Transfer Learning with Domain Adaptation for Unlabelled Sensor Faulty Data Classification

1st Md. Nazmul Hasan
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, South Korea
hasan01@mail.ulsan.ac.kr

2nd Rafia Nishat Toma
Electronics and Communication Engineering Discipline
Khulna University
Khulna, Bangladesh
rafiatoma.eceku@gmail.com

3rd Sana Ullah Jan
School of Computing, Engineering and Built Environment
Edinburgh Napier University
Edinburgh, U.K.
s.jan@napier.ac.uk

4th Insoo Koo
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, South Korea
iskoo@ulsan.ac.kr

***Abstract*—In the data analytics based IoT era, reliable sensor data is essential for maintaining the integrity and effectiveness of IoT applications. The reliability of sensor data can be compromised due to faults that that occur in the sensors and this faulty sensor signals can lead to performance degradation in IoT enabled systems. This study investigates a transfer learning method augmented by a domain adaptation technique for sensor fault classification. The domain adaptation technique allows the classifier model to accurately identify two types of sensor faults, drift and bias fault, in a target sensor node with an unlabeled dataset. We trained an ANN classifier using a semi-supervised approach, leveraging labeled samples from the source sensor node alongside unlabeled data from the target sensor. In the current work, both the source and target sensors are of the same type but are located in different locations. Results from two experiments demonstrate that our approach achieves approximately 97% accuracy when the target sensor is indoor and the source sensor is outdoor. When both sensors are located outdoors, the accuracy exceeds 99%. Other classification metrics also remain consistent. This approach not only maximizes the use of available data but also offers a robust solution for scenarios where obtaining labeled data is difficult or missing.**

***Index Terms*—sensor faults, domain adaptation, IoT, deep learning, transfer learning**

I. Introduction

In the thriving landscape of the Internet of Things (IoT), sensor-collected data is the most significant element for a wide array of applications, ranging from smart homes to industrial automation [1], [2]. The significance of this data lies in its ability to provide real-time insights, enhance operational efficiencies, and drive predictive analytics, which are critical for the seamless functioning of the overall IoT system. However, the reliability of IoT applications is profoundly impacted by the quality of sensor data. Imperfect sensors, which may produce faulty data, pose a significant challenge. Faulty sensor data can lead to incorrect decision-making, reduced system performance, and even pose safety risks in critical applications such as healthcare and autonomous vehicles. Typically, the sensors used for collecting data are inexpensive, small in size, and consume low power. In real scenarios, these sensors are often installed in harsh and extreme locations in industrial and outdoor environments, which increases the probability of damage. Additionally, aging, hardware malfunctions, and inaccurate installations can also make a sensor prone to producing abnormal or faulty signals. Sensor faults can be classified primarily in two categories: incipient faults and abrupt faults [3]. Incipient faults, such as bias and drift faults, develop slowly and are difficult to detect in their initial stages. On the other hand, abrupt faults like stuck and spike faults are sudden events that persist for a short amount of time.

A large number of networked sensors nowadays provide a huge amount of data. With the ease of implementing machine learning and deep learning models, the data-driven sensor fault analysis approach is now widely adopted by researchers. In this paper, we implemented an artificial neural network model for classifying drift and bias faults in a sensor which might have non-labelled samples. In real-world scenarios, it might happen that one of the sensors in a network has missing labels for its data. However, if there are similar types of sensors in the same network with labeled fault data, it is possible to effectively classify faults for the unlabeled sensor data by training a neural network that incorporates both the labeled and unlabeled data for training as discussed in [4], [5]. In this paper, we consider a source sensor with labeled data and a target sensor with unlabeled data. An Artificial Neural Network (ANN) is trained with a custom loss function to ensure good sensor fault classification performance for both the source and target sensors. Both sensors are of the same type and measure the same parameter, temperature. However, two different experiments are performed. In the first case, the source sensor is located outdoors and the target sensor is located indoors. In the second case, both sensors are located outdoors.

