Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 06, June - 2024

The Development of a System to Analyze, Inspect and Predict in Advance the Failure of Centralized Variable Refrigerant Flow Air Conditioners Using Machine Learning

Warit Siricharoensuk, Prasong Praneetpolgrang, Surasak Mungsing
Information Technology Program
Faculty of Information Technology Sripatum University,
Bangkok, Thailand

***Abstract*—This research is to develop a system for, analyzing, inspecting and predicting failures in centralized variable refrigerant flow air conditioners using machine learning. Researchers propose to develop the prediction system through the Cloud for ease of use, utilizing platforms such as Google Drive, Google Sheet, Google AppSheet, and Google Cloud Function. TensorFlow is used with Python scripts from Neural Networks algorithms, with a learning cycle set to 500 rounds. Data is fetched from Google Sheet, analysis, sent back and displayed through an application created with Google AppSheet to show the prediction results of system failures.**

***Keywords*— Variable Refrigerant Flow; Deep Learning; Predictive Model**

I. INTRODUCTION

Variable refrigerant flow (VRF) air conditioners [1] have gained widespread popularity in large buildings due to their high cooling efficiency and precise temperature control. However, such systems are complex in components and mechanisms, making them prone to malfunctions and damages, affecting cooling efficiency, energy consumption, and user satisfaction.

Therefore, developing a system that can analyze, inspect, and predict potential failures in such air conditioning systems is crucial. Applying machine learning techniques allows it to efficiently analyze data from various sensors, detect patterns and early signs of anomalies, and plan appropriate preventive maintenance. This leads to a longer system lifespan, reduced energy loss, continuous operational efficiency, and increased user satisfaction.

II. RESEARCH OBJECTIVES

The objective of this research is to develop a system for inspecting, analyzing and predicting the failure symptoms of a central air conditioner with variable refrigerant based on the amount of refrigerant using machine learning.

III. RELATED THEORIES AND RESEARCH

A. Neural Networks

Neural networks are mathematical models inspired by the functioning of the human brain in processing information and learning from experiences. They consist of numerous small processing units connected in a network, capable of processing various types of input data such as images, sounds, and text. The working process of neural networks [2], [3] begins with receiving input data, which is then processed through mathematical computations on different processing units. Parameters such as weights and constants are set to control the flow and transmission of information between processing units. The learning process of neural networks adjusts these parameters through supervised learning or unsupervised learning methods to achieve accurate results with minimal error. This enhances the efficiency of neural networks in processing and classifying data over time. The model structure is illustrated in Fig. 1.

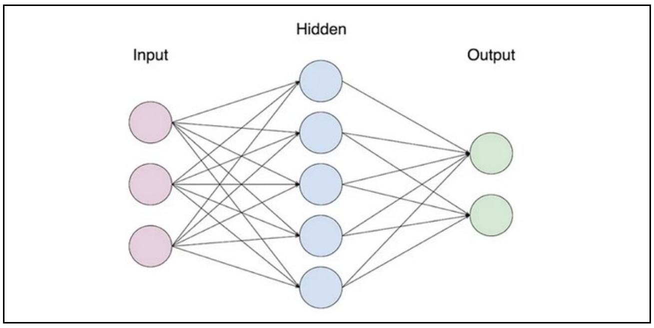

Fig. 1. Neural Networks Decision

B. The Synthetic Minority Over-sampling Technique

The Synthetic Minority Over-sampling Technique (SMOTE) [4], [5] is used to address the problem of data imbalance in machine learning by creating new samples in the minority class through synthetic data generation. The main steps involve selecting samples in the minority class and generating new samples using data from nearby samples. Using SMOTE helps improve the performance of learning models and reduces the risk of overfitting. However, it involves complex processing and the risk of generating unreasonable data. The use of SMOTE should consider the characteristics of the data and the specific application of each case.

