Deep Neural Network For Structured Data - A Case Study Of Mortality Rate Prediction Caused By Air Quality

Dian Maharani¹, Hendri Murfi²
Department of Mathematics, Universitas Indonesia, Depok 16424, Indonesia

E-mail:¹sdian.maharani71@sci.ui.ac.id, ²hendri@ui.ac.id

Abstract. The mortality rate is one of the important aspects in determining insurance premiums. The mortality rates have influenced by several factors, i.e., air quality. Therefore, we consider Deep Neural Network (DNN) model for prediction of the air quality-based mortality rate. In this paper, we examine two DNN architectures. The first architecture consists of five layers including an input layer, a hidden layer, two hidden dropout layers, and an output layer. The second architecture consists of four layers including an input layer, a hidden layer, a hidden dropout layer, and an output layer. We optimize dropout rates and activation functions to obtain the optimal accuracies. Our simulations show that the first DNN architecture produces a slightly better performance. The DNN architecture uses ReLu as activation function and applies a 40% dropout rate for both dropout hidden layers. This DNN architecture also gives slightly better accuracy than the standard one hidden layer Neural Networks.

1. Introduction
In insurance, one of an important task is determining the suitable premium rates. The determination of the premium is influenced by several factors, one of which is the mortality rate of a population. The mortality rate is a measure of the number of deaths in a population. In general, mortality rates represent the number of deaths per 100,000 individuals per year. The mortality rate is influenced by Coronary Heart Disease (CHD) that higher in an area with a lower average temperature and hour of sunshine [1]. Besides that, based on decomposition analysis revealed that the mortality rate was higher among males than females and concentrated in age groups 20-49 [2]. It can be concluded that the mortality rate is influenced by several factors, i.e., age, sex, disease, temperature, and an hour of sunshine. Another factor that is also considered by many insurance industries nowadays is climate changes such as air quality. The World Health Organization (WHO) predicts that poor air quality can cause more than one million premature deaths in the world [3].

From the machine learning point of view, the mortality rate prediction is a regression problem [4]. One of the machine learning models that can solve the regression problem is neural networks (NN). The NN model adopts the human nervous system which is a network of many processing units called neurons. NN is an adaptive system that solves problems based on external and internal information that flows through the network. In general, the common structure of the NN model consists of one input layer, one hidden layer, and one output layer. By using supervised learning, we build the NN model that can produce the output that best suits the desired target.

In general, data is characterized by several variables known as features. The suitable features greatly affect the accuracy of a method in data processing. Therefore, feature selection is an important step in data processing including NN. Selecting the right features can reduce the number of features, avoid the use of irrelevant data, and increase the speed of data processing [5]. At present, NN is extended into a
new model that incorporates the process of selecting features as part of the model. The model is known as Deep Neural Network (DNN), and the learning process is also called deep learning. DNN is an NN model that has more than one hidden layer. With many hidden layers and many neurons in each layer, the DNN model becomes more flexible than other models to handle a large number of parameters [6]. The DNN model widely used in the problem of pattern recognition for unstructured data such as sound, image, computer vision and robotics [7].

In this paper, we use the DNN model for the problem of predicting mortality rates based on air quality which has structured data. We examine two DNN architectures. The first architecture consists of five layers including an input layer, a hidden layer, two hidden dropout layer, and an output layer. The second architecture consists of four layers including an input layer, a hidden layer, a hidden dropout layer, and an output layer. We optimize dropout rates and activation functions to obtain optimal accuracies. Our simulations show that the first DNN architecture produces a slightly better performance. Both DNN architectures use ReLu as activation function and apply a 40% dropout rate in each first and second dropout hidden layer. This DNN architecture also gives slightly better accuracy than the standard one hidden layer Neural Networks.

The rest of the paper is organized as follows: In Section 2, the reviews of related works are presented. Section 3 describes the methodology. Section 4 describes the simulation. In section 5, we discuss the results of the simulations. Finally, we give the conclusion in Section 6.

2. Related work

There have been several previous studies that use machine learning models to predict mortality rates, including Sakr, et al. compares several machine learning models such as Decision Tree (DT), Support Vector Machine (SVM), Neural network (NN), Naive Bayes Classifier, Bayesian Network, K-Nearest Neighbor (KNN), and Random Forest to predict all-cause mortality using fitness data: the Hendry Ford Exercise Testing (FIT) project [8]. The results show that various ML techniques can significantly vary regarding its performance for the different evaluation metrics. It is also not necessary that the more complex the ML model, the more prediction accuracy can be achieved. Also, Lee et al. also researching the development and validation of a DNN model for prediction of postoperative in-hospital mortality [9]. The results showed that DNN could predict in-hospital mortality based on automatically extractable intraoperative data.