II. Related Works

In the data-driven techniques of sensor fault classification, the machine learning and deep learning based approaches are mainly followed by the research community. The machine learning algorithms generally classifies faults based on features extracted from the original sensor recordings as demonstrated in [6], where time domain features are utilized with support vector machine (SVM). SVM is also in [7] with empirical mode decomposition (EMD) extracted features. Various popular machine learning algorithms, such as K-nearest neighbors (KNN), random forest (RF) classifiers, Gaussian Naive Bayes (GNB), and multilayer perceptron (MLP), have been employed in tasks related to sensor fault classification and analysis [8], [9].

The convolutional neural network (CNN) is widely being used for sensor fault classification and analysis. Four type of sensor faults are classified using CNN in [10] in a structural monitoring system, where the convolution is performed on a stacked array of signals created from multiple sensor data. Analysis of similar types of faults in a automated car is conducted in [11], where authors suggested a multistage approach for fault detection, classification, and isolation. A CNN 1D architecture is suggested for fault detection and at the later stage MLP is utilized for fault classification from several time-frequency features. Few researchers leveraged the strength of 2D CNN in classifying sensor fault by converting the sensor signal in to images through various approaches. Authors in [12] converted sensor signal into gray matrix image to detect seven fault types in a hydrogen sensor using CNN architecture. Another CNN based sensor fault classifier is studied in [13], where Continuous Wavelet (CWT) scalograms of seven types of sensor faults in a aeroengine control system is used as the input of the model. Autoencoder (AE) architectures are also used in detecting multiple sensor fault in real sensor as presented in [14]. Hybrid architectures like the integration of CNN and LSTM also demonstrated good performance in detecting bias, drift, and random faults as reported in [15]. Apart from CNN and ANN models, recently advanced architectures such as generative adversarial networks have also been used in sensor fault analysis when there is an imbalanced data distribution [16], [17].

It is observed that the majority of works in sensor fault analysis fall under the category of supervised learning. In this work, the concept of transfer learning approach is used along with domain adaptation for sensor fault classification in a target sensor having non-labeled data. The classifier model is trained using the labeled source sensor data and the target sensor data. Considering the nature of the data used here, this proposed approach can be categorized as semi-supervised approach.

III. Methodology

In this paper a publicly available sensor dataset [18] is used. The dataset contains temperature and humidity data collected by TinyOS-based Crossbow TelosB motes in both single-hop and multi-hop settings. The temperature and humidity readings are collected at an interval of 5 seconds over 6 hours. An elaborate discussion on the dataset can be found in [19].

In our present work, temperature signal of three sensors are considered to demonstrate the sensor fault classification approach. In our proposed approach, we consider two sensors: one is referred to as the source sensor and the other is target sensor. Based on the location of the source and the target sensor, two different scenarios are realized. In the first case, the sensor located outside is considered as the source sensor and the target sensor is assumed to be located in side of a room and in the other case, both the sensors are located outside of a house but in separate locations . It is also assumed that the source sensor dataset has corresponding labels against the samples whereas the target sensor data have no label assigned to its samples.In our work, we train a single ANN model based on the labeled source sensor dataset and non-labeled target sensor dataset. Finally, the trained model is tested to classify potential sensor faults on the test set of the target sensor. This approach can be beneficial because without a labelled dataset it is possible to classify fault in the target sensor node. A graphical visualization of the proposed approach is presented in Fig. 1.

Although the source and the target sensor collects the same parameter, temperature in this case, both the sensors are located in different physical locations which might lead to different data distributions for the sensors. Therefore, while using the classifier model to perform classification on target sensor data, the model should adapt the difference between the two dataset from the source and the target. To achieve this a custom loss function is used during the training phase.