ISSN: 2278-0181
Vol. 13 Issue 06, June - 2024

Having balanced data is crucial for the performance of machine learning. Imbalanced data refers to situations where the number of samples in each class differs significantly, which can cause the model to bias towards the class with more samples. SMOTE is a technique used to create synthetic samples in the minority class to increase the number of samples to be closer to other classes, helping to reduce bias and improve model performance. However, SMOTE has limitations in terms of computational complexity, the risk of generating unreasonable synthetic data, and is not suitable for high-dimensional data. Therefore, it should be used cautiously and combined with other techniques to achieve the best results. SMOTE is a useful tool for managing data imbalance problems, helping to increase the accuracy and reliability of the system.

C. Model Evaluation

In the model evaluation step [6], [7], the performance of the model is assessed using various metrics, including accuracy, precision, recall, and F-measure. These can be calculated using Equation (1) – (4) as follows:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (1)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (2)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (3)$$

$$F1-Score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right) \quad (4)$$

TP is the value that the machine learning can decide correctly as expected.
TN is the value that the machine learning can decide to match the unexpected value.
FP is the value that the machine learning decides to be negative but not match the expected value.
FN is the value that the machine learning decides to be negative and match the unexpected value.
Accuracy is the accuracy value that occurs after the machine learning, equal to the total true positives and true negatives divided by the total number of predictions.
Precision is the precision value, which is the value that occurs when repeating the same thing.
Recall is the recall value, which is the value that is interested in the prediction value that is true according to the data.
F1-Score is the value that combines Precision and Recall to give a balanced value between the two values.

D. Related Research

Z. Hou et al. [8] developed a strategy using data mining (DM) principles to detect and diagnose sensor faults based on past performance data in heating, ventilation, and air conditioning (HVAC) systems. This approach combines rough set theory and artificial neural networks (ANN). The reduced data is used to develop classification rules and train the neural network to infer appropriate parameters. The difference between the measured thermodynamic state and the predicted state from the normal performance model (residuals) is used as a performance index to detect and diagnose sensor faults. Real test results from actual HVAC systems show that only the temperature and humidity measurements of many air handling units (AHUs) work well, as the measurements distinguish between simultaneous faults of the supply chilled water (SCW) and return chilled water (RCW) temperature sensors.

Y. Guo et al. [9] used 22 expert rule-based diagnosis rules, divided into 10 rules for the hot coil set and 12 rules for the cold coil set. These rules were created from expert knowledge and the operational characteristics of VRF systems. When tested on nine types of actual faults, such as refrigerant shortage within the system and pressure reducing valve blockage under cooling mode, the method achieved an overall correct diagnosis rate of 85.13%. This demonstrates that the rule-based fault diagnosis strategy is effective in identifying various faults in VRF systems and can be applied to building automation systems.

IV. METHODOLOGY

In this research, researchers imported the dataset of variable refrigerant flow (VRF) air conditioning systems into RapidMiner Studio.[14] then created the model by using SMOTE method to adjust the imbalance of the dataset, then imported the dataset into Google Sheet and used Google Cloud [10], [11] to create API for TensorFlow [15], [16] then used Python language [12] to retrieve the dataset to train Neural Networks model, then measured the performance, and created Google AppSheet and used Bot command menu to run API to predict the result of the failure of central air conditioner with variable refrigerant flow.

A. Conceptual Framework

The conceptual framework of the research consists of 3 steps as follows:

1. Importing and managing the dataset.
2. Creating a model using machine learning.
3. Use the model to develop a system for detecting the malfunction of a central air conditioner with variable refrigerant using machine learning as shown in Fig. 2.

