Intermittent Demand Forecasting with a Recurrent Neural Network Model Using IoT Data

Kangbae Lee[1], Dong Yeon Kang[*], Hyung Rim Choi[2],
Byung Kwon Park[3], Min Je Cho[4] and Doo-hwan Kim[5]

[1, *, 2, 3, 4, 5]*Department of Management Information Systems,
Dong-A University, Republic of Korea*
[1]*kanglee@donga.ac.kr, [*]ghkxk01@donga.ac.kr, [2]hrchoi@dau.ac.kr,
[3]bpark@dau.ac.kr, [4]mjcho@dau.ac.kr and [5]kdhblack@donga.ac.kr*

Abstract

Many businesses are becoming increasingly interested in managing the supply chain to survive in today's dynamic market. Of the many factors that influence supply network maintenance, demand forecasting forms the origin of all activities in the supply chain. It has, however, been irregular and inadequate at forecasting in a dynamic market. The intermittent and irregular characteristics of replacement demand for products and parts make forecasting demand particularly difficult. Moreover, as information concerning demand from a specific period is often required on site, a demand forecasting method unlike those available at present is needed.

In this study, an Internet of Things environment is built and a technique proposed to forecast replacement demand in a specific period featuring intermittent and irregular demand characteristics. The study measured the lifespans of batteries, and compared the Weibull distribution, artificial neural networks (ANNs), and recurrent artificial neural networks.

The ANN had a shorter application period and higher forecast rate than the demand forecasting method employing the Weibull distribution. Further, when this network was circulated, the forecast rate improved over the general ANN.

Keywords: *IoT, demand forecasting, artificial neural network, failure forecasting, intermittent demand*

1. Introduction

Businesses in a dynamically changing market work to dynamically respond to customer demands [1]. As demand forecasting is a factor that directly influences effective supply chain management, it is the origin of all activities in the supply chain. However, many limitations hinder effective demand forecasting at present, as it fails to consider complicated business environments and assumes that non-linear business environments are linear [2].

Irregular and intermittent demand may occur in a dynamic market, but current demand forecasting methods tend to forecast with excessive demand. Although a variety of demand forecasting methods have been researched, such as the dual exponential smoothing method and forecasting through a Weibull distribution, they rely on distributions and cannot provide an accurate forecast for a specific period, as demand information specific to the period in question or point on site is needed in this case [3].

---

[1] First Author
[*] Corresponding Author

This study proposes a method to forecast irregular and intermittent demand, which has been difficult to forecast thus far, using an artificial neural network (ANN) to accurately forecast replacement demand according to each time point.

2. Theoretical Basis

2.1. Demand Forecasting

Demand forecasting can be considered the most important factor assisting businesses make decisions, as it predicts the near future and helps advance a company's efficiency. A majority of companies today enforce supply chain management (SCM) to increase efficiency. Demand forecasting can be divided into qualitative, quantitative, and systematic methods according to the characteristics of demands and the information that can be secured.

This study uses ANNs and a systematic method to accurately forecast irregular and intermittent demand.

2.2. Characteristics of Intermittent Demand

The characteristics of intermittent demand must first be examined before forecasting demand. Demand can be divided into regular and irregular demand based on regularity. Irregular demand can be further divided into intermittent and non-intermittent demand.

Irregular and intermittent demands, the subject of this study, are difficult to accurately forecast using traditional demand forecasting methods. Intermittent demand features periods of zero demand interspersed with irregular demand. Therefore, it differs from the forecasting of irregular but non-intermittent demand, as both the point at which demand occurs and the demand scale need to be predicted.

2.2.1. Studies on Forecasting Intermittent Demand

Research has been conducted on forecasting a product's replacement time and replacement volume to forecast intermittent demand. The Weibull distribution is the most widely used method to forecast replacement demand.

The Weibull distribution indicates the failure rate through a general U-shaped distribution. An initially high failure rate occurs owing to a product's initial defect rate at the start of production. Once sales are complete, the product's lifespan is exhausted over time and the failure rate begins to increase.

The exponential smoothing method forecasts demand by analyzing general time series data, but Croston predicted intermittent demand through the exponential smoothing of demand size and the interval of demand occurrence. Intermittent characteristics, in other words, are reflected in the general time series analysis method [4].

The ANN used in this study was also used to forecast intermittent demand. Gutierrez used an ANN to forecast intermittent demand, and built a neural network that learns and analyzes the lengths of sections where demand does not occur as well as the size of the demand that occurs thereafter to forecast intermittent demand. This method can arrive at conclusions most similar to the data in comparison with other methods of inference. However, as it forecasts the size of future demand after a certain point in time, it is difficult to apply to industries with long production lead times, or those that require immediate responses.

Chan–Yeong Hong combined Bayesian probability and recurrent ANNs to analyze irregular time series data and predict intermittent demand. The ANN in his method did not learn data itself but instead learned processed them with different weighted values according to the lengths of sections without demand using Bayesian probability to forecast demand [6].

