Deep Learning Application in Mechatronics Systems' Fault Diagnosis, a Case Study of the Demand-Controlled Ventilation and Heating System

Ali Behravan, Roman Obermaisser, Mohammad Abboush
University of Siegen, Siegen, Germany. Email: ali.behravan@uni-siegen.de

***Abstract*—Mechatronics systems include a vast range of interdisciplinary area of electrical and mechanical systems such as heating, ventilation, and air-conditioning systems based on building automation systems are responsible to provide occupants a comfortable and productive environment in buildings. The demand-controlled ventilation system as an advanced control approach in smart buildings is used for the main goal of energy reduction. But, these kinds of systems because of their numerous components such as sensors and actuators are very prone to the faults. Arise of the faults, if they are not detected and diagnosed early, can lead to system's performance degradation or extra maintenance cost and effort. Nowadays, introducing a suitable generic technique for fault detection and diagnosis is an utmost challenge. The contribution of this paper is to present a novel fault detection and diagnosis framework based on deep learning method using long short-term memory units for a case study of mechatronics systems, demand-controlled ventilation and heating system. This paper presents all the steps including data acquisition, data preprocessing, network model design, model optimization and network model evaluation. Ten types of faults in different classes as well as the healthy data are used to train and evaluate the performance of the designed network model. The results describe a high accuracy (97.4%) with via the designed deep neural network. Also, this study describes the methodology of selecting the optimum parameters of training process by analyzing the effect of each parameter on the training accuracy.**

***Index Terms*—Deep Learning, Deep Neural Network, Fault Detection and Diagnosis, HVAC**

I. Introduction

In recent years, the efficient use of energy in building sector motivates the researchers to focus on the new technologies such as building automation systems (BAS). Most of these new technologies are based on a mechatronics system platform as they include both mechanical and electrical components. Heating, ventilation, and air conditioning (HVAC) is the major part of BAS from energy consumption perspective. Therefore, this sector absorbed lots of researches during last years. The other parts can be lighting control system, automated home security, appliance control, smart water supply, and irrigation. The demand-controlled ventilation system (DCV) besides heating system, plays an important role in energy reduction by the automatic adjustment of ventilation according to the fresh air demand and environment temperature. Brandenmuehl et al. illustrate 15% to 25% of the HVAC system's energy can be saved by setting the ventilation rates based on the maximum occupancy fresh air requirement [1]. Behravan et al. described thermal dynamic modeling and simulation of a heating system for a multi-zone office building equipped with demand controlled ventilation using MATLAB/Simulink [2]. These new technologies are developed based on wireless sensor and actuator networks (WSAN) that network components e.g. sensors, actuators, and controller in such network communicate through a wireless network. Authors in the last study, described all the details regarding the configuration of this type of WSAN in reference [3]. Several studies demonstrate arise of faults depending on their type and severity cause waste of energy ranging from 10% to 40%, performance degradation, or excess maintenance effort [4], [5], [6], [7], [8]. The faults may occur in components such as sensors and actuators or in network fabric. Faults can be defined in different aspects of data centric or system centric. Therefore, it is vital to detect and diagnose faults early using an optimized technique to prevent extra maintenance costs and efforts. There are several techniques for fault detection and diagnosis. Some main categories are data-driven methods, knowledge-based methods, and analytical-based methods that each category includes different methods in its sub-levels reported by Katipamula and Brambley [7].

During recent years, various researchers had demonstrated the success of deep learning models in the application of machine health monitoring. Hua et al. proposed an intelligent fault diagnosis network for variable refrigerant flow (VRF) systems using Bayesian belief network (BBN) by determining the suitable network structure and probability distributions of BBN and two typical faults had been taken into consideration for detecting and diagnosing (Refrigerant leakage and Refrigerant overcharge) [9]. Lee et al. had described a method to detect and diagnose three abnormal states in the air handling unit (AHU) with the popular deep learning model, called Deep Belief Network (DBN), combined with Restricted Boltzmann Machine (RBM). That study had taken three types of faults into account which instances are a fan getting stuck, leakage in the cooling coil valves and low efficiency of heat exchanger. The accuracy of the results of the study's fault detection and diagnosis was an approximate score of above 95% [10]. Guo et al. developed a fault diagnosis model based on the DBN and investigated its potential to diagnose the faults of the variable refrigerant flow system. The result of the study showed that the fault diagnosis correct rate of the optimized model is 97.7% [11].