Some studies that also analyze the performance of NN and DNN include Bianchini et al. compared the complexity of NN with DNN in classification problems [10]. The results showed that based on the number of hidden layers and activation functions, the DNN model was more effective than the NN model. Another study was also carried out by Dalto, et al. applied the DNN model to ultra short-term wind forecasting [11]. The results showed that the DNN with the selection of variables and the determination of the right parameters is better than the NN model for ultra short-term wind forecasting.

The previous above works did not consider both examine DNN model for the problem of predicting mortality rates based on air quality which has structured data and comparing DNN with NN in their works. Therefore, we focus on examine DNN and comparing the performance of the NN and DNN model for the problem of predicting mortality rates based on air quality.

3. Research method

In this study, two models were used to predict mortality rates that were affected by air quality. Both models are NN and DNN.

3.1 Neural Network (NN)

Neural Network (NN) is one of the machine learning models inspired by developments in computational neuroscience, especially information processing in biological NN. The NN model consists of one input layer, one hidden layer, and one output layer.

Mathematically the data processed by the NN model is executed with the following algorithm [4]:

\[ y = f(Wx + b) \]
The data enter (input layer) to neurons $z_j$:

\[ a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{jo}^{(1)} \quad j = 1 \ldots M \]  

(1)

The data output (output layer) from neurons $z_j$:

\[ z_j = h(a_j) \]  

(2)

The data enter (input layer) to neurons $y_k$:

\[ a_k = \sum_{j=1}^{D} w_{kj}^{(1)} z_j + w_{ko}^{(2)} \quad j = 1 \ldots M \]  

(3)

The data output (output layer) from neurons $y_k$:

\[ y_k = l(a_k) \]  

(4)

General form:

\[ y_k(x) = l \left( \sum_{j=1}^{M} w_{kj}^{(2)} h \left( \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{jo}^{(1)} \right) + w_{ko}^{(2)} \right) \]  

(5)

\[ y_k(x) = l \left( \sum_{j=1}^{M} w_{kj}^{(2)} h \left( \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{jo}^{(1)} \right) \right) \]  

(6)

$w_{ji}^{(1)}$ and $w_{kj}^{(2)}$: weight parameters,

$w_{jo}^{(1)}$ and $w_{ko}^{(2)}$: bias parameters,

$h(\cdot), l(\cdot)$: activation functions.

### 3.2 Deep Neural Network (DNN)

Deep Neural Network (DNN) model is the development of the NN where the number of layers in the hidden layer is more than one. DNN model has been widely used to solve problems related to speech recognition [12], pattern recognition, image recognition, and several other DNN applications that continue to be developed. Besides that, DNN is capable of learning high-level features with more complexity and abstract than shallow neural network [7].

In general, the architecture of the DNN in this study, adopts the general architecture used by convolutional NN, where there is a feature extraction layer that uses interchangeably between the convolution layer and max-pooling in the hidden layer, followed by a classification or regression layer such as Multi-Layer Perceptron is based on backpropagation [4].

DNN model that examined in this research consists of two architecture. The first architecture consists of five layers, including of the input layer, three hidden layers that consist of the dense layer with 120 neurons and two layers for dropout, and the last output layer. The second architecture of DNN consists of four layers, including the input layer, two hidden layers that consist of the dense layer with 120 neurons and one layer for dropout, and the last output layer.

The input layer both DNN architecture is suitable with five dimensions. This is because there are five features that affect mortality rates including $O_3$, PM10, PM25, NO2, and temperature. In the first hidden layers, it is consist of 120 neuron setting from the combination of five features. Also, the second and the third hidden layer are the dropout. This dropout layer aims to avoid overfitting or underfitting during model training. While the output layer only consists of one neuron. This is because the problem to be solved is a regression.

In this research, the DNN model used Stochastic Gradient Descent (SGD) as optimizers. SGD is an optimization model that updates parameters for each training data on a model [13]. The SGD method is an easy to use and fast method for problems with very much training data, but it needs manual tuning of optimization parameters such as learning rate that gives the best performance [14]. Srivastava et al. stated that using a high learning rate significantly speeds up the learning process [15]. Based on that, SGD with learning rate $= 0.01$ used in this research. Moreover, DNN model also used default batch-size $= 32$ to examine other parameters such as the rate of dropout and activation function in each hidden layers of DNN model.
The architecture of the DNN model in this study is illustrated in figure 1 and figure 2.