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 06, June - 2024

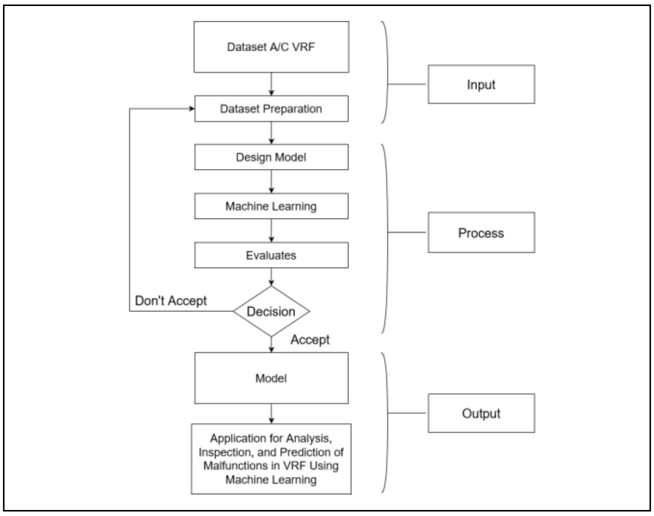

Fig. 2. Conceptual Framework

B. VRF Air Conditioning System Dataset

The dataset details include refrigerant pressure, average superheat temperature, average subcooling temperature, the number of units operating, the number of units not operating, and the status of normal and abnormal operation. The dataset consists of 402 records, with the data attributes detailed in Table 1.

TABLE I. DETAIL OF ATTRIBUTE

Attribute	Attribute Description	Detail
Pres_HL	Average Refrigerant Pressure	Min: 0 Psi/g Max: 316.73 Psi/g
SH_Temp.	Average Superheat Temperature	Min: -1.85 °C Max: 15.71 °C
SC_Temp.	Average Subcool Temperature	Min: -90.17 °C Max: 32.93 °C
Start	Total of Operation FCU	Min: 0 FCU Max: 14 FCU
Stop	Total of Shutdown FCU	Min: 0 FCU Max: 14 FCU
Label	Normal / Abnormal Operating Status	Status Fail / Normal

TABLE II. LABEL RATIO

Label	Amount	Ratio
Normal	323	80.35%
Fail	79	19.65%
Total	**402**	**100.00%**

Table 2. shows that the "Normal" operation status accounts for 80.35%, which is higher than the "Fail" operation status at 19.65%. Therefore, researchers used the SMOTE Upsampling command from RapidMiner Studio to balance the two operation statuses to 50%.

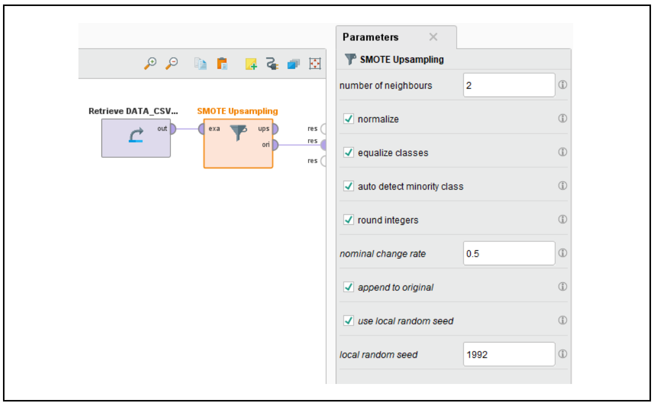

Fig. 3. Operator SMOTE in RapidMiner Studio

As shown in Fig. 3 , researchers set the Nominal Charge Rate to 0.5 to increase the number of records from 79 to 323, adding 244 records, resulting in a balanced ratio of Normal 50%: Fail 50%, as shown in Table 3 comparing the Label Ratio.

TABLE III. COMPARE LABEL RATIO OF SMOTE

Label	Amount	Ratio	SMOTE	SMOTE Ratio
Normal	323	80.35%	323	50%
Fail	79	19.65%	323	50%
Total	**402**	**100.00%**	**646**	**100%**

C. SYSTEM ARCHITECTURE

The system design and structure include three main components: Google Sheet, Google AppSheet, and Google Cloud Function. These components are used for inspecting, analyzing, and predicting failures in VRF air conditioning systems based on refrigerant quantity using machine learning, as shown in Fig. 4. The details are as follows:

