Journal of Big Data

RESEARCH Open Access

Detecting unregistered users through semi-supervised anomaly detection with similarity datasets

Dong Hyuk Heo[1], Sung Ho Park[2] and Soon Ju Kang[1*]

*Correspondence:
sjkang@knu.ac.kr

[1] School of Electronics Engineering, Kyungpook National University, Daegu, Republic of Korea
[2] Center of Self-Organizing Software, Kyungpook National University, Daegu, Republic of Korea

Abstract

Recent research has focused on exploring systems that incorporate anomaly detection models to automate the addition of users in user recognition systems. Anomaly detection, a method used to distinguish between new and existing users by identifying abnormal images, has gained significant attention. Researchers have been actively investigating the Semi-Supervised Learning method, which utilizes only existing user data to differentiate between existing and new users. However, existing semi-supervised learning based anomaly detection models exhibit high performance on datasets with low similarity but experience a sharp decline in performance on datasets with high similarity. Furthermore, their large model size makes it challenging to execute them on edge nodes. To address these limitations, this paper proposes a model that can be executed on edge nodes and guarantees good performance on both low and high similarity datasets. The proposed model utilizes the LeNet-5, a user recognition model with fewer weights and multiple images as input, for classifying new users. This study compared the existing anomaly detection models with the proposed model using three datasets with varying similarities. The performance evaluation involved comparing the accuracy, ROC curve, and AUC of each model on a training server. Subsequently, the top three models were optimized for execution on the edge node (STM32F207ZG MCU) and further evaluated by comparing their accuracy, inference speed, and model size. The results revealed that the proposed model achieved an approximate 53% improvement in accuracy compared to the existing anomaly detection models. Furthermore, when executed on the edge node, the proposed model demonstrated significant memory savings, with a maximum reduction of approximately 530% and approximately 40% reduction in flash memory usage compared to the existing models.

Keywords: Unregistered users, Similarity datasets, Semi-supervised anomaly detection, Edge computing, Artificial Intelligence on edge node

Introduction

Anomaly user detection is an essential system for automating user addition in user recognition systems. There are two ways to add users in a user recognition system: manually collecting user data and retraining the model [1–4] or determining whether

the input data is from a new user or an existing user and retraining the model accordingly [5–8]. The former method can achieve high accuracy because a supervisor recognizes new users, collects high-quality data, and retrains the existing model, but it requires many steps because a supervisor must do it manually [9, 10]. Therefore, various methods using anomaly detection to classify data into existing users and new users and retrain the model have been researched recently, despite the disadvantage of having slightly lower accuracy than the former method [7, 8, 11, 12]. The existing anomaly detection methods can be classified into machine learning algorithm-based methods [13–16] and deep learning algorithm-based methods [17–22]. Each algorithm is further divided into supervised learning [23, 24], semi-supervised learning [25, 26], and unsupervised learning [27, 28]. Among various techniques, semi-supervised learning methods are gaining considerable attention in research [15, 16, 18, 25, 26]. These methods utilize only existing user data to establish a discriminative boundary. This boundary is then tightened to identify any data outside of it as abnormal. Therefore, as the models currently under investigation are trained exclusively on the data of existing users, new users data that exhibits high similarity to the training data is likely to be mistakenly classified as belonging to an existing user [29, 30]. Additionally, existing anomaly detection models based on semi-supervised learning require high-performance servers due to their large model size, making a system capable of transmitting data to the server also necessary [31]. Therefore, there is a real-time performance degradation [32, 33] caused by the overhead of transmitting user data from the edge node to the server after measurement. Consequently, this paper proposes a system that directly recognizes new users on the edge node, rather than on the server, in order to achieve real-time detection of new users. This system involves measuring data and incorporates a semi-supervised learning-based anomaly detection model that relies solely on existing user data to determine new users. This paper proposes a system that detects anomaly data to identify new users in a user recognition system. Unlike sending data to a dedicated server for training, the system runs on an STM32F207ZG MCU-based edge node for real-time inference. The edge node includes a system [34] for measuring and generalizing users' foot pressure data. This paper proposes a system that augments the existing system by incorporating a model for classifying new users using multiple images and the LeNet-5 model [35–38], which is a CNN algorithm known for its low number of weights and high accuracy in image recognition. To determine a new user with multiple images, the captured images were inputted into the user recognition model, and the mean of all predicted values and the threshold value were compared. The model was trained using existing user data because it cannot have advance knowledge of new user data. So, it was trained based on the Semi-Supervised Learning approach to detect abnormal data. For the experiments, datasets with different levels of similarity were used to compare their accuracy. High similarity datasets, such as the foot pressure dataset, and low similarity datasets, such as Fashion-MNIST and Digit-MNIST, were utilized. As a result, the existing Anomaly Detection models exhibited an average accuracy of 83% on low similarity datasets. However, on the high similarity foot pressure dataset, there was a decrease of approximately 22%, resulting in an average accuracy of 61%. To overcome this issue, the model size needs to be adjusted, but this is limited by the edge node's computing resources. However, the model proposed in this paper demonstrated an accuracy

increase of approximately 29% compared to existing Anomaly Detection models when using datasets with high similarity. It achieved an accuracy of 89%. Additionally, when Quantization and Pruning [39] were performed for accuracy measurement on the Edge node, there was a slight decrease in accuracy by approximately 3%, resulting in an accuracy of 86% for new user classification. Additionally, this model utilizes the LeNet-5 model [35–38] on the edge node, enabling real-time recognition of both existing and new users. Consequently, it is possible to utilize the model in various systems, such as a system that automatically adds new users, by transmitting only abnormal data to the server for transfer learning and receiving the model from the server. The main achievements of this paper can be summarized as follows.

