A study on modeling using big data and deep learning method for failure diagnosis of system

1st Chung-Ki, Seo
Energy ICT Research Team
KDNERI, KEPCO KDN
Jeollanam-do, korea
inha1225@naver.com

2nd Jun-Ha, Kim
Energy ICT Research Team
KDNERI, KEPCO KDN
Jeollanam-do, korea
logicat@naver.com

3rd Soon-Youl, Kwon
Energy ICT Research Team
KDNERI, KEPCO KDN
Jeollanam-do, korea
kslo1968@naver.com

***Abstract*— Power data of the customers are generated in real time and in large quantities according to the activation of the low voltage meter reading service and accumulated and stored in the central FEP, NMS and SMS server. The power data has characteristics of big data and unlike big data of other types, it has the advantage of applying efficient and useful services by applying various analysis and prediction techniques without any additional data processing. Despite these advantages, the field of automated meter reading still focuses on the construction of the system, so data analysis and application aren't actively implemented. However, the auto metering infrastructure is being installed continuously and the amount of data is continuously increasing. Therefore, the importance of utilization of power data based on the auto metering infrastructure becomes an issue, and research on power big data analysis and data mining is actively being carried out. In this paper, we have performed a probabilistic analysis, diagnosis, and prediction of the fault condition of the system through application of the artificial intelligent deep learning algorithm using the system state data stored in real time in the low voltage meter reading system. In this paper, we propose an optimal method for designing and operating a reasonable automated meter reading system with stability and reliability.**

Keywords*— power big data; auto metering infrastructure; artificial intelligent; recurrent neural network; deep learning

I. INTRODUCTION

An automated meter reading (AMR) is an application of the auto metering infrastructure (AMI) that performs low-voltage automated meter reading by wired or wireless communication.[1] Power line communication(PLC) is a wired network communication that uses the power line of the customer as the communication network for the broadband communication and intelligent power network construction and it has a great advantage over the other broadband communication method in that it can perform the broadband communication using the existing power line without installing additional communication network Since it has, it is scheduled to install 18 million in the future including 2 million in 2015, starting with 500,000 in 2010. Data concentration unit (DCU), which is the core equipment of remote meter reading, generates various kinds of power data in units of 15 minutes and transmits them to a remote server. The DCU is a 365-day-a-year system that produces a huge amount of power data and exponentially growing data. This data has the nature of big data that does not require processing. Therefore, we are trying to apply various services through big data analysis and analysis. In this paper, we try to predict and predict the system failure rate using the system state information data among the data generated in this way. An artificial intelligence algorithm applying a deep learning technique using a Tensor-flow engine was applied to the learning and prediction of the failure rate.

II. THEORY

A. System Configuration

The simulation to be implemented in this paper is to be applied to a system used for low voltage automated meter reading. The AMR system consists of a front end processor (FEP) server, a DCU, a slave PLC modem, and a digital power meter. At this time, the slave PLC modem may simultaneously perform the functions of the repeater modem for routing. Fig. 1 shows the schematic configuration of a PLC-based AMR system. In the figure, the FEP server and the DCU are ethernet-based communication sections. DCU and slave PLC modem is a communication section connected to a PLC capable of bi-directional communication and has an advantage of being able to service using existing power line without installing a separate wired and wireless communication network. However, the PLC section has a disadvantage in that it can't guarantee an appropriate quality of service (QoS) continuously because it is very susceptible to the influence of the noise generated from the old power line and the household appliance of the consumer. Since the system is located in the outer pole, it is inevitably subject to external environmental influences (lightning, temperature, humidity, etc.). Nevertheless, the system should continue to operate for 365 days in terms of reliability and stability of automated meter reading. Therefore, in this paper, we tried to activate the inspection service by ensuring the stability and reliability of the system by minimizing the failure of the system and automatically performing quick recovery even in case of failure. The fault diagnosis system that we want to implement is a PC simulator that uses system state big data that is stored in real time on a system that is running in the field. For the implementation of the simulator, we first performed the model fitting work for the fault diagnosis and the prediction for the learning using the neural network algorithm. We construct a model that includes recurrent neural network (RNN) algorithm considering each input data is closely related to other input data. The simulator consists of four parts as follows. The input part normalizes the state data of the system and processes it into a file form. The learning part learns and processes the input data. The processing part analyzes and diagnoses the failure probability of the system based on learning. Finally, the output part displays the result and visualizes it