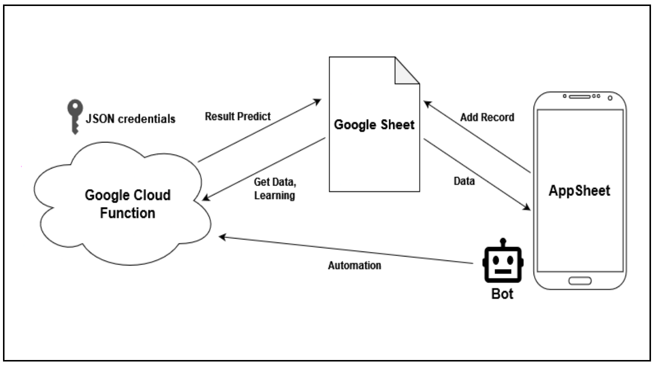

Fig. 4. System Architecture

1) Google Sheet: Used to record data for learning and prediction. Researchers imported the VRF air conditioning system dataset using the SMOTE Upsampling technique into Google Sheet, totaling 646 records as shown in Fig. 5.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 06, June - 2024

	A	B	C	D	E	F
1	Label	Pres_HL	SH_Temp	SC_Temp	Start	Stop
2	Fail	196.53	0.92	-5.43	0	14
3	Fail	202.43	0.66	-5.5	0	14
4	Fail	206.72	0.43	-5.7	0	14
5	Fail	210.97	0.31	-6.17	0	14
6	Fail	218.32	0.36	-5.83	0	14
7	Normal	223.53	1.41	-6.07	0	14
8	Normal	224.70	1.2	-6.7	0	14
9	Normal	225.67	3.99	-7.73	14	0
10	Normal	222.12	3.13	-9.7	14	0
11	Normal	216.88	4.45	-11.57	14	0
12	Normal	236.12	7.52	-3.07	14	0

Fig. 5. Example Data for Machine Learning Training

2) Google AppSheet Uses data from Google Sheet to create an application for recording air conditioning data and displaying prediction results from the model. The connection is set up using a JSON Credential file stored on Google Drive as shown in Fig. 6.

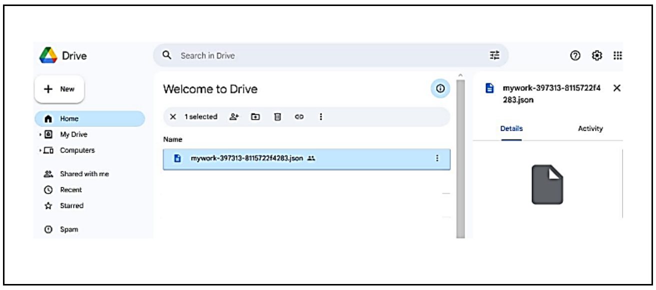

Fig. 6. JSON Credential file on Google Drive

3) Google Cloud Function is used to store Script Python for writing Tensorflow [13] to create models for machine learning with Neural Networks algorithms and set up the use of Script by Bot in Google AppSheet as shown in Fig. 7.

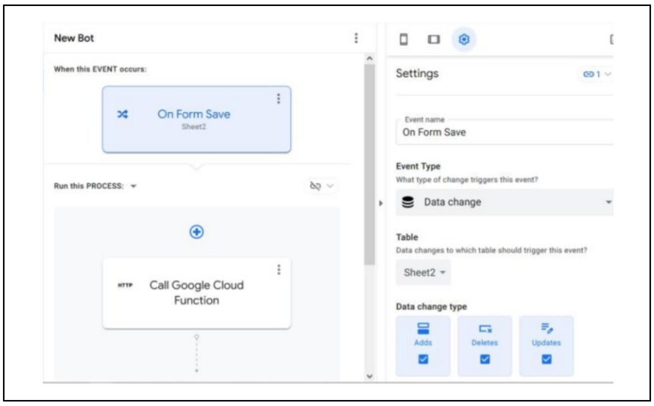

Fig. 7. Set Up Bot

D. Google Cloud Function

Uses Python scripts with Neural Networks algorithms and uploads them to Google Drive to fetch data from Google Sheet, as illustrated in Fig. 4. The commands are set for model training as follows: 80% for training data, 20% for testing, Learning Rate 0.03, Momentum 0.9, Shuffle yes. To measure learning performance, researchers adjusted the Epoch value in different amounts as shown in Fig. 8.