A. Dataset Preparation

In the context of sensor fault classification tasks, the majority of research papers create fault datasets by introducing synthetic fault samples using defined equations for different types of faults [9], [11], [20]. Similarly, in this work, we created a fault dataset by altering normal sensor readings using mathematical equations that represent certain types of faults. Specifically, we considered two types of sensor faults: drift fault and bias fault. Temperature readings from two outdoor sensors and one indoor sensor are considered in this task. As the first step in data preparation, each of the sensor readings are segmented in non-overlapping vector of 100 data points which is equivalent to about 8.33 seconds reading by the sensor. Each of the segments are considered as samples of non faulty condition. This process of segmentation provides multiple samples from the complete duration of the sensor recordings. However, for deep learning based fault classification a dataset with sufficient number of samples are necessary. Therefore to increase the number of non-faulty samples, zero mean, low variance random signal of same length of the segmented vector is added to the primarily created samples. The steps listed in Algorithm 1 results a dataset, S_j contains synthetic samples created from the available non-faulty instances and $\mathcal{H}_n(0, \mu)$ represents the zero mean random signal

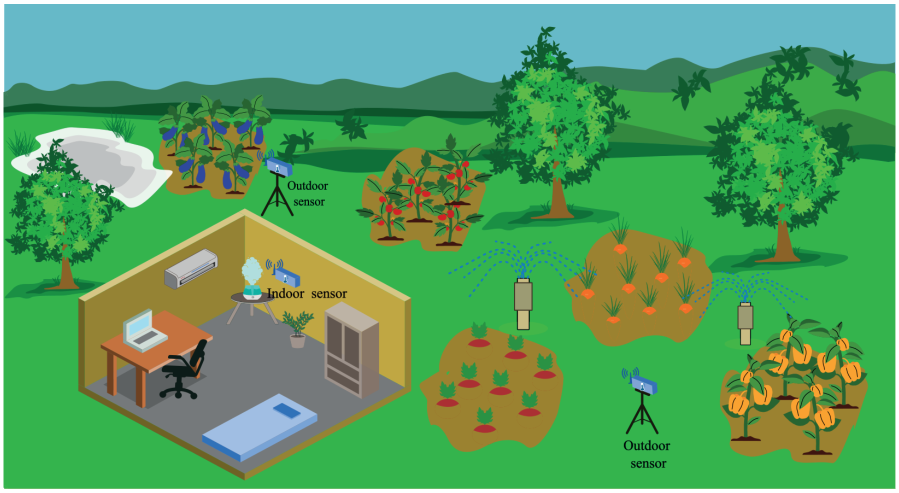


Fig. 1: Visual depiction of the sensor distribution scenario.

Algorithm 1 Creating synthetic dataset.

Input: S_i : Matrix represents the segmented samples from the sensor reading, M : The number of additional samples
Output: S_j : Dataset with increased samples
1: Initialize model parameters, θ
2: **for** j $= 0$ to M **do**
3: Choose a random instance from $S_i \rightarrow S_m$
4: Add zero mean, low variance random signal with S_m
5: $S_j = S_m + \mathcal{H}_n(0, \mu)$
6: **end for**
7: Dataset with increased samples, S_j

vector. Finally, concatenating S_i and S_j provides the complete non-faulty dataset.

The final step in creating the complete dataset is to create artificial sensor fault samples from non-faulty samples. As mentioned earlier, the drift fault and bias fault samples are considered in this work, the artificial fault samples are generated using the following mathematical equations of the faults. In drift fault case, the sensor reading linearly changes over time possibly due to external factors or change in circuit parameters. If S_n denotes the normal signal then the drift fault can be realized with (1).

$$S_{drift} = S_n + \mathcal{H}_n\left(0, \mu\right) + \beta_n \tag{1}$$

Here, β_n represents the drift parameter that introduces the linear change in the sensor recording.

When the sensor output shifts to a value higher than a regular value, the sensor is said to be affected with bias fault and this can be mathematically represented as (2).

$$S_{bias} = S_n + \mathcal{H}_n\left(0, \mu\right) + v \tag{2}$$

Here, v represents the constant bias term added to the normal signal values. Using the same steps individual datasets for each of the three sensors are created for implementing the fault classification scheme using an ANN model.