2.3. Research Subject Selection

Specific product information can be acquired in real time in the Internet of Things (IoT) environment. Thus, demand can be predicted using a variety of past information. This study builds a model that forecasts demand under the assumption of an IoT environment.

Research subjects were selected from products involving replacement demand by considering whether real-time information can be acquired based on the IoT environment.

Batteries are the most representative functional product with replacement demand. Regarding lifespan experiments, as batteries do not possess much internal power, information concerning the status of various batteries can be collected within a short period. Therefore, this study verified the proposed demand forecasting model using batteries as subject.

2.3.1. Failure Modes According to Product Characteristics

Figure 1 illustrates three failure modes for the product. "Functional parts" are replaced due to factors affecting the product's lifespan, such as wear and tear; "consumable parts" must regularly be replaced with time; and "incidental parts" are replaced due to incidents. A product's consumable and incidental parts are replaced owing to the lapse of time or because a certain incident has occurred, and not because demand occurs according to the product's state [7]. Therefore, this study focuses on forecasting intermittent demand based on functional parts.
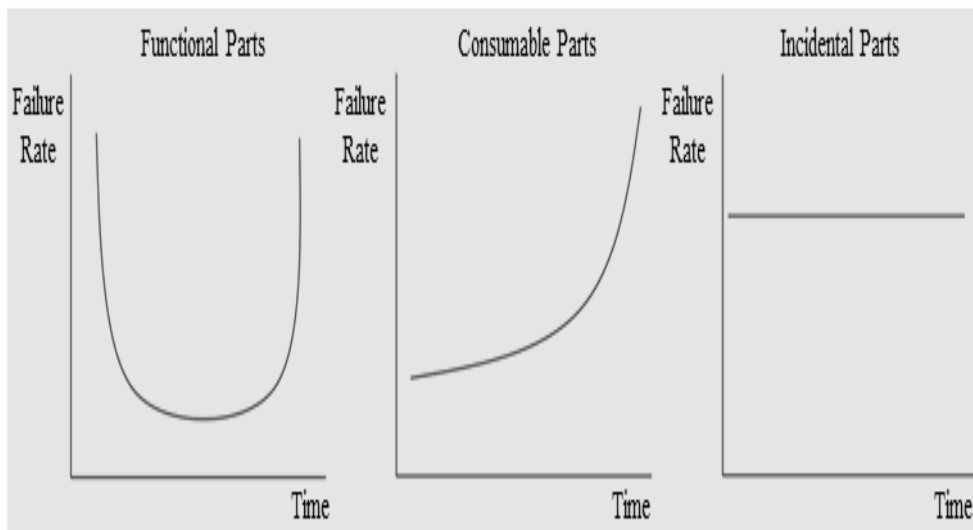

Figure 1. Modes of Parts' Failure

Functional parts include tires or batteries, and their lifespans are affected by wear and tear, power, and other similar factors. Products that have completed their lifespan must be replaced, which differentiates them from consumable parts. Functional parts have a main body where IoT equipment can be installed, and the time of use influences the product's lifespan [5].

Therefore, if the user only intermittently uses the product, it becomes difficult to forecast when demand for a replacement occurs. This study examines the states of products in real time and analyzes the relevant data using an ANN. Further, it seeks to sensitively respond to the amount of intermittent usage to accurately forecast demand.

2.4. Artificial Neural Network

The ANN is a systematic model of complex activities that occur in the human brain. It is known to have greater predictive power than a regression analysis for a non-linear correlation between independent and dependent variables [8].

Neurons transmit information in the brain, where each neuron receives weak electric signals from connected neurons. When the sum of these signals exceeds a certain threshold, the neurons become excited (ignition) and signal other neurons [9]. Delivering these countless signals allows for a high-level information processing mechanism in humans.

This ANN model has been applied to a variety of fields. Lippmann explained that the ANN is the "method with the most standout results in the modeling and forecasting fields, signal processing, professional systems, and other various fields [10]."

The ANN simulates the human brain, and works to more effectively resolve complex issues by simultaneously checking multiple possibilities to deduce a conclusion [8]. Therefore, it is important to quickly process complex calculations. Many input variables are added using their weighted values and are output through an activation function. The Heaviside function, which outputs a value of zero or one based on a specific threshold, and the S-shaped sigmoid function are typically used as ANN activation function.