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE
978-1-5386-5035-6/18/$31.00 ©2018 IEEE

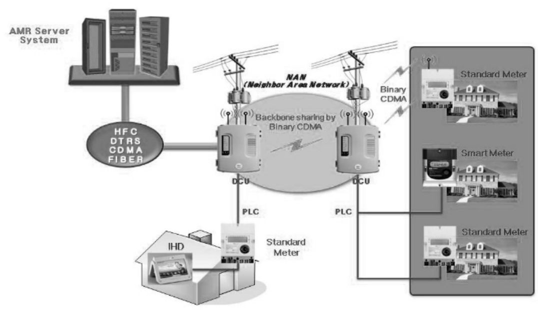

Fig. 1. Configuration diagram of PLC low voltage automated meter reading system [2]

B. Algorithm

We used deep learning method which applied neural network algorithm to simulation for system fault diagnosis and prediction. In general, the neural network algorithm consists of an input layer, a hidden layer, and an output layer as shown in Fig. 2.

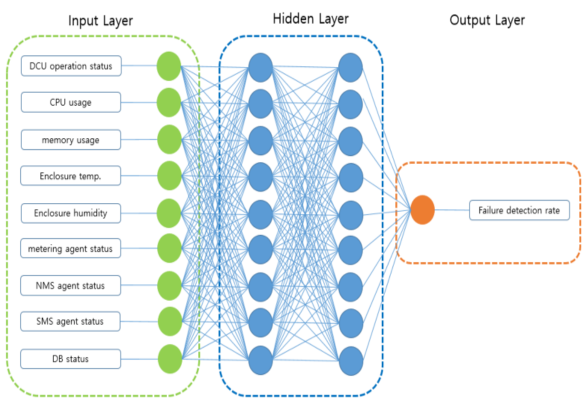

Fig. 2. General neural network configuration

RNN is optimized for continuous data processing such as statistical data of time series and exhibits good performance. That is, the system has an excellent function for predicting the next state with the present data. Since RNN has a recursive structure with a hidden state node with a directional edge in the structure as shown in Fig. 3, it computes the state after the state is calculated, and the previous data affects the next. [3]

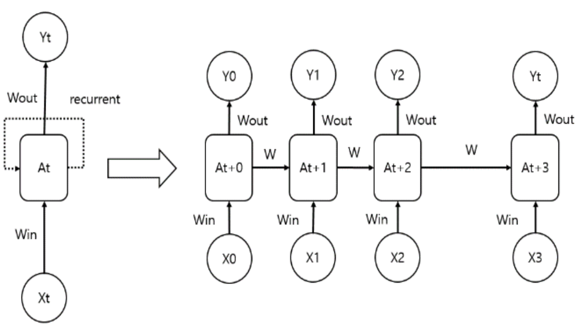

Fig. 3. RNN sequence

The RNN model can be expressed as follows.

$$h_s = f_w (h_{s-1}, x_s) \qquad (1)$$

Here, h_s = final state, f_w = some function with parameters W, h_{s-1} = old state, x_s = input vector at some state step, RNN calculates and outputs the state. In order to learn the model, a basic cell must be constructed. In the tensor flow, various basic cell configuration support functions are supported. We constructed a cell with long short term memory (LSTM) method, which is widely used because of its high accuracy. LSTM is a special cyclic neural network first considered by Hochreiter and Schmidhuber in 1997. LSTM is a network composed of interconnected cells. Each LSTM block contains three types of gates: input gates, output gates, and forget gates, which allow writing, reading, and resetting to the cell memory. The presence of this gate allows the LSTM cell to store indefinitely. In fact, if the next input gate is the activation threshold, the cell remains in the previous state and is combined with the input value when the current state is active. As the name suggests, the forget gate resets the current state of the cell, and the output gate determines whether the cell value is executed. [4]

C. Implementation

The equations are an exception to the prescribed specifications. In this paper, we use a python program and a Tensor-flow engine to easily implement a deepener for the implementation of the simulator. Tensor-flow is an open source software library for machine learning that is used in google products. It is an engine that makes it possible to implement artificial intelligence with data only, and it is provided by google free of charge.