B. Training the Artificial Neural Network Classifier

As described before, in this approach we focus on developing a ANN sensor fault classifier which can be used effectively to classify the fault in a target sensor having a non-labelled dataset. To achieve this a custom loss function is used during the training of the model. The source and the target sensor in this work are considered to be similar type but they located in different locations as depicted in Fig.1. The custom loss function reduces the domain difference between two different datasets of source and target sensors. The custom loss function constitutes of two parts. The first part accounts for the labeled dataset of the source sensor, the loss is the common cross-entropy loss used for classification problem and defined as (3).

$$L_s\left(x_i^s, y_i^s\right) = -\sum_{i=1}^{n_j^s}\sum_{k=1}^{N} y_{i,k}^s \; logp\left(y = k \mid x_i^s\right) \tag{3}$$

where, N is the number of classes, and $y_{i,k}^s$ takes a value 1 if the actual label of the $i-th$ sample is class k or 0 otherwise, and $p\left(y = k \mid x_i^s\right)$ denotes the model output. The second part of the loss function provides a measurement of the domain difference between the source and the target sensor datasets which is defined as (4).

$$L_T\left(x_i^S, x_j^T\right) = \left\| \frac{1}{\mathcal{B}^s}\sum_{i=1}^{\mathcal{B}^s}\mathcal{M}\left(x_i^s\right) - \frac{1}{\mathcal{B}^T}\sum_{i=1}^{\mathcal{B}^T}\mathcal{M}\left(x_j^T\right)\right\|_2 \tag{4}$$

where, x_i^S and x_j^T represent the source and target sensor dataset, $\mathcal{B}$ denotes the batch size for each dataset, and $\mathcal{M}\left(\cdot\right)$ is the feature obtained from the last fully connected layer of the ANN architecture. This part of the loss function computes the Euclidean norm of the features returned by last fully connected layer of the model for the source and the target sensor dataset. The features are averaged over the number of samples in a single batch. Since, it is assumed that the target dataset has no labels associated with it, for that reason the features from

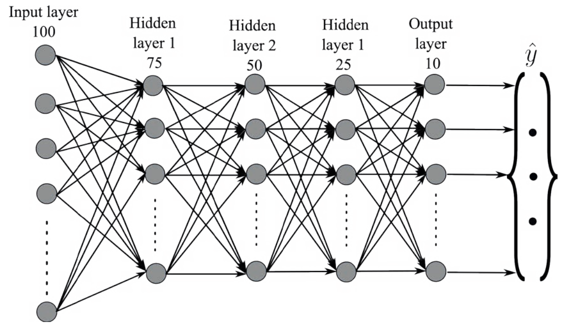


Fig. 2: Multilayer ANN model.

the last fully-connected layer is considered in this part of the loss function. Therefore the complete loss function defined in (5) is the summation of equations (3) and (4),

$$L_L = L_S + L_T \tag{5}$$

The model considered in this paper is an ANN consisting of four hidden layers as depicted in Fig. 2 indicating the number of nodes in each layer. Rectified Linear Unit (ReLU) is the activation function and categorical cross-entropy loss is considered as the loss function. After training the ANN classifier model, the classification performance is tested on the target sensor. In the testing phase the actual labels of the target sensor dataset is included to assess the performance of the model. The algorithm for training the ANN is briefly presented in Algorithm 2.

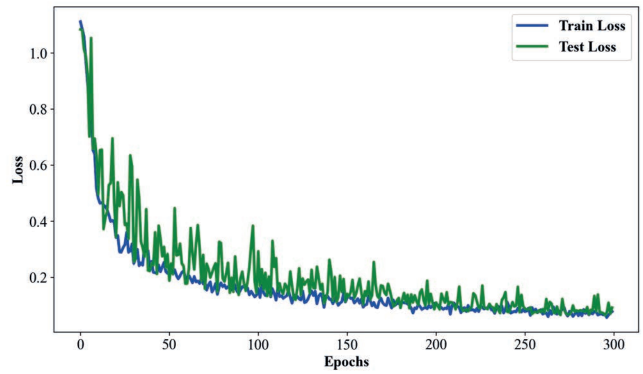
Algorithm 2 Training the classifier model.