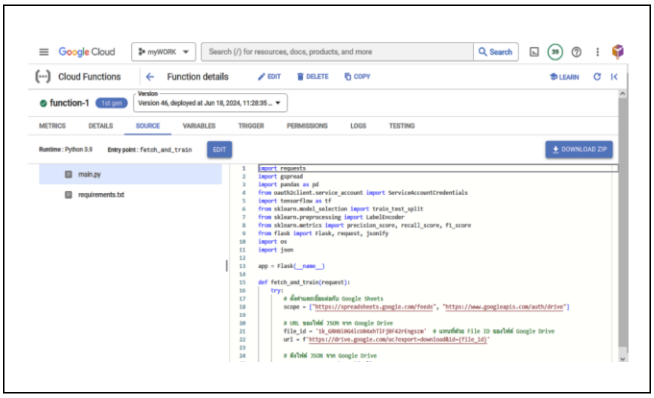

Fig. 8. Script Python on Google Cloud Function

E.Model Evaluation

Using TensorFlow to measure Accuracy, Precision, Recall, and F1 - Score by comparing different training rounds, as shown in Table 4.

TABLE IV. MODEL PERFORMANCE COMPARISON BY LEARNING CYCLE

Learning Cycle	Accuracy	Precision	Recall	F1-Score
500	0.91	0.91	0.91	0.91
400	0.89	0.91	0.91	0.91
300	0.85	0.89	0.85	0.85
200	0.86	0.87	0.86	0.86
100	0.86	0.89	0.86	0.86

Table 4 shows that the performance increases with the number of training rounds, with Accuracy peaking at 500 training rounds and decreasing with fewer rounds.

F. System for Analyzing, Inspecting, and Predicting

The system for analyzing, inspecting, and predicting failures in VRF air conditioning systems based on refrigerant quantity using machine learning consists of three parts:

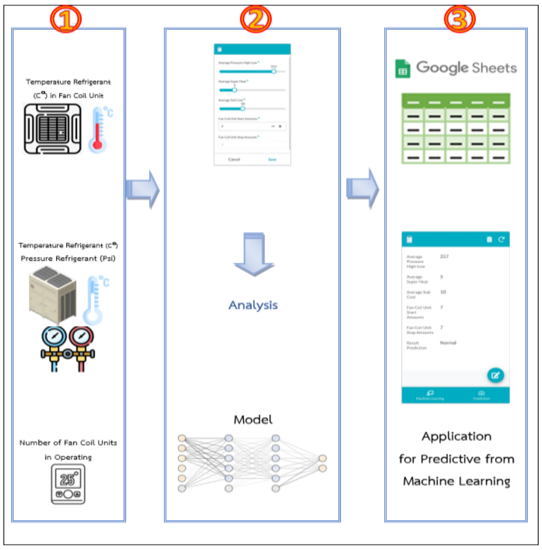

Fig. 9. System for Analyzing, Inspecting, and Predicting

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 06, June - 2024

1. Importing the air conditioning data into the application for analysis, inspection, and prediction.
2. Recording the data into Google Sheet and sending the prediction results to Google Sheet.
3. Google Sheet records the learning data and prediction data, then displays the prediction results through the application as shown in Fig. 9.

V. RESEARCH RESULTS

This research aims to develop a system for inspecting, analyzing, and predicting failures in VRF air conditioning systems based on refrigerant quantity using machine learning. Researchers used the SMOTE technique to balance the data and imported the dataset into the developed system using Google Cloud, Google Sheet and Google AppSheet.

In model creation, researchers used Neural Networks algorithms through TensorFlow on Google Cloud Function, dividing the data into 80% for training and 20% for testing. The research results showed that the model achieved the highest performance at 500 learning rounds, with an Accuracy of 0.91, Precision of 0.91, Recall of 0.91, and F1-Score of 0.91.

The outcome of this research includes the development of an application that operates through the Cloud for predicting failures using a Neural Networks algorithm to make decisions based on refrigerant quantity in VRF air conditioning systems as shown in Fig. 10.