2.4.1. Concept of the Artificial Neural Network

The basic structure of the ANN is similar to that of neurons in the human brain. Multiple variables are combined using an activation function and entered accordingly. The sum of the inputs is then converted through a non-linear function, or the activation function, and generates the output [9].
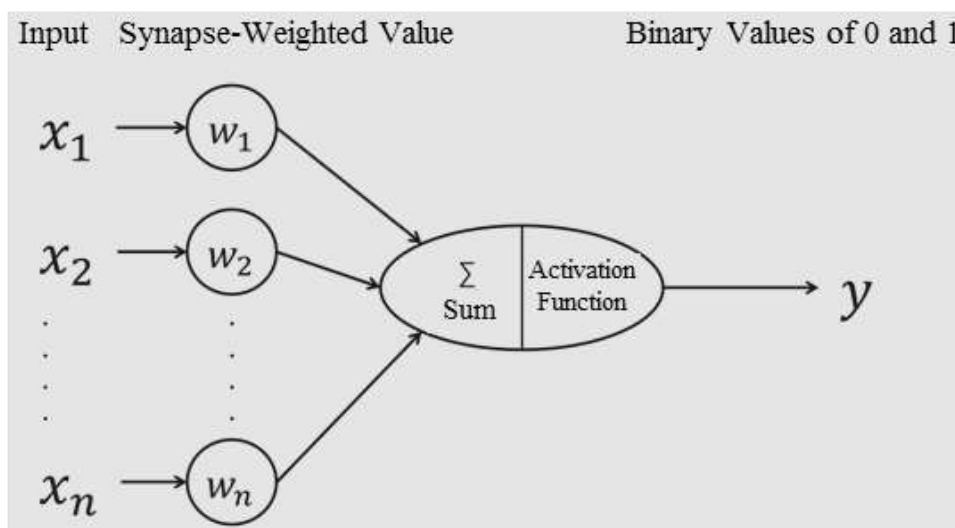

Figure 2. Model of Neuron

2.4.2. Application of Artificial Neural Networks

Odom and Sharda, and Geon-Chang Lee used ANN model to forecast company bankruptcies [11] [12]. Yeong-Tae Kim used this model to evaluate a company's stock credit level and compared it with another statistical model. So-Ye Han used this model in a study to determine a product's warranty [13] [14]. Another study assessed a vehicle's failure type and predicted additional failures that may occur to establish a quality assurance policy. In-Hwan Ryu used the ANN on time series data to forecast demand, and Yu-Seop Hwang used a case-based inference technique and the ANN to build a service

demand forecasting framework [8] [15]. Moreover, research is actively ongoing on robot control and Artificial Intelligence. Gwang-Ju Lee used an ANN and the genetic algorithm to learn the behaviors of autonomous mobile robots [16].

2.5. Recurrent Neural Network

The recurrent neural network (RNN) is a model that remembers content previously learned by the ANN or outputs to adjust to a dynamic situation, unlike the typical prospective structure of the neural network model. The RNN network can be divided into the Elman or Jordan recurrent neural network depending on factors that are remembered. As the RNN remembers the past, this is deemed appropriate to handle the "time" factor, which advantageous because it is stricter than a regular neural network model [8].

2.5.1. Elman Recurrent Neural Network

The Elman RNN determines the weight of each variable based on the learned information. Newly input information is combined through an activation function according to the determined weights to initiate the activation function [17]. The activation function yields its results to the output layer and circulates them to the hidden layer where they are output. When new information is entered, the Elman RNN includes information from the previous hidden layer in the input layer and learns new information to remember the past.
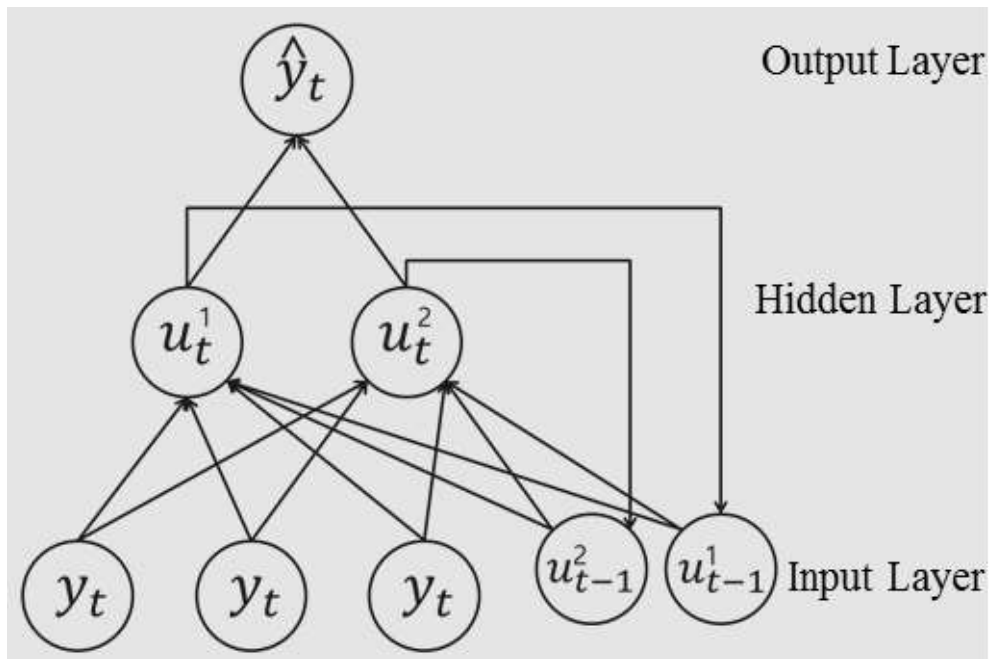

Figure 3. Elman Recurrent Neural Network

3. Research Method and Model

3.1. Research Model

This study differentiated battery depletion using power consumption because the status information of batteries, by assuming an IoT environment, was analyzed through an ANN to forecast the final replacement demand.