In this study, the deep learning approach is developed using

MATLAB R2018b in details and a deep neural network is designed for the main goal of fault detection and diagnosis. The study proposed a novel model using long short-term memory units (LSTM) to detect different types of faults in a case study of the demand-controlled ventilation and heating system based on wireless sensor network.

The proposed model cover the gap of other models in terms of monitoring the training progress and loss function as is described in section II.C.b), as well as decreasing the required training time and handling huge amount of data set, especially for this kind of mechatronics frameworks.

II. Deep learning Application for Fault Detection and Diagnosis

One promising way to study the fault detection and diagnosis methods in different scenarios and comparing accuracy is using simulations. Beforehand, a fault injection framework is needed. Fault injection is a technique to produce faults with a reproducing ability in the system to trace the behavior of the system in existence or absence of different types of faults [3].

Deep Learning (DL) is defined as a machine learning technique that employs a deep neural network and can learn a long chain of causal links. The deep neural network is a multi-layer neural network which contains two or more hidden layers.

In traditional machine learning methods, a domain expert is a prerequisite in order to extract features and to make the data more clear for classification. The major advantage of deep learning algorithms is that they do not need the domain expert knowledge for feature extraction as they do this process automatically in an incremental way. The other superiority is the performance of deep learning algorithms with increase in amount of data will be boosted up, in contrast to machine learning algorithms which are not very suitable for a huge amount of data.

This section describes a step-by-step deep learning approach of the fault diagnosis developed in MATLAB R2018b. Figure 1 shows these main steps include: data acquisition, data preprocessing, network model design, network model evaluation, and optimization of the best-chosen deep neural network.

A. Data Acquisition

Data acquisition is the first step of the training model in all types of fault detection and diagnosis. Basically, the fault detection model is implemented along with the existing data that had been collected from the system. In this study, the data are collected from variables e.g. occupancy, temperature, and CO_2 concentration sensors and damper and heater actuators during simulations. In order to collect the data respective to several types of fault, the model runs in different healthy (non-faulty) and faulty scenarios. Behravan et al. described the complete procedure of fault injection in reference [3]. In this paper, Healthy Mode (HM), Gain Fault in Temperature Sensor (GF T), Offset Fault in Temperature Sensor (OF T), Damper Fault On (DF), Damper Fault Off (DF), Heater Fault On (HF), Heater Fault Off (HF), Battery Fault in Temperature Sensor (BF T), Battery Fault in CO2 Sensor (BF CO2), Gain Fault in CO2 Sensor (GF CO2) and Communication Fault in Router (CF) are some lables for different scenarios. The simulation time for each scenario is one day (86400 seconds) and the sampling time is one second. Table 1 describes a sample of the data acquisition.

B. Data Preprocessing

The main goal of the data preprocessing is to represent the raw data in a more clear and useful representation. Deep learning is a design based on algorithms that learn from data, so it is essential to use the right data for solving the problem of interest. Therefore the quality of collected data is very important. It is needed to make sure that the data is in the right format, correctly scaled and includes the features which are a description the use case. In this step, the more important features for the training process are selected by the network and the labels of the selected data are generated manually by writing the MATLAB code. The labels of the data are : 'Gain Fault in Temperature Sensor Room1', 'Offset Fault in Temperature Sensor Room1', 'Damper Fault in Room1(OFF)', 'Damper Fault in Room1(ON)', 'Heater Fault in Room1(OFF)', 'Battery Fault in Temperature Sensor Room1', 'Offset Fault in CO2 Sensor Room1', 'Gain Fault in CO2 Sensor Room1' ,'Healthy Mode in Room1', 'Communication Fault(Routing)in Room1', 'Battery Fault in CO2 Sensor Room1'. The last step of data preprocessing is dividing the collected data into training data and testing data. The training data are used for training network and testing data are used for network evaluation.

C. Deep Neural Network Design

This section shows the steps to design the deep neural network. The basic steps of designing the deep neural network are presented in figure 2. The process starts by providing the training data with its labels to the training black box. The black box of training contains three main steps which are: network layers, network parameters and the network training.

a) Network Layers: There are several types of layers that are used for deep learning. The layers that have been used in our network are five layers as following: Input sequence layer that passes the time series data to the network. The second layer, LSTM layer, defines the subtype of the sequence layers, and learns long-term dependencies between time steps in time series and sequence data. Fully connected layer is the third layer after LSTM layer. The main task of this layer is multiplying the input by a weight matrix and then adding the bias vector. The output size of the last fully connected layer is equal to the number of classes of data set.