Input: Source dataset $\left(X^S, y^S\right)$, target dataset $\left(X^T, y^T\right)$, learning rate α, number of epochs E
Output: Trained model parameters, θ

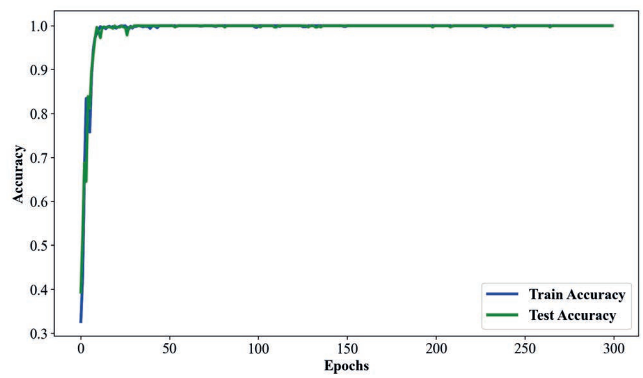
```
Initialize model parameters, θ
for epoch = 1 to E do
    for each batch (X^S_batch, y^S_batch), (X^T_batch) do
        Compute predictions, ŷ = f(X^S_batch, θ)
        Compute features, M(X^S_batch) and M(X^T_batch)
        Compute loss, L_L = L_S + L_T
        Calculate gradients, ∇_θ L_L
        Update model parameters, θ ← θ − α∇_θ L_L
    end for
end for
Return trained model parameters, θ
```

IV. Results

In this section, we present the classification reports generated from the experiments. These reports encompass common performance indicators such as precision, recall, F1-score, and accuracy, providing a measure of the model's effectiveness. Also the model's training and testing performance over the training epochs are provided using the loss and accuracy plots which will provide a better understanding of the model's learning process and its generalization capabilities. The ANN



(a) Training vs. testing loss per epoch



(b) Training vs. testing accuracy per epoch

Fig. 3: Accuracy and loss comparison over training epochs with outdoor source sensors and target indoor sensor.

model is trained together with the labeled dataset from the source sensor and the unlabeled samples of the target sensor. In both the cases the dataset consists of three classes: no-fault, drift fault, and bias fault. The dataset is partitioned in 80:20 ratio for the training and testing purpose and standardized using the standard scalar method. Initially, one outdoor sensor is considered as the source sensor and the indoor sensor is considered as the target sensor. The model is trained for 300 epochs with Adam optimization technique setting the learning rate at 0.001. The dataset is partitioned at an 80:20 ratio into training and testing sets.

The loss and accuracy patterns for training and testing over the training epochs are shown in Fig. 3. The losses on the two sets seem to decrease gradually over the epochs, as shown in Fig. 3a. The decrease in loss is a little faster up to the 100^{th} epoch, then the rate of decrease becomes smaller and eventually settles around a value of 0.07. The variation in the test loss is a bit higher compared to the training loss; however, the variations diminish as the training progresses. In the evaluation of the fault classifier model on the test set data, an accuracy of 97.24% is obtained. While the accuracy reflects the overall performance, the precision and recall values highlight the model's effectiveness in correctly identifying positive instances, which in this case are 97.43% and 97.24%

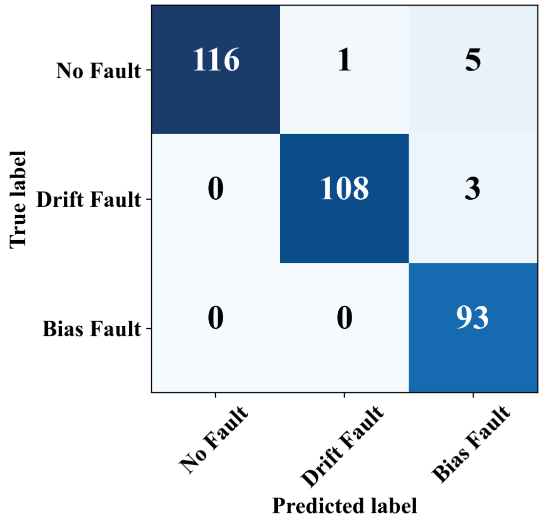
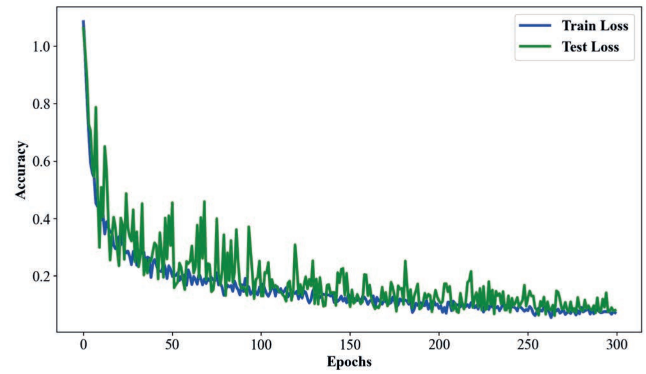