It was assumed that failures do not occur as a result of battery defects or accidents. When a battery's power is depleted, it is determined to require a replacement: in other words, demand is generated.

3.1.1. Activation Function of the Model

The ANN outputs a value using the input of many variables. Information input through the activation function can be formulated through the following formula:

$$\text{sum of Inputs} = \sum_{i=1}^{n} (w_n x_n)$$

n = Battery status data count

x_n = Battery's nth status data

w_n = Weighted value of the nth status data

The sum of inputs is converted through the activation function and output accordingly. If the value of the sum of inputs minus the threshold is positive, the output is one, and is zero if the value is negative [9]. Power was measured in real time in this study and entered for each period. The threshold was then subtracted from the sum of inputs. If the calculated value was positive, the output was one, and was zero if the output was negative. When the output was one, the battery was determined to have depleted its lifespan, which was translated as a type of demand.

When defining the activation function as f, the output value as y, and the threshold value as θ, this can be formulated as follows:

$$y_i = \sum_{k=1}^{n} (x_{ki} w_{ki} - \theta)$$

y = Depletion $(1 \text{ or } 0)$

f = Activation funcion

θ = Threshold

k = ith battery's kth status data

3.1.2. Demand Forecast Modeling

In the above formula, y determines whether batteries are depleted by displaying one or zero as output. Therefore, the final demand can be observed as the sum of y, which is formulated as follows:

$$D = \sum_{k=1}^{N} y_k$$

where D is the overall demand and y_k the depletion of the kth battery.

Thus, the formula for overall demand forecasting is as follows:

$$D = \sum_{k=1}^{N} f_k \left(\sum_{i=1}^{n} x_{ki} w_{ki} - \theta \right)$$

D = Overall demand

N = Number of batteries for which data with acquired

f_k = Activation function of the kth battery

n = Number of battery status data

x_i = ith battery status data

w_i = Weighted value of the ith variable

θ = Threshold (battery lifespan)

3.2. Implementing the Measurement Modeling

The most recently preprocessed power data were used to build and verify the ANN. The National Instruments NI9203 module was used to measure battery power, and three 4WD (four-wheel-drive) toy cars were used to consume battery power.

The LabView software was used to record power data from the power measurement model. Power was measured in real time and saved in Microsoft Excel to implement the measurement model.
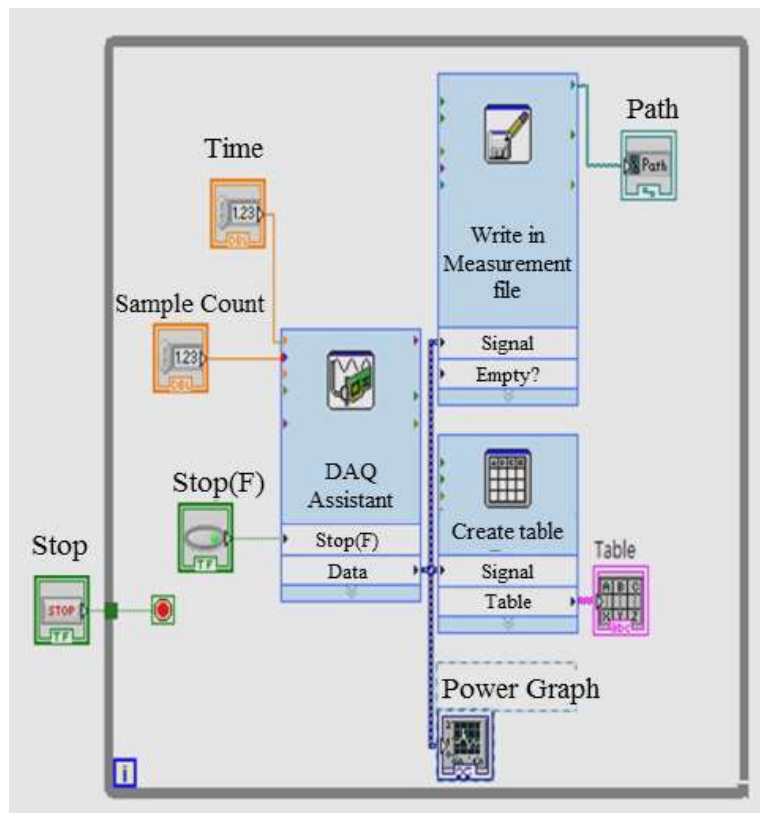

Figure 4. Block Diagram of LabView

The power data measured in real time and saved in Microsoft Excel were accumulated to assume the IoT environment, as Figure 5 indicates.