The fourth layer, softmax layer, is to calculate the probability of the predicted output. The last layer, classification output layer, provides the last output of the classified data.

b) Network Parameters: The next step after defining the neural network layers is to set up the training options of the network. By using the trainingOptions function, the global training parameters are defined. The optimizer used in

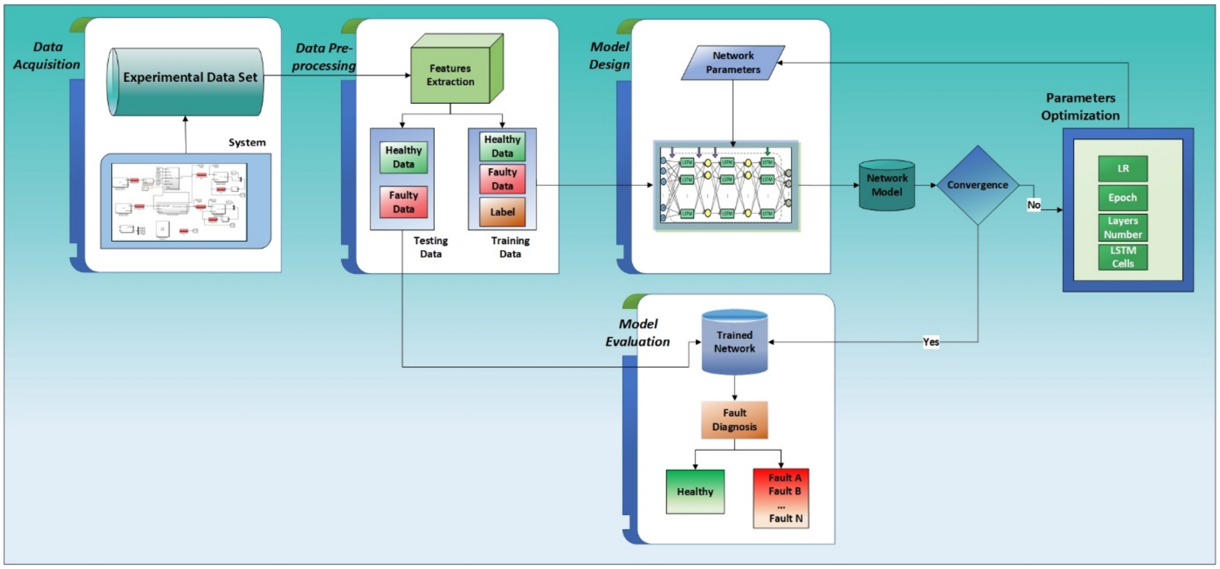

Fig. 1. A Pictorial Overview of Fault Detection and Diagnosis Approach based on Deep Neural Network.

TABLE I
A SAMPLE OF DATA ACQUISITION.

Time	Occupants	Lower Threshold	Stair_Temperature	Upper threshold	SecondFloor Temperature	Daily_Temperature	Faulty sensor reading(Gain)	Heater Status(On/Off)	Damper (Open/Close)	Real Room temperature	CO2 Value
0	0	17.5	13.5	22.5	20	7	60	0	0	20	400
1	0	17.5	13.5	22.5	20	7.00036361	59.97149871	0	0	19.99049957	400
2	0	17.5	13.5	22.5	20	7.000727221	59.94305964	0	0	19.98101988	400
3	0	17.5	13.5	22.5	20	7.001090831	59.9146825	0	0	19.97156083	400
4	0	17.5	13.5	22.5	20	7.001454441	59.88636701	0	0	19.96212234	400
5	0	17.5	13.5	22.5	20	7.001818051	59.85811285	0	0	19.95270428	400
6	0	17.5	13.5	22.5	20	7.002181661	59.82991974	0	0	19.94330658	400
7	0	17.5	13.5	22.5	20	7.002545272	59.45187648	0	1	19.81729216	400
8	0	17.5	13.5	22.5	20	7.002908882	59.07577071	0	1	19.69192357	400
9	0	17.5	13.5	22.5	20	7.003272492	58.70398974	0	1	19.56799658	400

this study is 'adam' which is derived from adaptive moment estimation. 'MiniBatchSize' is a subset of the data that is used by the solver to update the parameters in each step. Each parameter updated by the solvers is called an iteration, and the full pass through the entire data set is called an epoch.