Fig. 4: Confusion matrix with outdoor source sensor and indoor target sensor.

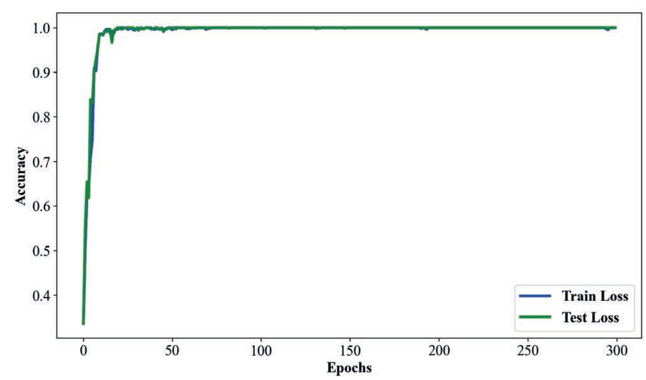
respectively. The reported f1-score value of 97.26% indicates the model's balanced performance in sensor fault classification. The confusion matrix presented in Fig. 4 provides a detailed breakdown of the model's performance across three classes. The matrix reveals that the model achieves high accuracy in identifying "No Fault" and "Bias Fault" instances, with 116 and 93 correctly classified samples, respectively. Misclassification rates are minimal, with only 5 "No Fault" instances misclassified as "Bias Fault," and 3 "Drift Fault" instances misclassified as "Bias Fault." Importantly, no "Bias Fault" instances were misclassified as "Drift Fault," indicating the model's robustness in differentiating between these specific classes.

In the second case, one of the outdoor sensor is considered as the source sensor and the other outdoor sensor is treated as a target sensor. The loss and accuracy pattern for training and testing over the training epochs are depicted in Fig. 5. As illustrated in Fig. 3, the losses on both the cases exhibits a gradual decrease over the epochs. Similar to the formal case the decrease in loss is more pronounced until 100^{th} epoch, after that it decreases steadily and attains a minimum loss around 0.07. While evaluating the model on the target sensor's test dataset a slightly higher accuracy of about 99.68% is attained, which indicates the model's superior performance in classifying unlabeled instances when both the target and source sensors are located outside. The precision, recall, and f1-score are also better marginally as 99.70%, 99.69%, and 99.70% respectively. The classification performance indicators for both scenarios are presented in Table. I.

The numbers reported in the confusion matrix shown in Fig.6 also indicate that the model performs better in classifying faults when both the source and target sensors are located outside, as there is only one sample misclassified as a drift fault. Domain adaptation-based transfer learning for sensor fault classification has recently gained attention. While several



(a) Training vs. testing loss per epoch.



(b) Training vs. testing accuracy per epoch.

Fig. 5: Accuracy and loss comparison over training epochs with outdoor source sensor and outdoor target sensor.

TABLE I: Comparison of the performance metrics.