1) Input data

The power data collected by remote meter reading include various types of data such as load profile (LP), current meter reading, periodic meter reading, and system status information. In the simulator for fault diagnosis, the system state information data is used and the power data and status information data items are shown in Table □. [5]

TABLE I. ITEM BY DATA TYPE

Data Type	***Items***
Power data	Current meter reading LP Regular meter reading Blackout Maximum load current Maximum demand Average voltage / current Instantaneous voltage / current
System Status Data	DCU operation status CPU utilization Memory utilization Enclosure temperature Enclosure humidity Metering Agent Status NMS Agent Status SMS Agent Status DB status

In this paper, we use the state information of the operating system in the field as input data of the simulator. The input engine of the simulator processes the system state information data collected by the server into the sqlite DB file, which is used as the input of the simulator, and inputs the processed data to the learning engine when the learning engine is operated. The input data required for learning is reduced by using preprocessing process using data mining technique to eliminate meaningless data and to shorten the accuracy of diagnosis and the analysis time required through learning.

2) *Model Fitting*

We fitted the model to diagnose the system failure. We applied RNN, which is a model that can effectively learn each state information because they have correlation with each other and previous data are continuously influencing. It consists of 10 system state data and the output is faulty through input learning. Based on this, we performed model fitting work suitable for implementation. The most important consideration when performing model fitting operations is the state of the input data. Ten of our input data are received periodically every 15 minutes and are correlated to each other in terms of fault conditions. Therefore, we used RNN model which is suitable for the learning of related input elements.

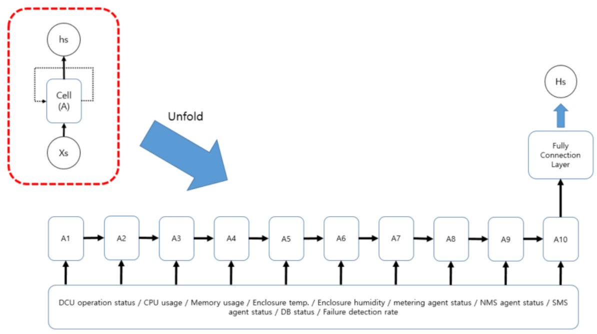

Fig. 4. RNN block diagram

In the fault diagnosis using RNN, each state information is relate to each other, and the state of the current item can be obtained through calculation using the two inputs of the state and value of the previous item. If we perform this procedure in a sequential manner, we can finally obtain the current state of the system we want to obtain and are shown in (2) and (3). In order to improve the accuracy of the system diagnosis, various input items, weight for each item, and experience for model fitting are important.

$$h_s = \tanh(W_{hh}h_{s-1} + W_{xh}x_s) \quad (2)$$

$$y_s = W_{hy}h_s \quad (3)$$

Cell configuration uses LSTM method which can solve long-term dependency problem while sharing previous information, and its code is as follows.

```
Cell = tf.contrib.rnn.BasicLSTMCell(num_unit = hidden_size)
```

3) *Training*

After fitting the model to diagnose the system failure, we proceeded to study. The parameters used for learning are shown in Table □.

TABLE II. RNN PARAMETER

Item	***Values***
Sequence number	10
data dimension	10
hidden dimension	20
output dimension	1
learning rates	0.01
epoch number	2000

Here, the data dimension represents the type of state data to be used as an input, totaling ten. The double failure rate is not the data collected from DCU, but the final data item we want to know. The sequence dimension is for 10 sequential associativity data calculations. A hidden size means the number of intermediate outputs and arbitrarily set to 20. The input dimension is the input data item for learning and the output dimension is the final output. A total of 1,000 set of 15 minute collected data were used for the study. We also normalize the input data to overfitting and also perform a preprocessing process to prevent duplicated features. In addition, the dropout technology for processing a large amount of data has not been used because it is considered that the system does not need to be used.[6] Here is the Python code for learning:

```
multi_cells = tf.contrib.rnn.MultiRNNCell([lstm_cell() for _ in range(1)], state_is_tuple=True)
hypothesis, _states = tf.nn.dynamic_rnn(multi_cells, X, dtype=tf.float32)
Y_pred = tf.contrib.layers.fully_connected(hypothesis[:, -1], output_dim, activation_fn=None)
loss = tf.reduce_sum(tf.square(Y_pred - Y))
optimizer = tf.train.AdamOptimizer(learning_rate)
train = optimizer.minimize(loss)
rmse = tf.sqrt(tf.reduce_mean(tf.squared_difference(targets, predictions)))
with tf.Session() as sess:
        init = tf.global_variables_initializer()
        sess.run(init)
        for epoch in range(epoch_num):
            _, step_loss = sess.run([train, loss], feed_dict={X: trainX, Y: trainY})
            test_predict = sess.run(Y_pred, feed_dict={X: testX})
            rmse_val = sess.run(rmse, feed_dict={targets: testY, predictions: test_predict})
```