Power Observation

Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8
0.008603	-0.000447	0.006265	-0.000974	-0.000000	0.007686	0.000000	-0.000000
0.007893	-0.000877	0.006623	-0.000507	-0.000203	0.007388	-0.000000	-0.000000
0.008591	-0.000412	0.005866	-0.000794	-0.000554	0.007899	0.000000	0.000000
0.008538	-0.000488	0.006917	-0.000935	-0.000639	0.007204	0.000000	-0.000000
0.008117	-0.000719	0.005420	-0.000495	-0.000717	0.007497	0.000000	-0.000000
0.008262	-0.000679	0.007683	-0.000469	-0.000000	0.007974	0.000000	0.000000
0.008425	-0.000536	0.007712	-0.000471	-0.000000	0.008008	-0.000000	-0.000001
0.008388	-0.000571	0.007564	-0.000437	-0.000635	0.006654	0.000000	0.000000
0.008356	-0.000572	0.006158	-0.000976	-0.000346	0.007330	0.000000	0.000000
0.007963	-0.000807	0.006953	-0.000890	-0.000271	0.008596	0.000000	-0.000000
0.008626	-0.000391	0.006974	-0.000784	-0.000677	0.007797	-0.000000	-0.000000
0.008119	-0.000698	0.006657	-0.000937	-0.000357	0.007665	-0.000000	-0.000000

Figure 5. Front Panel of Power Observation

Figure 5 illustrates each channel's connection to a battery, and power was measured in real time with a measurement interval of 10 seconds.

The data collected for each battery were divided by 50 to derive information concerning their status.

AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	Volt
0.013	0.0122	0.013	0.0122	0.0125	0.0129	0.0128	0.0128	0.0127	0.013	F
0.0029	0.0027	0.0029	0.0027	0.0029	0.0029	0.0028	0.0026	0.0028	0.0028	F
0.0024	0.0024	0.0025	0.0027	0.0024	0.0023	0.0018	0.0017	0.002	0.0017	F
0.012	0.0119	0.0114	0.0119	0.0115	0.0117	0.0116	0.012	0.0113	0.0115	F
0.0025	0.002	0.0012	0.0026	0.0024	0.0021	0.002	0.0023	0.002	0.0019	F
0.0121	0.0115	0.0116	0.0122	0.0118	0.0116	0.0111	0.0116	0.0117	0.0114	F
0.013	0.0122	0.013	0.0122	0.0125	0.0129	0.0128	0.0128	0.0127	0.013	F
0.0029	0.0027	0.0029	0.0027	0.0029	0.0029	0.0028	0.0026	0.0028	0.0028	F
0.0024	0.0024	0.0025	0.0027	0.0024	0.0023	0.0018	0.0017	0.002	0.0017	F
0.012	0.0119	0.0114	0.0119	0.0115	0.0117	0.0116	0.012	0.0113	0.0115	F
0.0025	0.002	0.0012	0.0026	0.0024	0.0021	0.002	0.0023	0.002	0.0019	F
0.0121	0.0115	0.0116	0.0122	0.0118	0.0116	0.0111	0.0116	0.0117	0.0114	F
0.013	0.0122	0.013	0.0122	0.0125	0.0129	0.0128	0.0128	0.0127	0.013	F
0.0029	0.0027	0.0029	0.0027	0.0029	0.0029	0.0028	0.0026	0.0028	0.0028	F
0.0024	0.0024	0.0025	0.0027	0.0024	0.0023	0.0018	0.0017	0.002	0.0017	F
0.012	0.0119	0.0114	0.0119	0.0115	0.0117	0.0116	0.012	0.0113	0.0115	F
0.0025	0.002	0.0012	0.0026	0.0024	0.0021	0.002	0.0023	0.002	0.0019	F
0.0121	0.0115	0.0116	0.0122	0.0118	0.0116	0.0111	0.0116	0.0117	0.0114	F
0.0085	0.008	0.0089	0.0082	0.0089	0.0083	0.0086	0.0077	0.0081	0.0081	F
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	T
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	T

Figure 6. Power Data

Figure 6 indicates that the rows and columns were switched for the data collected and divided for each battery. Thus, each battery denoted individually observed values. For example, for variable "AY," T was inserted if the battery was depleted at each time point, and F was inserted if it had not been depleted, which enabled the ANN to learn the results.