By using the 'MaxEpochs', the maximum number of epochs is specified in order to train the network. 'InitialLearnRate' is one of the most essential parameters that effects the training process. The learning rate is determined by using the 'InitialLearnRate'. By default, trainNetwork uses this value

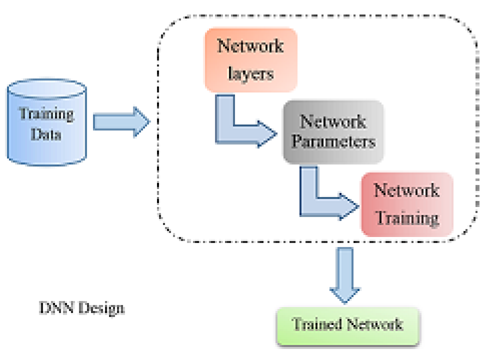

Fig. 2. A Pictorial Overview of Deep Neural Network Design.

throughout the entire training process. To perform the network training in a short time, trainNetwork uses GPU by default if it is available. Otherwise, trainNetwork uses a CPU. Alternatively, the execution environment (hardware resource) can be specified using the 'Execution Environment'. During the training of network, there is a possibility to monitor the training progress. It is useful to learn how well is the progress of training by monitoring the improvement of the network accuracy, and how the network fits the training data. The monitoring can be enabled by setting 'training-progress' as the 'Plots' variable in trainingOptions and running network training.

c) Network Training: Network training is the last step of network design. After specifying the layers of the network and the training parameters, network training runs using the training data.

III. RESULTS

In this section, the training results in optimization process and the network evaluation results based on statistical measures of the performance are discussed.

A. Deep Neural Network Optimization

Several parameters had been selected to be optimum according to their effect on the training process. The main training parameters that are selected are learning rate, number of hidden layers, number of hidden nodes, and the number of the epochs. Several network trainings for each parameter was performed separately and the results were saved. Based of the acquired accuracy respective to each parameter's values, the evaluation figures 3 to 6 were produced. The final deep neural network is established based on the optimum point of these parameters based on the figures.

a) Training Results of FDD Model with different Learning Rate: This study describes the value of learning rate in each training has a great influence on the performance of the network. In this study, instead of using a fixed learning rate throughout the training process, a larger learning rate can be set in the first training and gradually reduce this value to find the optimized point. Therefore, the network is trained based on a wide range of learning rates as the following values: 0.1, 0.05, 0.025, 0.01, 0.005, 0.001 and 0.0005. On the other hand, the initial settings of the other network parameters had been set to fixed value as the following: the number of the hidden units is set to 180, the number of output classes is set to 11, the mini batch size is set to 150 and the maximum epoch is set to 150. Figure 3 shows the accuracy results over the FDD model's training network parameter of learning rate. According to these results the high value of learning rate causes low accuracy rate of the network training, and when the learning rate value decreases, the performance accuracy increases gradually. The best accuracy for the FDD model is 93% at 0.005. With more reduction in learning rate parameter, the accuracy tends to decrease. However, with small learning rates, the training time increases because more time is needed to meet the convergence.

b) Training Results FDD Model with different Number of Hidden Layers: Figure 4 demonstrates the accuracy results over the FDD model's training network parameter of the number of hidden layers. The study shows the number of hidden layers is an important factor affecting on the training performance. It is considered also as the main difference between the traditional neural network and deep neural network. The FDD model is trained with the following parameters: For each training, the number of hidden layers varies and increases gradually with the values (3, 4, 5, 6, 7). The other parameters are the same in all training, these parameters are:

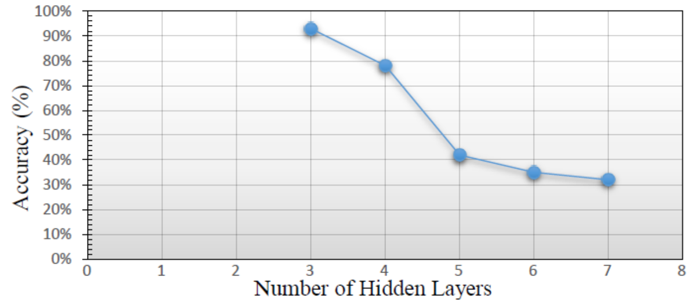

Fig. 4. FDD Model with Different Number of Hidden Layers.