3.3 Network optimization

In the building the DNN models, it is very important to optimize some parameters that used. This is done to reduce prediction errors from the model. These parameters will be described as follows.

3.3.1 Activation functions. The activation function is used in DNN and NN models including linear functions, sigmoid functions, softmax functions, tanh functions, and ReLu functions [4]. The activation function is a function that making the layer active and mapping neurons from the input layer to neurons in the output layer. There are some of the activation function that can use in DNN and NN models.

- The linear activation function is defined by:
  \[ l(x) = x \]  
  (7)
  This activation function in the DNN and NN models used in the output layer. This is because of the problems that will be solved in the research are related to regression problems.

- The Rectified Linear Unit (ReLu) activation function is defined by:
  \[ f(x) = \max(0, x) \]  
  (8)
  where \( x \) is input to neurons.

- The sigmoid activation function is defined by:
  \[ f(x) = \frac{1}{1 + e^{-x}} \]  
  (9)

- The softmax activation function is defined by:
  \[ \sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \text{for } j = 1, ..., K. \]  
  (10)

- The tanh activation function is defined by:
  \[ \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \]  
  (11)
The eLu activation function defined by:

\[ f(x) = \begin{cases} 
  x \alpha(\exp(x) - 1), & x > 0 \\
  x \leq 0 
\end{cases} \quad (12) \]

The hyperparameter \( \alpha \) controls the value to which an eLu saturates for negative network input [16].

### 3.3.2. Dropout

On the DNN model, the number of hidden layers is more than one and it certainly allows DNN to use a large number of parameters and tends to exhibit a highly complex cost function. Besides that, it certainly causes overfitting. Therefore, the dropout technique is used in the DNN model. Dropout is a technique that randomly excludes neurons during the training process. The goal is to reduce the possibility of noisy neurons. Srivastava et al. found that dropout improves the performance of NN in the fields of vision, speech recognition, computational biology, and document classification [15].

### 4. Simulation

In this study, the simulation was carried out four times. These simulations include simulations for pre-processing data, processing, learning process, and final simulations for evaluation of the models. In each simulation also carried out several processes that explained below.

#### 4.1 Pre-processing data

The data used in this study is data mortality rates in several regions in 2007-2012 which are affected by air quality. The air quality that caused the death was reviewed based on the data of ozone (O\(_3\)), PM10, PM25 particles, nitrogen dioxide (NO\(_2\)) levels in the air, and the temperature of several regions. In this study, three steps of data processing were carried out.

##### 4.1.1 Handle missing value.

Based on the data distribution, it is known that there are many data "NaN" on the data features that affect mortality rates. There are several handle missing methods can use to overcome the problem, including FillNaN 0, FillNaN mean, median, most-frequent, and DropNaN methods, and others. In this research, DropNaN method used to handle missing values. The DropNaN is the handle missing value method that has been done by deleting NaN data (scikit-learn.com).

##### 4.1.2 Normalization.

In this study, the second step of pre-processing data is normalizing data. Normalization is the process of scaling data to have a norm unit. Each sample (i.e., each row of the data matrix) with at least one nonzero component is transformed independently from the other sample and causes the norm (l1 or l2) is equal to one (the norm parameter l1 / l2 is used to normalize each nonzero sample).

##### 4.1.3 Splitting data.

The third step of pre-processing data is splitting data. It has done with the aim of dividing the data into training and testing data with the proportion of 80%: 20%.

#### 4.2 Processing data

Processing data in this study used the NN and DNN models. The DNN model is the NN method where the number of hidden layers is more than one layer. The NN model is one of the machine learning models inspired by developments in computational neuroscience, especially information processing in biological NN.

#### 4.3 Learning processes.

The learning process in this study is carried out in the three processes. There are model selection, compile, and training models.

##### 4.3.1 Models selection.

This research used the sequential model in the Keras package to build the NN and DNN models. In this study, the differences between the NN and DNN are the number of hidden
layers and the dropout parameters. The NN model consists of only one hidden layer, and the DNN model consists of more than one hidden layers. Moreover, there is no dropout technique used in NN models.

4.3.2. Compile. In the learning process, the model created will be compiled by applying Stochastic Gradient Descent (SGD) as an optimizer. Other parameters used are loss parameters. A loss in the compile process used to identify errors that occur or error functions of the model. The loss function used in this research is mean squared error (MSE).