3.2.1. Building the Artificial Neural Network Model

The most recently preprocessed power data were used to build and verify the ANN. All "NA" values, or power data values when power was depleted and was no longer measured, were processed as zero and the ANN was built accordingly. The network was verified by dividing these data into three datasets.

```
voltT <- A1[A1$Volt == "TRUE",]
voltF <- A1[A1$Volt == "FALSE",]
set.seed(1234)
ind1 <- sample(3, nrow(voltT), replace=T, prob=c(0.5, 0.3, 0.2))
set.seed(4321)
ind2 <- sample(3, nrow(voltF), replace=T, prob=c(0.5, 0.3, 0.2))

volt.train <- rbind(voltT[ind1==1,], voltF[ind2 == 1,])
volt.test <- rbind(voltT[ind1==2,], voltF[ind2==2,])
volt.valid <- rbind(voltT[ind1==3,], voltF[ind2==3,])

volt.train$Volt <- class.ind(volt.train$Volt)
volt.test$Volt <- class.ind(volt.test$Volt)
volt.valid$Volt <- class.ind(volt.valid$Volt)

myformula <- Volt ~ .
```

Figure 7. Data Separation

The "Volt" variable in each column extracted only the same columns to divide data into "VoltT" or "VoltF." When configuring the three data types to build the ANN, the seed values for "ind1" and "ind2" were differently set to evenly and randomly insert the data categorized through the Volt variable. Each sample was then extracted and saved.

The ANN then learned all data from the completed "Volt.train" and "Volt.test," and was then configured to start from the first experiment to the verification stage. The ANN was built based on the Volt variable in "myformula" so that the remaining variables could be analyzed.

3.2.2. Deducing the Number of Nodes in Hidden Layer

```
test.err <- function(h.size, maxit0){
  voltmodel1 <- nnet(myformula, data=Volt.train, size=h.size, decay=5e-4, trace=F, maxit=maxit0)
  y <- Volt.test$Volt
  p <- predict(voltmodel1, Volt.test)
  err <- mean(y != p)
  c(h.size, err)}

out <- t(sapply(1:15, FUN = test.err, maxit0=100))
plot(out, type="b", xlim=c(0,15), ylim=c(0.9,1.05), xlab="The number of Hidden units", ylab="Test Error")
```

Figure 8. Deducing the Number of nodes in Hidden Layer

The number of hidden layer nodes in the ANN changes according to the amount of data learned. Therefore, a process is needed determine the number of hidden layers whenever change occurred in the input data to properly use the ANN. Figure 8 shows the process by which the test data were used to check the error rate for each hidden layer.

A graph was used to express the error rate for each result for the hidden layers 1–15. The "Volt.train" data were learned for the model that determined the number of hidden layers and "Volt.test" data were analyzed to derive the error rate for each number of hidden layers.
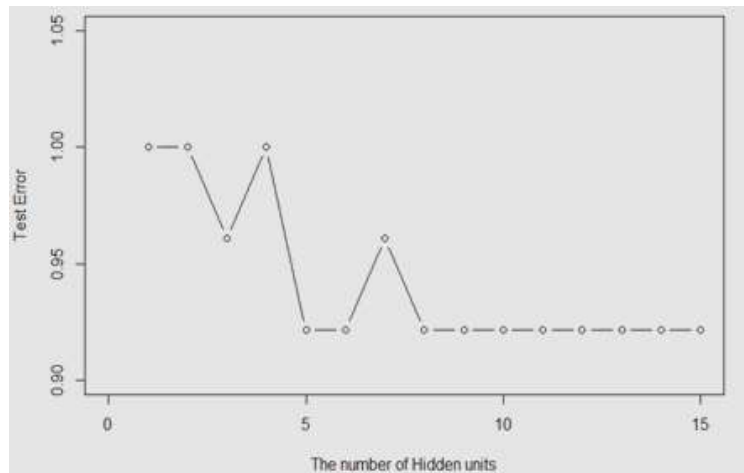
Figure 9. Error Rate of the Hidden Layer

Figure 9 reveals that both the error rate and the number of nodes were lowest when there were five hidden layers. Therefore, this model used five hidden layers, and the ANN was trained as indicated in Figure 9.

3.3. Verification of the Artificial Neural Network

```
model12 <- nnet(myformula, size=5, decay=5e-4, range=0.1, maxit=200, data=volt.train)
summary(model12)

volt.pred.valid <- predict(model12, new=volt.valid)
```

Figure 10. Building the Artificial Neural Network Model

The ANN model was verified as shown in Figure 10. The number of nodes of the hidden layer was set to five through "size = 5" for "model12." The "Volt.train" data were learned for verification and "Volt.valid" data were input to record the predicted value in the "Volt.pred.valid" variable through the ANN.

Table 1. Model Categorization Table (valid data)

Category		Obs (Observed Value)	
		Remaining Batteries	Depleted Batteries
Pred (Predicted Value)	Remaining Batteries	16	0
	Depleted Batteries	0	13
Sensitivity		1	
Specificity		1	
Accuracy		1	

Table 1 records the results of the ANN. The network used was evaluated in terms of "valid data" by configuring and assessing three evaluation metrics: sensitivity, specificity, and accuracy. The accuracy obtained was 100%.