1. The proposed system is capable of accurately recognizing new users based on similar datasets with high similarity.

2. The system can recognize new users in real-time with high accuracy, using limited computing resources on the edge node.

Research motivation

In an automated user recognition system that adds new users automatically, anomaly detection is essential to determine whether the current data belongs to a new user or an existing user [5–8]. The most effective approach is to utilize open-set recognition [5, 6, 29, 40, 41], where the system can identify the user when provided with data from existing users and classify data from new users as unknown individuals. However, open-set recognition heavily relies on the distance between logit vectors, making it highly influenced by data similarity [29]. In other words, if the similarity between classes is high, accurate judgment cannot be made. Moreover, anomaly detection models trained using traditional semi-supervised learning approaches are significantly affected by similarity [30]. Therefore, a model capable of anomaly detection in datasets with high similarity, such as fingerprints, iris, or foot pressure, is required. One approach to achieve this is by using large-scale models, but it is essential to have a high-performance server [42–44]. Additionally, there is a need for data transmission from the client to the server for execution [31]. However, there is a disadvantage of reduced real-time inference due to the predictability degradation caused by data transmission [32, 33]. Therefore, in this paper, we aimed to address the anomaly detection in the high similarity foot pressure dataset at the edge node. To achieve this, we utilized the LeNet-5 model [35–38], which has a relatively low number of weights. This choice enables faster execution at the edge node and offers satisfactory performance. Additionally, we employed the LeNet-5 model multiple times to tackle this issue. Additionally, in this paper, to ensure predictability, we applied pruning and quantization techniques [39] to the LeNet-5 model. By pruning, we reduced the number of weights, and through quantization, we reduced the bit size of the weights. This enabled the edge node to directly determine the presence of abnormal data. Therefore, by utilizing this model, we can distinguish between existing users and new users, enabling the provision of real-time personalized services. Ultimately, in the planned research system, which includes the automation of user addition, the user recognition system will be capable of determining whether the current user is an existing user or a new user.

Related research

The proposed model in this paper operates on the edge node and serves as a model that identifies new users by detecting abnormal data in datasets with high similarity. Therefore, this paper describes research on detecting Unregistered users [5–8] and explores relevant studies to select the optimal anomaly detection model from the perspective of the edge node [11, 12, 36–38, 42–45].

Detection models for unregistered users

Detecting new users can be done through two methods: manual detection by a supervisor [1–4], and using Anomaly Detection [7, 8, 13–22, 25, 26, 30, 45] or Open-Set Recognition models [5, 6, 29] to detect and identify new users. When a supervisor collects the data, high-quality data can be obtained, and there is potential for achieving high accuracy through transfer learning [1, 4]. However, this approach necessitates the presence of a supervisor throughout the data collection process. However, when using open-set recognition or anomaly detection models to detect new users, it is not possible to classify with 100% accuracy. This can lead to potential data contamination, and consequently, a decrease in accuracy when performing transfer learning [40, 41]. However, one advantage of these methods is that they do not require the presence of a supervisor, reducing the need for human intervention. Therefore, there has been a significant amount of research in recent years on models that can distinguish between existing and new users in order to develop systems that automatically add new users by detecting abnormal data. The methods of open-set recognition and anomaly detection, which are used for recognizing new users, can be described as follows.

Open-set recognition

Existing user recognition models tend to classify unknown class data as existing users with high probability. Open-set recognition [5, 6, 29] is a method that supplements existing user recognition models by adding a process of classifying unknown class data as an unknown class. This is achieved by calculating a Logit Vector at the layer just before the softmax layer of the existing user recognition model when unknown class data is input. Then, the calculated value is compared with the mean value of each class's training data that has been pre-calculated, and the results including the unknown class are output, and the final probability is obtained using the softmax layer. This method has the advantage of being able to recognize users and detect new users simultaneously. However, because this method uses the mean of the Logit Vectors between each class, it shows good performance in datasets with a low degree of similarity between classes, but the performance deteriorates in datasets with high degrees of similarity between classes [29]. In addition, there are unnecessary operations, such as computing the Logit Vector for each input data, and unnecessary resources are consumed by storing the mean Logit Vector for each class separately. Therefore, it is not suitable for use on edge nodes with limited computing performance.