4.3.3 Training models. The last learning process is the training model. In this training process will be determined the number of samples that will be updated for every updating the gradient (batch size), the number of iterations on the entire data train (epoch), the distribution of training data to be used as validation of data, verbose determination, initial epoch, and validation step.

4.4 Model evaluation.
In the model evaluation process, the function used is the Mean Square Error (MSE), with calculations:

\[ MSE = \frac{1}{n} \sum_{i=1}^{n} (c_i - \hat{c}_i)^2 \]  

where \( c_i = \) true values; \( \hat{c}_i = \) predict values

In this study, the MSE function used because the problem to be solved is a regression problem. The MSE function can calculate the mean square error, a risk metric that matches the expected value of an error or quadratic loss. The smaller the value of MSE, the better the model used [17].

5. Results and Discussion
In this study, several processes of DNN was carried out. The first step is the process of optimizing the DNN architecture model respectively. After the model of the DNN is optimized, the second step is comparing the DNN and NN model based on the loss value of models to predict the mortality rate caused by air quality.

5.1 Optimization DNN model
There are two parameters optimized on the DNN model. Both parameters are the right dropout rate selection in the hidden layer and the activation function. Other parameters in the DNN model have been determined as described in the previous section. Some optimization processes for the DNN model are explained as follows.

5.1.1 Optimize dropout parameters. Dropout optimization is very necessary for building the DNN model. Dropouts can dramatically improve the convergence and also reduce the testing error [18]. Optimization dropout applied in two architecture DNN models. Mohaimenuzzaman et al. recommended that the number of dropouts is around 20%-50% [19]. Also, Zhang et al. also used a 50% dropout as a baseline configuration of CCN's parameters [20]. In this study, several dropouts rates in the range 20%-50% were performed.

Firstly, the optimization of dropout is applied in the first architecture DNN model whose dropout layers are existed in the second and third hidden layer. The MSE results of several dropouts rates in the range 20%-50% are presented in table 1. Next, the optimization of dropout rates is applied for the second DNN architecture whose a dropout layer is only existed at the second hidden layer. The MSE results of several dropouts rates in the range 20%-50% are presented in table 2. Based on the simulation results in table 1 and table 2, it can be seen that the 40% dropout rate in all dropout hidden layers produces the smallest MSE.
The last step in this research is tested in the first hidden layer for the best activation function to be selected. The results of the simulation presented in Table 3.

### Table 3. The Non-Linear Activation Function in Hidden Layer.

| No | Activation Function | MSE the first DNN architecture | MSE the second DNN architecture |
|----|---------------------|--------------------------------|---------------------------------|
| 1  | ReLu                | 0.08560 ± 0.298 x 10^8         | 0.08598 ± 4.775 x 10^8          |
| 2  | Softmax             | 0.08703 ± 0.016 x 10^8         | 0.08703 ± 0.0348 x 10^8         |
| 3  | Sigmoid             | 0.08738 ± 31.96 x 10^8         | 0.08722 ± 0.0213 x 10^8         |
| 4  | Tanh                | 0.08598 ± 20.75 x 10^8         | 0.08600 ± 0.4015 x 10^8         |
| 5  | Elu                 | 0.08604 ± 1.758x10^8          | 0.08602 ± 14.472x10^8          |

From Table 3, it can be seen that the activation function with ReLu gives the smallest loss value on the DNN model. Therefore, it can be concluded that the ReLu activation function is better than other activation functions. The ReLu activation function is shown to be remarkably adapted to sentiment analysis, a text-based task with a very large degree of data sparsity [20]. Besides that, ReLu activation function is easier to quickly train DNN than sigmoid [21]. Also, the linear activation function was used in the output layer because the problem to be solved is about regression.