number of output Classes is 11, Max number of Epochs is 200, Mini Batch Size is 150, and the learning rate is 0.001. According to the results in figure 4, it can be observed that in this case, the accuracy decreases gradually when increasing the depth of the network. The rate decreases from 94% with 3 hidden layers to be around 30% at 7 hidden layers besides increasing the required training time by increasing the layers. Therefore, three hidden layers is selected as a suitable depth of the designed network model.

c) Training Results of FDD Model with the different Number of Hidden Layer Units: Hidden layer nodes (or units) should be considered in the network training stage, because of its great influence on the training progress. In general, the number of nodes of the input layer is set based on the number of training data features, whereas the nodes of the output layer are set to be equal to the output classes. Figure 5 illustrates the accuracy of the FDD model's training network over different number of hidden nodes. The authors assumed multiple trainings with different hidden layer units ranging as the following: 25, 50, 100, 150, 200, 250, 300, 350, 400. The other experimental parameters in these setups are Number of classes of 11, Max Epochs of 250, Mini Batch Size of 150, and learning rate of 0.001. The results in figure 5 shows that the training accuracy is poor with small hidden nodes. Increasing the number of nodes improves the accuracy to exceed 80% with 50 nodes which seems not sufficient. By increasing the number of nodes, it is observed that the accuracy rate improves and reaches to the best value with 250 nodes (accuracy around 98%), but after this increase, the more nodes are not useful

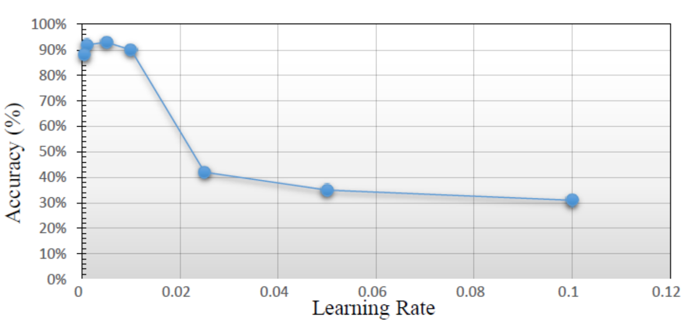

Fig. 3. FDD Model with Different Learning Rates.

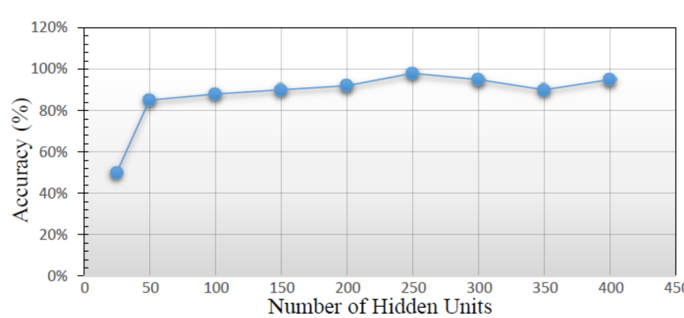

Fig. 5. FDD Model with Different Number of Hidden Units.

anymore. The number of nodes is related to training data samples. However, the training time increases by increasing the number of hidden nodes.

d) Training Results of FDD Model with the different Number of Epoch: The last parameter which was detected as an important factor in FDD model accuracy is the number of epochs. The analysis of this effect is done by taking the previous results into account. That means, the best parameters of the previous experiments are selected with different number of training epochs. The other setup parameters are the number of the hidden units of 200, the learning rate of 0.001, the output classes of 11 and mini-batch size of 150. Figure 6 describes accuracy of FDD model over different number of epoch. At the beginning, the epoch is 25 and then increases gradually in the next training. The results shows the effect of increasing the epoch number on the performance, where the accuracy is improved by increasing the epoch number to be around 85% with 150 epoch. When the number of epoch exceeds 250, the accuracy of the FDD model improved to reach around 95%. The best observed value of the accuracy is 98% with 490 epochs.