Source sensor location	Target sensor location	Precision	Recall	F1-score	Accuracy
Outdoor	Indoor	97.43%	97.24%	97.26%	97.24%
Outdoor	Outdoor	99.70%	99.69%	99.70%	99.68%

studies have explored this approach with different datasets, comparing their results with ours using the WSN dataset is challenging. However, when comparing classification accuracy from papers that used the same dataset without domain adaptation, our proposed approach shows comparable performance, as shown in Table. II

V. Conclusion

In this study, we explored a transfer learning method assisted by a domain adaptation technique to classify sensor faults. The domain adaptation technique enables the classifier model to effectively classify two types of sensor faults in a target sensor node with a non-labeled sensor fault dataset. The ANN classifier is trained using a semi-supervised approach, utilizing labeled samples from the source sensor node and non-labeled data from the target sensor.

The results obtained from two experiments, conducted at two different locations of the target sensor, show that the

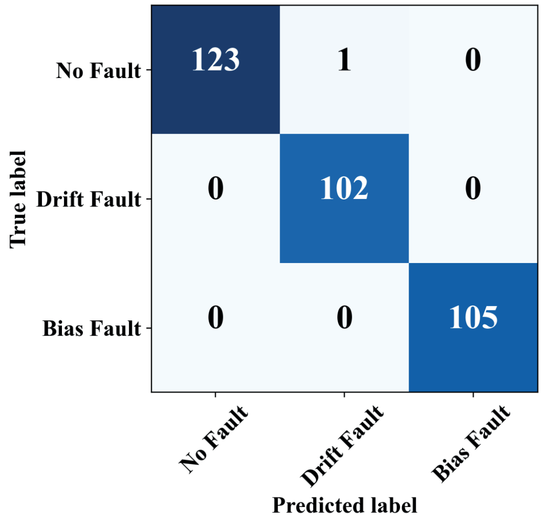


Fig. 6: Confusion matrix with outdoor source sensor and outdoor target sensor.

TABLE II: Comparison with existing works.

Reference	Model	Faults considered	Accuracy
[9]	ExtraTree	Drift, bias, stuck, spike, erratic, dataloss, random	81%
[21]	ExtraTree	Drift, bias, stuck, spike, erratic, dataloss, random	90%
[20]	MLP	Drift, bias	89.8%, and 86%
[22]	DNN	Drift, bias	99.6%
This paper	ANN with domain adaptation	Drift, bias, stuck	97.24% to 99.68%

proposed approach provides impressive classification performance. When an indoor sensor is considered as the target sensor and an outdoor sensor acts as the source sensor, approximately 97% accuracy is achieved. Conversely, when both the target and source sensors are located outdoors at different locations, more than 99% accuracy is reported. The values of other classification metrics for both experiments are also very consistent, ensuring the effectiveness of the model in classifying different classes in the fault dataset. The overall results imply that this approach can be an effective solution for missing label datasets in IoT sensor fault classification tasks.

In future work, we plan to include more sensor nodes and other types of faults into consideration. Additionally, we will analyze the classification performance by varying the location and the number of source and target sensor nodes.

References

[1] S. C. Mukhopadhyay, S. K. S. Tyagi, N. K. Suryadevara, V. Piuri, F. Scotti, and S. Zeadally, "Artificial intelligence-based sensors for next generation iot applications: A review," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 24920–24932, 2021.

[2] L. Babangida, T. Perumal, N. Mustapha, and R. Yaakob, "Internet of things (iot) based activity recognition strategies in smart homes: A review," *IEEE Sensors Journal*, vol. 22, no. 9, pp. 8327–8336, 2022.

[3] D. Li, Y. Wang, J. Wang, C. Wang, and Y. Duan, "Recent advances in sensor fault diagnosis: A review," *Sensors and Actuators A: Physical*, vol. 309, p. 111990, 2020.

[4] J. Chen, J. Li, R. Huang, K. Yue, Z. Chen, and W. Li, "Federated learning for bearing fault diagnosis with dynamic weighted averaging," in *2021 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD)*, pp. 1–6, IEEE, 2021.

[5] J. Shao, Z. Huang, and J. Zhu, "Transfer learning method based on adversarial domain adaption for bearing fault diagnosis," *IEEE Access*, vol. 8, pp. 119421–119430, 2020.