3.4. Experiment on Artificial Neural Network Model

The verified ANN model was tested using data on a larger scale. As with model verification, Volt.train data were learned and a forecasting experiment conducted on Volt.test data. The Volt.test data had the lowest error rate using five hidden layer nodes, and the method of analysis was the same as in the verification process.

Table 2. Model Categorization Table (test data)

Category		Obs (Observed Value)	
		Remaining Batteries	**Depleted Batteries**
Pred (Predicted Value)	**Remaining Batteries**	23	1
	Depleted Batteries	0	27
Sensitivity		1	
Specificity		0.9642857	
Accuracy		0.9803922	

The Volt.test data, which consisted of data for 51 batteries, recorded an accuracy of 0.98, which indicates that the configured model is capable of accurate predictions.

4. Results of Analysis

4.1. Circulation of the Artificial Neural Network

The ANN used in this study was circulated to create a dynamic forecasting model that could adjust to realistic changes. It was circulated based on Elman's RNN model, which circulates nodes in the hidden layer to the input layer to remember the weights of past activation functions. New data from the input layer were entered again into the previously remembered hidden layer to reflect new hidden layers.

A method that only returns properly categorized information was used to circulate information in the hidden layer based on Elman's neural network. Therefore, only data that were correctly predicted in the ANN experiment based on Volt.test were merged with new data and learned.

```
A <- as.data.frame(obs)
B <- as.data.frame(pred)
C <- cbind(A,B)
D <- volt.test
E <- cbind(D,C)
View(E)

F <- subset(E, obs==pred)
View(F)
G <- F[,-c(52,53)]
View(G)
View(spam)
```

Figure 11. Recurrent Data Selection

The subset function in Figure 11 was used to select only properly predicted data. The additional "obs" and "pred" variables were deleted and preprocessed to be merged with new data.

4.2. First Recurrent Neural Network

Data correctly categorized in the previous experiment were relearned using the previous set of learning data. The ANN was then circulated through the same procedure as above.

A maximum of 14 suitable hidden layer nodes occurred in the first recurrent neural network and 144 newly categorized battery data items were entered. The measurement point was randomly set for each battery.

Table 3. Model Categorization Table (1st circulation of data)

Category		Obs (Observed Value)	
		Remaining Batteries	Depleted Batteries
Pred (Predicted Value)	Remaining Batteries	66	3
	Depleted Batteries	3	72
Sensitivity		0.9565217	
Specificity		0.96	
Accuracy		0.9583333	

Table 3 shows that six errors occurred in the first circulation of data. The RNN was less sensitive than the regular ANN to them because when the amount of newly entered data is greater than learned data, sensitivity decreases by one. However, it remained high. The specificity also increased as the neural network was circulated; as more data were

learned, more stringent standards were applied on batteries corresponding to the replacement demand.

4.3. Second Recurrent Neural Network

Data correctly distinguished in the first recurrent neural network were relearned and 144 new battery data were added and distinguished accordingly.

Table 4. Model Categorization Table (2nd Circulation of Data)

Category		Obs (Observed Value)	
		Remaining Batteries	Depleted Batteries
Pred (Predicted Value)	Remaining Batteries	66	1
	Depleted Batteries	3	74
Sensitivity		0.9565217	
Specificity		0.9866667	
Accuracy		0.9722222	

Table 4 reveals that the results were better than those from the first recurrent neural network analysis. A total of 144 battery data items were extracted at random times; 75 depleted and 69 remaining batteries were analyzed. Of the 75 depleted batteries, 74 were correctly predicted, and 66 of the 69 remaining batteries were correctly predicted.

Sensitivity spiked at 0.956, identical to sensitivity results from the first RNN. However, specificity spiked at 0.986, superior to the results of the first RNN.

Of the 144 batteries in the last 500 seconds, 77 were predicted for replacement demand with an accuracy of 97%. Thus, the ANN can provide better results when circulated.

5. Conclusion

A forecasting method was proposed in this study based on the IoT environment to predict intermittent demand. This method is more accurate than forecasts that rely on distributions when acquiring real-time information. If an ANN can acquire information in real time, more accurate forecasts can be made than using existing demand forecasting methods.

This study collected and analyzed battery data to test the proposed method. An environment similar the IoT was implemented by assessing power using a power measurement module.

The ANN model proposed here can make predictions in a shorter time than existing time series-based forecasting methods. It is more accurate even though it learns a relatively small amount of data. If data are circulated and learned accordingly, its accuracy increases. The RNN proposed in this study is applicable to the industry in the context of an IoT environment.

However, this study is limited in that it only used data concerning the amount of power. If a future IoT environment is built and more data are combined, more complex predictions will become possible.