B. Optimized FDD Model Evaluation

The optimized parameters selected to train the final network model are as following: Feature Dimension of 4, Number of Hidden Units of 255, Number of Classes of 11, Max Epochs of 468, mini Batch Size of 128, learning rate of 0.001. After training the proposed deep neural network, the model gives the predicted output as the results. Figure 7 shows a sample of model prediction result for the case of battery fault as an example.

In this section first performance statistical measures are discussed, then the evaluation results are presented. The evaluation strategy is used to evaluate the proposed deep neural network based on different types of faults to find out the designed model is successful to which extend.

Performance statistical measures are defined based on various rates of true positive (TP) that are the relevant (target) samples predicted correctly, true negative (TN) that are the irrelevant samples, false positive (FP) that are the irrelevant samples predicted incorrectly as the target, false negative (FN) that are the relevant (target) samples which are not predicted at

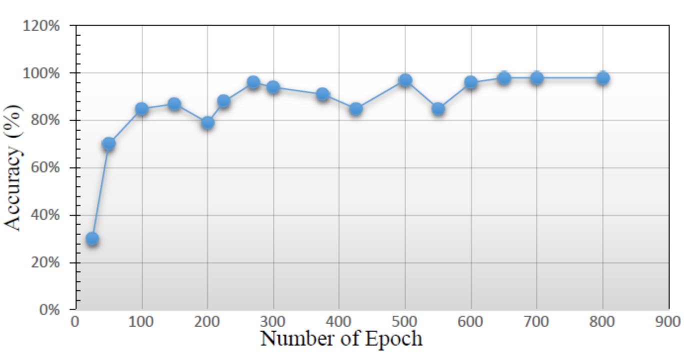

Fig. 6. FDD Model with Different Number of Epochs.

Healthy Data

Fault Injection Instance

Battery Fault

	1	2	3	4
36589	21.0432	23	0	600.2640
36590	21.0489	23	0	600.4486
36591	21.0545	23	0	600.6332
36592	21.0602	23	0	600.8178
36593	21.0658	23	0	601.0024
36594	21.0776	23	0	626.9885
36595	21.0776	23	0	627.0346
36596	21.0776	23	0	627.0808
36597	21.0776	23	0	627.1269
36598	21.0776	23	0	627.1731
36599	21.0776	23	0	627.2192
36600	21.0776	23	0	627.2654
36601	21.0776	23	0	627.3115
36602	21.0776	23	0	627.3577
36603	21.0776	23	0	627.4038
36604	21.0776	23	0	627.4500
36605	21.0776	23	0	627.4961
36606	21.0776	23	0	627.5423
36607	21.0776	23	0	627.5884
36608	21.0776	23	0	627.6346

a) Test Data (Battery Fault)

	1
36589	Healthy Mode in Room1
36590	Healthy Mode in Room1
36591	Healthy Mode in Room1
36592	Healthy Mode in Room1
36593	Healthy Mode in Room1
36594	Battery Fault in Temperature Sensor Room1
36595	Battery Fault in Temperature Sensor Room1
36596	Battery Fault in Temperature Sensor Room1
36597	Battery Fault in Temperature Sensor Room1
36598	Offset Fault in Temperature Sensor Room1
36599	Battery Fault in Temperature Sensor Room1
36600	Battery Fault in Temperature Sensor Room1
36601	Battery Fault in Temperature Sensor Room1
36602	Healthy Mode in Room1
36603	Offset Fault in Temperature Sensor Room1
36604	Battery Fault in Temperature Sensor Room1
36605	Battery Fault in Temperature Sensor Room1
36606	Battery Fault in Temperature Sensor Room1
36607	Healthy Mode in Room1
36608	Battery Fault in Temperature Sensor Room1

b) Predicted Labels Output of Model

Fig. 7. Test Data and Predicted Labels Output of Designed Model.

all, recall, precision, and F-measure which describe the quality of the classification [12].

The confusion matrix [12] respective to the designed FDD network model in this study is shown in figure 8. In this matrix the HM, BF (CO2), RF, DF (OFF), GF (CO2), OF (T), HF (OFF), OF (CO2), DF (ON), GF (T), and BF (T) stand for Healthy Mode, Battery Fault in CO2 sensor, Routing Fault, Damper Fault with value 0, Gain Fault in CO2 sensor, Offset Fault in temperature sensor, Heater Fault in off mode with value 0, Offset Fault in CO2 sensor, Damper Fault in open position with value 1, Gain Fault in temperature sensor, amd Battery Fault in temperature sensor, respectively. The overall accuracy of the model based on all classes is 97.4%. This value is considered as a perfect result of fault detection and diagnosis in such kind of mechatronics systems using the deep neural network, particularly LSTM model.