### 5.2 The Performance of DNN and NN Model

The last step in this research compared both of architecture DNN model and also NN model. The architecture of the DNN model that used in this part is the first architecture of the DNN model with a

### Table 1. The dropout rate simulations at the first architecture DNN model.

| No | Dropout (2nd layer) | Dropout (3rd layer) | MSE         |
|----|---------------------|---------------------|-------------|
| 1  | 20%                 | 20%                 | 0.08610 ± 2.746 x 10^-8 |
| 2  | 20%                 | 30%                 | 0.08591 ± 1.377 x 10^-8 |
| 3  | 20%                 | 40%                 | 0.08593 ± 1.523 x 10^-8 |
| 4  | 20%                 | 50%                 | 0.08618 ± 5.192 x 10^-8 |
| 5  | 30%                 | 20%                 | 0.08596 ± 1.823 x 10^-8 |
| 6  | 30%                 | 30%                 | 0.08606 ± 2.159 x 10^-8 |
| 7  | 30%                 | 40%                 | 0.08598 ± 2.509 x 10^-8 |
| 8  | 30%                 | 50%                 | 0.08590 ± 8.281 x 10^-8 |
| 9  | 40%                 | 20%                 | 0.08605 ± 3.631 x 10^-8 |
| 10 | 40%                 | 30%                 | 0.08595 ± 3.448 x 10^-8 |
| 11 | 40%                 | 40%                 | 0.08560 ± 0.298 x 10^-8 |
| 12 | 40%                 | 50%                 | 0.08594 ± 0.487 x 10^-8 |
| 13 | 50%                 | 20%                 | 0.08602 ± 2.054 x 10^-8 |
| 14 | 50%                 | 30%                 | 0.08612 ± 4.906x10^-8 |
| 15 | 50%                 | 40%                 | 0.08609 ± 8.033 x 10^-8 |
| 16 | 50%                 | 50%                 | 0.08608 ± 2.600x10^-8 |

### Table 2. The dropout rate simulations at the second architecture DNN model.

| No | Dropout (2nd layer) | MSE         |
|----|---------------------|-------------|
| 1  | 20%                 | 0.08646 ± 37.86 x 10^-8 |
| 2  | 30%                 | 0.08619 ± 17.03 x 10^-8 |
| 3  | 40%                 | 0.08598 ± 4.775 x 10^-8 |
| 4  | 50%                 | 0.08599 ± 4.600 x 10^-8 |

**5.1.2 Optimize activation functions.** This research is tested in the first hidden layer for the best activation function to be selected. The results of the simulation presented in Table 3.
40% dropout rate in each of second and third hidden layers and the second architecture of DNN model with a 40% dropout rate only in the second hidden layer. Besides that, the NN model used consists of three layers, including an input layer, one hidden layer, and an output layer. In the NN model, we use SGD (0.01) as optimizer, batch-size 32, and ReLu activation function. Also, the dropNaN method applied in each model. The simulation results are presented in table 4.

Table 4. The performance of DNN and NN model.

| No | Model                  | MSE         |
|----|------------------------|-------------|
| 1  | The first DNN architecture | 0.08560 ± 0.298 x 10⁻⁸ |
| 2  | The second DNN architecture | 0.08598 ± 4.775 x 10⁻⁸ |
| 3  | NN                     | 0.08621 ± 4.855 x 10⁻⁸ |

Based on the analysis of the lost value in table 4, it can also be seen that the first DNN architecture gives smaller loss value compared with the second DNN architecture and the NN model. It is mean that the more dropout layers on the DNN architecture, the better the performance of DNN model. We guess that it caused by the ability of parameter dropout to reduce the possibility of noisy neurons, help prevent overfitting and increase the performance of DNN model. In addition, the optimization hyperparameter in the DNN model is very important. Moreover, the optimized both of DNN architecture produces a smaller loss value compared with NN. These results are consistent with the previous studies conducted by Bianchini, M et al. [10] and Dalto M, et al. [11], that the DNN model is more effective than NN model for classification problems as well as for regression problems. Although the DNN produces smaller loss value than NN model, the differences of each loss value are small enough. We guess that it caused by the number of features of this problem is small.

6. Conclusion
We use the DNN model for the problem of predicting mortality rates based on air quality. Two DNN architectures are examined for the air quality-based mortality rate prediction. Our simulation shows that the optimization of their architectures and their dropout parameters have affected their accuracies. According to the accuracies of both DNN architecture, we concluded that the first DNN architecture gives better accuracy than the second DNN architecture. Both architectures use 40% dropout rate in all hidden dropout layer. Moreover, the optimized DNN architecture gives slightly better accuracies than the standard one hidden layer NN.

7. Acknowledgments
This work was supported by Universitas Indonesia under PITTA 2018 grant. Any opinions, findings, and conclusions or recommendations are the authors’ and do not necessarily reflect those of the sponsor.