This study also determined only whether a battery was depleted. Thus, it was difficult to predict its expected remaining lifespan. Future studies should distinguish such values, either by using the sigmoid function or duplicating the Heaviside function. Hence, they can not only forecast demand in a short period, as proposed in this study, but can also make more accurate demand forecasts over the long term.

Acknowledgments

References

[1]   D. Simchi-Levi, E. Simchi-Levi and P. Kaminsky, "Designing and Managing the Supply Chain: Concepts", Strategies, and Cases, McGraw-Hill, **(1999)**.
[2]   I. Jagielska and A. Jacob, "A Neural Network Model for Sales Forecasting", in Artificial Neural Networks and Expert Systems, Proceedings, First New Zealand International Two-Stream Conference, **(1993)**.
[3]   J.-G. Sun, J.-S. Park, N. T. Khan and Y. Kim, "A Study on Developing Forecasting Methodology for Parts including Intermittent Demands", Conference in Korean Institute of Industrial Engineers, **(2016)**.
[4]   J.D. Croston, "Forecasting and Stock Control for Intermittent Demands", Journal of the Operational Research Society, vol.23, no.3, **(1972)**, pp. 289-303.
[5]   R.S. Gutierrez, A.O. Solis and S. Mukhopadhyay, "Lumpy Demand Forecasting Using Neural Networks", International Journal of Production Economics, vol.111, no.2, **(2008)**, pp. 409-420.
[6]   C.-Y. Hong, J.-H. Park, T.-S. Yoon and J.-B. Park, "A Study on the Bayesian Recurrent Neural Network for Time Series Prediction", Institute of Control, Robotics and Systems, vol.10, no.12, **(2004)**, pp. 1295-1304.
[7]   S.-G. Hong, "Development of the Forecasting Model for Parts in an Automobile", Seoul University, **(2000)**.
[8]   I.-H. Ryu, "A Comparative Study of Time Series Forecasting by Artificial Neural Networks", Yonsei University, **(2006)**.
[9]   I. Kobayashi, "Basics of Artificial Intelligence", Dream Media, **(2008)**.
[10] R. Lippmann, "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, vol.4, no.2, **(1987)**, pp. 4-22.
[11] M. Odom and R. Sharda, "A Neural Network Model for Bankruptcy Prediction", Proceedings of the IEEE International Conference on Neural Networks, **(1990)**.
[12] K. C. Lee, "A Comparative Study on the Bankruptcy Prediction Power of Statistical Model and AI Models: MDA, Inductive Learning, Neural Network", International Journal of Management Science & Financial Engineering, vol.18, no.2, **(1993)**, pp. 57-81.
[13] Y.-T. Kim and M.-H. Kim, "The Comparison of Predict MDA with Artificial Neural Networks for Credit Valuation of Bonds: Focus on NASDAQ". Korean Accounting Journal, vol.10, no.4, **(2001)**, pp. 1-24.
[14] S.-Y. Han, "Analysis of Two-dimensional Warranty Data Using Artificial Neural Network". KAIST, **(2004)**.
[15] Y.-S. Hwang, "A Hybrid Forecasting Framework based on Case-based Reasoning and Artificial Neural Network", Journal of Intelligence and Information Systems, vol.18, no.4, **(2012)**, pp. 43-57.
[16] K.-J. Lee, "Online Evolution of Complex Robot Behavior Using Neurogenetic Programming". Seoul University, **(2000)**.
[17] D.T. Pham and D. Karaboga, "Training Elman and Jordan Networks for System Identification Using Genetic Algorithms", Artificial Intelligence in Engineering, vol.13, no.2, **(1999)**, pp. 107-117.

Authors

Kangbae Lee, he is a professor of MIS at the School of Business in Dong-A University. He received his Ph.D. in industrial engineering from Korea Advanced Institute of Science and Technology (KAIST). His research interests include the IoT, demand forecasting, machine learning, and industrial engineering.

Dong Yeon Kang, he received his B.S. in Political Science and Diplomacy from Dong-A University. His research interests include the IoT, machine learning, and demand forecasting.

Hyung Rim Choi, he is a professor of MIS at the School of Business in Dong-A University. He received his Ph.D. in management science from Korea Advanced Institute of Science and Technology (KAIST). His research interests include the IoT, supply chain management, logistics, and demand forecasting.

Byung Kwon Park he is a professor of MIS at the School of Business in Dong-A University. He received his Ph.D. in industrial engineering from Korea Advanced Institute of Science and Technology (KAIST). His research interests include the IoT, supply chain management, database, and demand forecasting.

Min Je Cho, he is a professor of MIS at the School of Business in Dong-A University. He received his Ph.D. in MIS from Dong-A University. His research interests include the IoT, supply chain management, demand forecasting, and business intelligence.

Doo-hwan Kim, he received his Master's in Business Administration from Dong-A University. His research interests include the IoT, supply chain management, machine learning, and demand forecasting.