Predicted Output \ Target Output	HM	BF CO_2	RF	DF OFF	GF CO_2	OF	HF OFF	OF CO_2	DF ON	GF	BF	
HM	202629 38.1%	395 0.1%	1293 0.2%	1 0.0%	0 0.0%	3 0.0%	24 0.0%	1 0.0%	394 0.1%	1 0.0%	135 0.0%	98.9% 1.1%
BF CO_2	1884 0.4%	31605 5.9%	796 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	430 0.1%	91.0% 9.0%
RF	1 0.0%	0 0.0%	29911 5.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100.0% 0.0%
DF OFF	908 0.2%	0 0.0%	0 0.0%	31998 6.0%	4 0.0%	0 0.0%	0 0.0%	12 0.0%	0 0.0%	0 0.0%	0 0.0%	97.2% 2.8%
GF CO_2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	31995 6.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
OF	14 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	31950 6.0%	994 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	96.9% 3.1%
HF OFF	333 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	30981 5.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.9% 1.1%
OF CO_2	2441 0.5%	0 0.0%	0 0.0%	1 0.0%	1 0.0%	1 0.0%	1 0.0%	31987 6.0%	40 0.0%	1 0.0%	48 0.0%	92.7% 7.3%
DF ON	45 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	31566 5.9%	0 0.0%	0 0.0%	99.9% 0.1%
GF	22 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	46 0.0%	0 0.0%	0 0.0%	0 0.0%	31998 6.0%	75 0.0%	99.6% 0.4%
BF	3722 0.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	31313 5.9%	89.4% 10.6%
	95.6% 4.4%	98.8% 1.2%	93.5% 6.5%	100.0% 0.0%	100.0% 0.0%	99.8% 0.2%	96.8% 3.2%	100.0% 0.0%	98.6% 1.4%	100.0% 0.0%	97.9% 2.1%	97.4% 2.6%

Fig. 8. Confusion Matrix of Designed FDD Model Network.

Figure 9 shows the rates of recall and precision in fault detection and diagnosis where the lowest rate of the precision is 89% for the battery fault diagnosis in the temperature sensor node. This precision rate indicates the rate of the predicted samples correctly or a successful diagnosis among all of the predicted data, for example. That means, 89% of the predicted data for the case of battery fault in temperature sensor are correctly diagnosed. On the other hand, the lowest recall value is 93.5% for routing fault diagnosis, and this indicates the sensitivity of the model to detect the relevant samples of routing fault among target data set. In other words, the model has an ability to detect 93.5% of the target data correctly (low false negative). Figure 10 presents the F-score values of each class as a result of recall and precision rates of each class.

Conclusion

In this study, a deep learning application for the goal of fault detection and diagnosis of mechatronics systems for a modeled demand-controlled ventilation and heating system based on wireless sensor and actuator networks in MATLAB/Simulink is studied. For this, a novel model is developed based on a deep neural network called LSTM. All of the model developing steps had been described. The proposed FDD model had been evaluated based on testing data which include data of ten different types of faults such as gain and offset faults in temperature and CO2 concentration sensor nodes and stuck-at faults in damper and heater actuator nodes. Another evaluated fault type is the communication fault, such as battery fault and routing algorithm fault. The results of the evaluation show that the optimized fault diagnosis model is strong enough to correctly detect and diagnose all the types of faults with a high accuracy based on the evaluation parameters discussed in the paper. The highest accuracy percentage based on F-score measure is assigned to the gain fault detection in CO2 sensor with 100%. The lowest accuracy percentage is assigned to battery fault detection in temperature sensor with 93.45%. The results also introduce the precision and recall percentage of each class. Totally, the accuracy of the FDD model is 97.4%. This value is considered as a perfect result of fault detection and diagnosis in these kind of mechatronics system based on a deep neural network (LSTM method).

Acknowledgment

This work was supported by the German research funding organization (DFG) with research grant number OB384/5-1.