References
[1] Scarborough P, Allender S, Rayner M and Goldacre M 2012 Contribution of Climate and Air Pollution to Variation in Coronary Heart Disease Mortality Rates in England PloS ONE 7 pp 1-6
[2] Chisumpa H V and Odimegwu C W 2018 Decomposition of Age and cause-specific adult mortality contributions to the gender gap in life expectancy from census and survey data in Zambia Population Health 5 pp 218-226
[3] Bahadori M, Sanaeinasab H, Ghanei M, Tavana A M, Ravangard R and Karamali M 2015 Disease Prevention with an Emphasis on the Lifestyle of Military Personnel According to the Social Determinants of Health International Journal of Medical Reviews 2 pp 261-272
[4] Bishop C H 2006 Pattern Recognition and Machine Learning (Berkeley: Springer)
[5] Isabelle G and Elissieff 2003 An Introduction to Variable and Feature Selection Journal of Machine Learning Research 3 pp 1157-82
[6] Hinton G, Deng Y, Yu D, Dahl G, Mohamed A R, Jaitly N, Senior A, Vanhoucke V, Nguyen P,
Sainath T and Kingsbury B 2012 Deep Neural Networks for Acoustic Modeling in Speech Recognition IEEE Signal Processing Magazine pp 82-97

[7] Sze V, Chen Y, Yang T and Emer J S 2017 Efficient Processing of Deep Neural Network: A Tutorial and Survey Proceeding of the IEEE 105 pp 2295-2329

[8] Sakr’ S, Elshawi R, Ahmed A M, Qureshi W T, Brawner C A, Keteyian S J, Blaha M J and Al-Mallah M 2017 Comparison of machine learning technique to predict all-cause mortality using fitness Data: the Henry Ford Excercise Testing (FIT) project BMC Medical Information and Decision Making pp 17-174

[9] Lee C K, Hofer I, Gabel E, Baldi P, Cannesson M 2018 Development and Validation of a Deep Neural Network Model for Prediction of Postoperative In-hospital Mortality The American Society of 129 pp 649-66

[10] Bianchini M and Scarselli F 2014 On the Complexity of shallow and deep neural network classifier: A comparison between shallow and deep architectures IEEE Transactions on Neural Network and Learning Systems 25 pp 1553-1565

[11] Dalto M, Jadranko M AND Vasak M 2015 Deep Neural Network for Ultra short term wind forecasting Proceeding of IEEE pp 1657-1663

[12] Yin S, Liu C, Lin Y, Wang D, Tejedor J, Zheng T F and Li Y 2015 Noisy Training For Deep Neural Network In Speech Recognition EURASIP Journal on Audio, Speech, and Music Processing 2 pp 1-14

[13] Lathuiliere S, Mesejo P, Alameda-Pineda X and Horaud R 2018 A Comprehensive Analysis of Deep Regression [arXiv:1803.08450v1]

[14] Le Q V, Ngiam J, Coates A, Lahiri A, Prochnow B and Ng A Y 2011 On Optimization Methods for Deep Learning ICML’11 Proceedings of the 25th International Conference on Machine Learning pp 265-272

[15] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: A Simple Way to Prevent Neural Networks from Overfitting The Journal of Machine Learning Research 15 pp 1929-1958

[16] Clevert, D-A, Unterthiner T and Hochreiter S 2016 Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) International Conference on Learning Representations [arXiv:1511.07289v5]

[17] Culkin R and Das A R 2017 Machine learning in Finance: The Case of Deep Learning for Option Pricing Journal of Investment Management 15 pp 1-15

[18] Li Z, Gong B and Yang T 2016 Improved Dropout for Shallow and Deep Learning NIPS’16 Proceeding 30th Conference on Neural Information Processing Systems pp 2531-2539

[19] Mohaimenuzzaman Md, Abdallah Z S, Kamrizzaman J and Srinivasan B 2018 Effect of Hyper-Parameter Optimization on the Deep Learning Model Propose for Distributed Attack Detection in Internet of Things Environment [arXiv:1806.07057]

[20] Zhang Y and Wallace B C 2016 A Sensitivity Analysis Of (Practitioners’ Guide To) Convolutional Neural Network For Sentence Classification Proceedings of the 8th International Joint Conference on Natural Language Processing pp 253-263

[21] Glorot X, Bordes A and Bengio Y 2011 Deep Sparse Rectifier Networks Proceedings of the 14th International Conference on Artificial Intelligence and Statistics 15 pp 315–323

[22] Agostinelli F, Hoffman M, Baldi P and Sadowski P 2015 Learning Activation Functions to Improve Deep Neural Networks Accepted as a workshop contribution at ICLR 2015 [arXiv:1412.6830v3]