[6] S. U. Jan, Y.-D. Lee, J. Shin, and I. Koo, "Sensor fault classification based on support vector machine and statistical time-domain features," *IEEE Access*, vol. 5, pp. 8682–8690, 2017.

[7] D. Peng, S. Yun, D. Yin, B. Shen, C. Xu, and H. Zhang, "A sensor fault diagnosis method for gas turbine control system based on emd and svm," in *2021 Power System and Green Energy Conference (PSGEC)*, pp. 682–686, IEEE, 2021.

[8] A. Naimi, J. Deng, S. Shimjith, and A. J. Arul, "Fault detection and isolation of a pressurized water reactor based on neural network and k-nearest neighbor," *IEEE Access*, vol. 10, pp. 17113–17121, 2022.

[9] U. Saeed, S. U. Jan, Y.-D. Lee, and I. Koo, "Fault diagnosis based on extremely randomized trees in wireless sensor networks," *Reliability engineering & system safety*, vol. 205, p. 107284, 2021.

[10] D. Jana, J. Patil, S. Herkal, S. Nagarajaiah, and L. Duenas-Osorio, "Cnn and convolutional autoencoder (cae) based real-time sensor fault detection, localization, and correction," *Mechanical Systems and Signal Processing*, vol. 169, p. 108723, 2022.

[11] S. Safavi, M. A. Safavi, H. Hamid, and S. Fallah, "Multi-sensor fault detection, identification, isolation and health forecasting for autonomous vehicles," *Sensors*, vol. 21, no. 7, p. 2547, 2021.

[12] Y. Sun, H. Zhang, T. Zhao, Z. Zou, B. Shen, and L. Yang, "A new convolutional neural network with random forest method for hydrogen sensor fault diagnosis," *IEEE Access*, vol. 8, pp. 85421–85430, 2020.

[13] L. Gou, H. Li, H. Zheng, H. Li, and X. Pei, "Aeroengine control system sensor fault diagnosis based on cwt and cnn," *Mathematical Problems in Engineering*, vol. 2020, no. 1, p. 5357146, 2020.

[14] L. M. Ghinea, M. Miron, and M. Barbu, "Semi-supervised anomaly detection of dissolved oxygen sensor in wastewater treatment plants," *Sensors*, vol. 23, no. 19, p. 8022, 2023.

[15] A. M. Seba, K. A. Gemeda, and P. J. Ramulu, "Prediction and classification of iot sensor faults using hybrid deep learning model," *Discover Applied Sciences*, vol. 6, no. 1, p. 9, 2024.

[16] Y. Sun, T. Zhao, Z. Zou, Y. Chen, and H. Zhang, "Imbalanced data fault diagnosis of hydrogen sensors using deep convolutional generative adversarial network with convolutional neural network," *Review of Scientific Instruments*, vol. 92, no. 9, 2021.

[17] M. N. Hasan, S. U. Jan, and I. Koo, "Wasserstein gan-based digital twin inspired model for early drift fault detection in wireless sensor networks," *IEEE Sensors Journal*, 2023.

[18] S. Suthaharan, "Labelled Wireless Sensor Network Data Repository (LWSNDR)." accessed 2024-05-01.

[19] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," in *2010 sixth international conference on intelligent sensors, sensor networks and information processing*, pp. 269–274, IEEE, 2010.

[20] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. S. Rossi, "A data-driven architecture for sensor validation based on neural networks," in *2020 IEEE SENSORS*, pp. 1–4, IEEE, 2020.

[21] U. Saeed, Y.-D. Lee, S. U. Jan, and I. Koo, "Cafd: Context-aware fault diagnostic scheme towards sensor faults utilizing machine learning," *Sensors*, vol. 21, no. 2, p. 617, 2021.

[22] J. Huang, M. Li, Y. Zhang, L. Mu, Z. Ao, and H. Gong, "Fault detection and classification for sensor faults of uav by deep learning and time-frequency analysis," in *2021 40th Chinese Control Conference (CCC)*, pp. 4420–4424, IEEE, 2021.