References

[1] M. J. Brandenmuehl and J. E. Braun, "The impact of demand controlled and economizer ventilation strategies on energy use in buildings," ASHRAE Transactions, 1999.

[2] A. Behravan, R. Obermaisser, A. Nasari, "Thermal dynamic modeling and simulation of a heating system for a multi-zone office building equipped with demand controlled ventilation using MATLAB/Simulink," International conference on circuits, system and simulation, London, United Kingdom, July 2017, Conference proceeding pp. 103-108. IEEE.

[3] A. Behravan, R. Obermaisser, M. Abboush, "Fault Injection Framework for Demand-Controlled Ventilation and Heating Systems Based on Wireless Sensor and Actuator Networks", The 9th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON 2018) "BEST PAPER AWARD", University of British Columbia, Vancouver, Canada, 2018.

[4] J. Schein, S. T. Bushby, N. S. Castro, J. M. House, A rule-based fault detection method for air handling units, Energy and Buildings, Vol. 38, 2006, pp. 1485-1492.

[5] K. Bruton, P. Raftery, N. Aughney, M. M. Keane, D. O'Sullivan, "Development of an automated fault detection and diagnosis tool for AHUs," Proceedings of the 12th international conference for enhanced building operations, Manchester, UK, October 2012.

[6] Lee, S. H., Yik, F. W. H. 2009. "A study on the energy penalty of various air-side system faults in buildings," Energy and Buildings, January 2010, pp. 2-10.

[7] Katimapula, S., Brambley, M. R. "Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems—A Review, Part I, HVAC & R Research," Volume 11, Issue 1, 2005, pp. 3 – 25.

[8] Q. Zhou, S. Wang, Z. Ma, "A model-based fault detection and diagnosis strategy for HVAC systems," International journal of energy research, Vol. 33, February 2009, pp. 903-918.

[9] Min Hua, Huanxin Chen, Limei Shen, Guannan Li , Yabin Guo, Haorong Li , Jiong Li , Wenju Hu," A machine learning bayesian network for refrigerant charge faults of variable refrigerant flow air conditioning system", Energy and Buildings 158 (2018) 668–676.

[10] D.Lee, B. Lee, J. Shin," Fault Detection and Diagnosis with Modelica Language using Deep Belief Network", Proceedings of the 11th International Modelica Conference September 21-23, 2015, Versailles, France.

[11] Y. Guo, Z. Tan, H. Chen, G. Li, J. Wanga, R. Huanga, J. Liua, T. Ahmad," Deep learning-based fault diagnosis of variable refrigerant flow air conditioning system for building energy saving" Contents lists available at ScienceDirect,Applied Energy 225 (2018) 732–745.

[12] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel," PERFORMANCE MEASURES FOR INFORMATION EXTRACTION" BBN Technologies, GTE Corp. Cambridge, MA 02138.

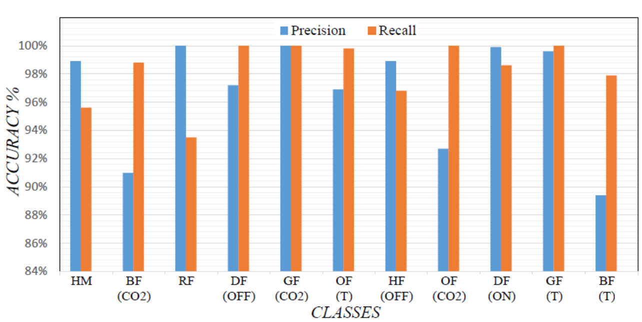

Fig. 9. FDD Evaluation Result Accuracy for Different Fault Types based on Precision and Recall Parameters.

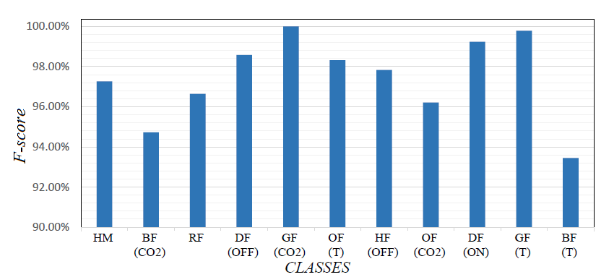

Fig. 10. FDD Evaluation Result Accuracy for Different Fault Types based on F-Score Parameter.