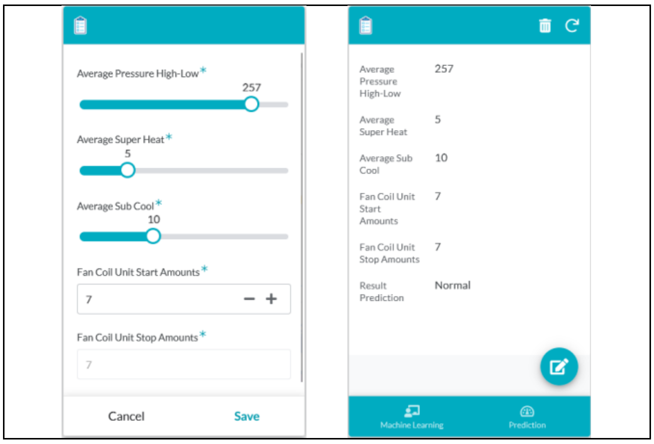

Fig. 10. Application for Failure Prediction

VI. RESEARCH LIMITATIONS

This research aims to develop a system for inspecting, analyzing and predicting failures in VRF air conditioning systems based on refrigerant quantity using machine learning. The limitations include the diversity of hot coil and cold coil quantities, as well as biased data in operational scenarios, which may lead to prediction bias.

VII. FUTURE RESEARCH

For future research on developing a system for inspecting, analyzing, and predicting failures in VRF air conditioning systems based on refrigerant quantity using machine learning, researchers suggest the following:

1. The system requires a larger amount of data to improve prediction accuracy.
2. It can be used to predict failures caused by other factors.
3. The development of this prediction system can be applied through the Cloud, making it more convenient to use.

REFERENCES

[1] W. Goetzler, "Variable refrigerant flow systems," Ashrae Journal, vol. 49, no. 4, pp. 24-31, 2007.
[2] S. Krishna. "Building an Artificial Neural Network(ANN)," https://medium.com/@saikrishna3599/building-an-artificial-neural-network-ann-502a63d76fb3 (accessed June 30, 2020).
[3] R. Zulunov, U. Akhundjanov, K. Musayev, B. Soliyev, A. Kayumov, and M. Asraev, "Building and predicting a neural network in python," in E3S Web of Conferences, 2024, vol. 508: EDP Sciences, p. 04005.
[4] M. Mujahid et al., "Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering," Journal of Big Data, vol. 11, no. 1, pp. 1-32, 2024.
[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of artificial intelligence research, vol. 16, pp. 321-357, 2002.
[6] T. Srivastava. "12 Important Model Evaluation Metrics for Machine Learning Everyone Should Know (Updated 2023)," https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/ (accessed Jan 08, 2024).
[7] Ž. Vujović, "Classification model evaluation metrics," International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, pp. 599-606, 2021.
[8] Z. Hou, Z. Lian, Y. Yao, and X. Yuan, "Data mining based sensor fault diagnosis and validation for building air conditioning system," Energy Conversion and Management, vol. 47, no. 15-16, pp. 2479-2490, 2006.
[9] Y. Guo et al., "An expert rule-based fault diagnosis strategy for variable refrigerant flow air conditioning systems," Applied Thermal Engineering, vol. 149, pp. 1223-1235, 2019.
[10] E. Bisong, Building machine learning and deep learning models on Google cloud platform. Springer, 2019.
[11] A. Gupta, P. Goswami, N. Chaudhary, and R. Bansal, "Deploying an application using google cloud platform," in 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020: IEEE, pp. 236-239.
[12] G. Van Rossum and F. L. Drake Jr, "Python tutorial," ed: Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.
[13] T. Hope, Y. S. Resheff, and I. Lieder, Learning tensorflow: A guide to building deep learning systems. O'Reilly Media, Inc., 2017.
[14] N. Baharun, N. F. M. Razi, S. Masrom, N. A. M. Yusri, and A. Rahman, "Auto modelling for machine learning: a comparison implementation between rapid miner and python," Int. J. Emerg. Technol. Adv. Eng., vol. 12, no. 5, pp. 15-27, 2022.
[15] P. Sarang, "Artificial neural networks with TensorFlow 2," Apress: Berkeley, CA, USA, 2021.
[16] S. Pattanayak, J. S. Pattanayak, and S. John, Pro deep learning with tensorflow. Springer, 2017.