.0031 | 0 | 0 | 0.878802 | 1112.15753 |
| | 19 | 0.918941 | 903.9453 | 0 | 0 | 0.853783 | 1173.28736 |
| | 18 | 0.911369 | 915.1447 | 0 | 0 | 0.890912 | 1303.51518 |
| | 17 | 0.885051 | 1192.842 | 0 | 0 | 0.863004 | 1470.37955 |
| | 16 | 0.885524 | 1168.441 | 0 | 0 | 0.879397 | 1417.15989 |
| | 15 | 0.872812 | 1311.083 | 0 | 0 | 0.869364 | 1401.18755 |
| | 14 | 0.860046 | 1376.064 | 0 | 0 | 0.85037 | 1985.78901 |
| | 13 | 0.900248 | 1032.067 | 0 | 0 | 0.89733 | 1183.10315 |
| | 12 | 0.898504 | 1070.087 | 0 | 0 | 0.85694 | 1451.76712 |
| | 11 | 0.895213 | 1101.453 | 0 | 0 | 0.857239 | 1464.51257 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 10 | 0.852611 | 1624.729 | 0 | 0 | 0.839322 | 1120.31755 |

| TRAINING ALGORITHM | NUMBER OF HIDDEN NEURON | TRAINING (60%) | | VALIDATION (20%) | | TESTING (20%) | |
|---|---|---|---|---|---|---|---|
| | | R VALUE | MSE VALUES | R VALUE | MSE VALUES | R VALUE | MSE VALUES |
| BR | 28 | 0.945648 | 595.3857 | 0 | 0 | 0.920256 | 797.27412 |
| | 27 | 0.938974 | 649.8984 | 0 | 0 | 0.92432 | 847.41831 |
| | 26 | 0.916631 | 859.7564 | 0 | 0 | 0.887396 | 1312.02921 |
| | 25 | 0.924349 | 817.3155 | 0 | 0 | 0.883692 | 1169.91703 |
| | 24 | 0.938079 | 612.4422 | 0 | 0 | 0.923362 | 1087.08732 |
| | 23 | 0.918623 | 839.62 | 0 | 0 | 0.907067 | 1098.57558 |
| | 22 | 0.917322 | 841.3038 | 0 | 0 | 0.910876 | 1105.52524 |
| | 21 | 0.917533 | 829.8857 | 0 | 0 | 0.893198 | 1360.86259 |
| | 20 | 0.917882 | 873.0229 | 0 | 0 | 0.901934 | 1034.24869 |
| | 19 | 0.896144 | 1096.629 | 0 | 0 | 0.848284 | 1519.3661 |
| | 18 | 0.894709 | 1114.683 | 0 | 0 | 0.866588 | 1349.0879 |
| | 17 | 0.940498 | 653.3598 | 0 | 0 | 0.920255 | 776.17198 |
| | 16 | 0.886237 | 1151.499 | 0 | 0 | 0.86617 | 1554.41226 |
| | 15 | 0.884525 | 1194.189 | 0 | 0 | 0.868814 | 1409.35863 |
| | 14 | 0.89084 | 1236.171 | 0 | 0 | 0.836596 | 1207.38126 |
| | 13 | 0.900737 | 1041.825 | 0 | 0 | 0.862296 | 1441.32939 |
| | 12 | 0.900517 | 1094.114 | 0 | 0 | 0.832713 | 1437.1686 |
| | 11 | 0.869457 | 1396.954 | 0 | 0 | 0.821773 | 1575.84241 |
| | 10 | 0.852075 | 1475.895 | 0 | 0 | 0.836374 | 1856.35651 |

| TRAINING ALGORITHM | NUMBER OF HIDDEN NEURON | TRAINING (50%) | | VALIDATION (25%) | | TESTING (25%) | |
|---|---|---|---|---|---|---|---|
| | | R VALUE | MSE VALUES | R VALUE | MSE VALUES | R VALUE | MSE VALUES |
| BR | 28 | 0.937957 | 650.749 | 0 | 0 | 0.890042 | 1230.60116 |
| | 27 | 0.949933 | 558.5493 | 0 | 0 | 0.918841 | 779.56726 |
| | 26 | 0.928187 | 767.6115 | 0 | 0 | 0.894174 | 1118.2117 |
| | 25 | 0.931528 | 722.9591 | 0 | 0 | 0.894737 | 1150.05314 |
| | 24 | 0.934303 | 690.0156 | 0 | 0 | 0.908696 | 1023.41827 |
| | 23 | 0.949453 | 545.1831 | 0 | 0 | 9.23586 | 819.23473 |
| | 22 | 0.927393 | 789.4129 | 0 | 0 | 0.892965 | 1070.90664 |
| | 21 | 0.921607 | 862.6807 | 0 | 0 | 0.889849 | 1068.74085 |
| | 20 | 0.896479 | 1075.129 | 0 | 0 | 0.890682 | 1185.78661 |
| | 19 | 0.945296 | 595.1601 | 0 | 0 | 0.892619 | 1035.4111 |
| | 18 | 0.91543 | 875.9593 | 0 | 0 | 0.886899 | 1266.48846 |
| | 17 | 0.929578 | 787.5053 | 0 | 0 | 0.875856 | 1139.8997 |
| | 16 | 0.933705 | 701.2292 | 0 | 0 | 0.896168 | 1134.19692 |
| | 15 | 0.890801 | 1111.604 | 0 | 0 | 0.857403 | 1602.13543 |
| | 14 | 0.867314 | 1333.312 | 0 | 0 | 0.838834 | 1791.74227 |
| | 13 | 0.881731 | 1284.225 | 0 | 0 | 0.838442 | 1472.73078 |
| | 12 | 0.865862 | 1363.745 | 0 | 0 | 0.84899 | 16300.6275 |
| | 11 | 0.886299 | 1213.077 | 0 | 0 | 0.853765 | 1416.89195 |
| | 10 | 0.836657 | 1552.376 | 0 | 0 | 0.827685 | 2102.01166 |