III. EXPERIMENTAL RESULTS

We have developed a simulator for system fault diagnosis through analysis and learning of big data related to system state information among power data transmitted in real time in the system and simulated for its verification. The test used randomly generated data for the verification of the learning algorithm and compared the predicted value with the set value. For each of the 3 DCUs operating in the field, 200 actual data were applied to calculate the failure rate.

The specifications of the computer used in the test are shown in Table III.

TABLE III. TEST SYSTEM SPECIFICATIONS

Item	*Specifications*
CPU	Intel Core i7-6500U CPU@2.50GHz, x64 based
MEMORY	16GB RAM
HD	240GB SSD
OS	WIN10, 64 bit based

As the first step of the test, we used 1,000 randomly generated 15 minute collection data for simulation verification. The larger the amount of data, the higher the probability of fault diagnosis. 1,000 randomly generated data were used for learning and 400 for prediction of failure rate. We compare the failure rate with the random failure rate and the failure rate predicted through learning. The results are shown in Fig. 5.

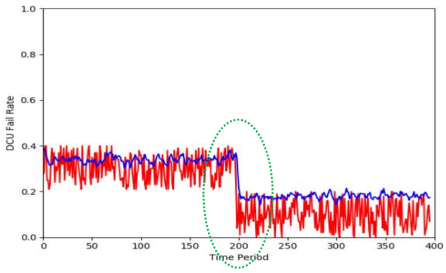

Fig. 5. Failure rate through random setting vs failure rate

In the figure, blue represents the failure rate due to learning, and red represents the failure rate arbitrarily set. It can be seen that the result of the prediction for the total of 400 data is almost the same as the set value. Therefore, the parameter application and accuracy of the algorithm can be relied on. DCU is a system that continuously produces and stores various kinds of power data, and is a key device of remote meter reading. DCU is also a system that operates 365 days a year, so reliability and stability are important. However, since the DCU is installed outside as shown in Fig. 5, the system is frequently damaged due to its environmental impact. We tried to predict the failure rate of the system and take appropriate measures in a short period of time to improve the stability and reliability of the system. Therefore, the power data collected by the DCU as the input of the simulator was applied to the test. Fig. 6 shows a real picture of the DCU that brought the test data

Fig. 6. Demonstration DCU

We conducted experiments using real data using simulation algorithm that was verified in the first step. The test consisted of a total of 1,000 data sets, including 600 data used for learning in step 1, 200 data for diagnosis used in step 1, and 200 data generated from the target system running on the field. The 200 sets of actual data in our system are very small compared to the data we have accumulated so far. However, our purpose in this paper is to understand the applicability of the algorithm through simulation and to apply the algorithm to the system in the future so as to grasp the failure rate of the system in real time. We simulated three DCU systems and the results are shown in Fig. 7.

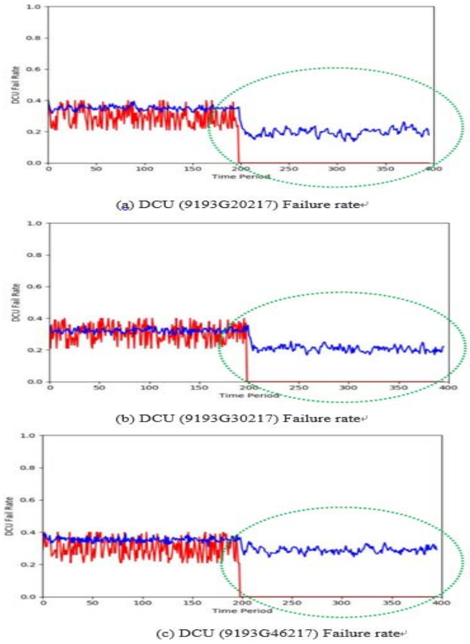

Fig. 7. Probability of Failure Diagnosis by DCU

The three systems show a similar tendency in the result of prediction of failure rate through learning. In the figure, time 200 ~ 400 is the section we want to predict. Actually, the failure rate of the system in this section is 0% and it is shown as red color in the figure and the system is in normal operation state. The system failure rate of 0% is an arbitrary value that means that the system is currently operating normally and that the meter reading service is in progress. Our simulation predicts a failure rate of approximately 20 to 40% in this section. This shows a normal operation state with a low value compared with a failure rate of 100%. In our experience, we mean a normal operating condition with a fault rate of 0 to 60%, a 61 to 80% risk of future failure, and a very high probability of actual failure at 81 to 100%. Of course, 0 ~ 60% of the failure rate is classified as a normal operation state, but should take appropriate action through detailed definition. In Fig. 7, the difference in failure rates between each DCU is related to CPU utilization. CPU usage is similar in most cases, generally 0 ~ 30% is used. However, some systems may use up to 90% CPU utilization. It is classified as an abnormal state with extremely exceptional conditions. The system we used in our tests is 10-20% for 9193G20217, 20-30% for 9193G30217, and 30-40% for 9193G46217. Therefore, it can be seen that the system failure rate difference between 200 and 400 is proportional to the CPU utilization as shown in the figure.

IV. CONCLUSION

In this paper, we performed the simulation for the diagnosis of the failure rate of DCU, which is a remote meter reading system, in this paper. The power field, including AMI and Smart Grid, is expected to be driven by the application of services through data mining and big data analysis as the services are expanded. As the AMR service is expanded, the amount of electric power data is increasing exponentially and it is expected that it will be possible to apply various useful services through analysis and prediction of such data. Therefore, in this paper, we conducted a study to diagnose the failure of the system using the system state data among the remote meter reading power data provided by AMR and to infer the state of the system in the future. A Tensor-flow engine and an artificial intelligent deep-running algorithm are applied as an algorithm for fault diagnosis and prediction of the system. For the first time, we fitted the model with the state information of the system needed to determine the failure rate. Since each state information is related to each other, we applied the RNN model and applied the LSTM method to solve the long - term dependency problem by sharing the previous information to construct the cell. After the model fitting, simulator was implemented and tested. The test proceeded in two steps. First, to verify the accuracy of the algorithm and the appropriateness of the parameters, arbitrary data was generated and simulated. As a result of our experiments with 1,000 data sets, we confirmed that there is no significant difference between our set value and the simulation result through learning. Based on these results, we conducted a test with 200 sets of data for the second actual operating system. Simulation results for all three systems show a failure rate of 20 to 40% in all systems. This can be classified as a normal operating state in the experience. In order to perform appropriate control according to the future failure rate, detailed classification should be made for each step. Each system has about 20% difference in failure rate, but it is considered to be affected by the CPU usage of the system. Based on the results of the simulation, we plan to conduct a deep learning study on a large amount of data in the future. In the future, we will implement a system that operates reliably and reliably by reducing the failure rate of the system by applying mining, big data interpretation, and prediction of the accumulated and stored power data based on the results of this study. By analyzing various kinds of power data including the state information of the system, it can be used for power grid design, countermeasures against old wires, demand response, forecasting of demand, and contributing to activation of smart grid service.

REFERENCES

[1] Park. B.S, Hyun, D.H, Cho. S.K, “Implementation of AMR system using power line communication,” Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES, Vol. 1, pp. 18-21, Oct. 2002.

[2] Seo. C.K, Jeong. S.M, Chung. D.J, “A Research on the Extraction and Interpretation of power line communication Noise Pattern Using Genetic Algorithm,” Advanced Engineering Forum, Vol. 2-3, pp. 645-648, 2012.

[3] Kwon Yun Lee, Sang Sue Lee, "Artificial Intelligence Implemented with Python and Tensor Flow," Book Publishing Global, pp. 300-301, Sep. 2017.

[4] Jeongsam Bum, "Deep Learning and Reinforcement Learning for Implementing Tensor Flow," ACOR Publishing Co., pp. 23-25, Oct. 2017.

[5] General Technical Specifications of KEPCO, "Data Concentration Unit for Low Voltage AMI System," KEPCO General Purchase Specification, GS-5895-0026, pp. 256-257, 101-102, Feb. 2011.

[6] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” Journal of Machine Learning Research 15, 2014. .