Anomaly detection

Anomaly detection [7, 8] is a method of determining whether input data belongs to an existing or new user. There are various algorithms depending on the presence or absence of labels in the dataset and whether new label data is available during training. In this paper, we used a semi-supervised learning approach to detect abnormal data by labeling an existing dataset since new user data was not available. The Semi-Supervised learning [25, 26] approach aims to detect abnormal data using only existing datasets. The key idea is to narrow down the boundary surrounding the normal data as much as possible, classifying external data as abnormal. Therefore, the loss function becomes a combination of supervised loss and unsupervised loss. In other words, instead of reducing the loss by classifying as many normal data as possible, the model is trained to classify both normal and abnormal data as accurately as possible. Isolation Forest, SVM, Auto-Encoder, Auto-Encoder with K-NN, and CBIR with K-NN are representative models that use the semi-supervised learning approach [25, 26] to detect abnormal data. Isolation Forest [13, 14] represents the training data as a decision tree and uses the feature of finding abnormal data at the top to identify it. SVM [15, 16] uses one-class SVM to cluster the data and identifies data as abnormal when it is far from the clustered data. Neither method is suitable for image datasets as their accuracy decreases with increasing number of dimensions. Deep learning algorithms are being studied to supplement these methods. Auto-Encoder [17, 18] trains the internal algorithm as MLP or CNN layers to create data similar to the training data and then compares the difference in the results to determine abnormal data. In addition, there is a method to determine abnormal data by using K-NN to extract N data from the compressed middle layer of the Auto-Encoder [19, 20] or the Convolution Layer of CNN-based algorithms [21, 22] like LeNet-5 and counting the number of classes to which they belong. However, existing models cannot guarantee performance for datasets with high similarity [30], such as the open-set recognition method. Moreover, to ensure good accuracy, they require many layers and weights, making it difficult to guarantee good accuracy on edge nodes with limited computing performance [45].

Optimal anomaly detection model on edge node

Previous research on anomaly detection on edge nodes has utilized models [11, 12, 45] based on Sparsity Profile or Echo State Networks. These studies involve statistical algorithms or multilayer perceptron-based approaches. However, these methods may not guarantee performance in datasets with high similarity unless the model size is increased. Additionally, they may not be suitable for extracting and classifying features in image datasets. Furthermore, anomaly detection models based on machine learning algorithms such as LightGBM or XGBoost, which are currently being explored, require a significant number of weights and take more than 0.4 s for prediction on high-performance servers [42–44, 46]. Even if the weights are pruned to make them feasible on edge nodes, the execution time on the edge node remains excessively long, making it impractical for deployment. However, instead of such models, using a model based on CNN algorithms [5, 6, 17, 18, 21, 22], which can effectively extract features from images,

is more suitable for detecting abnormal data. Furthermore, a classification model [5, 6, 21, 22, 47, 48], which is less dependent on data similarity, would be more appropriate. However, since it needs to be executed on an edge node, it is important to select a model that has minimal weights and demonstrates satisfactory performance. Among models with low weights and good performance, two prominent examples are LeNet-5 [35–38] and AlexNet [36, 37]. Although the accuracy of LeNet-5 is approximately 4% lower than that of AlexNet [36, 37], its weight count is about 1/100 times compared to AlexNet [38]. This makes LeNet-5 the most suitable model for execution on edge nodes, considering its minimal weight requirement. Therefore, in this paper, we propose a system for detecting new users on datasets with high similarity using multiple images from the moment a user steps on the footpad to the moment their foot leaves the footpad. We utilize the LeNet-5 user recognition model, which ensures satisfactory performance with minimal weights.

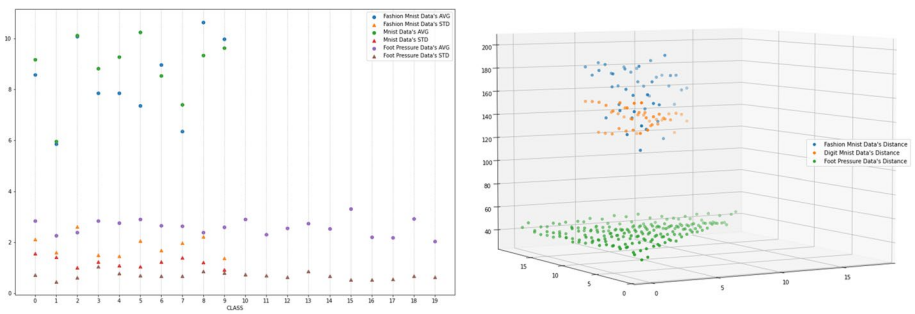
Problems definition

In this paper, experiments were conducted to compare the accuracy based on the similarity using three different datasets. The models used in the experiments included six existing models and the proposed model. Each dataset consists of 48 × 48 grayscale image data. The characteristics of each dataset are shown in Fig. 1.

In this paper, we analyzed the similarity of the datasets used in the experiments based on the similarity within the datasets belonging to the same class and between the classes. The similarity within the datasets belonging to the same class was measured by randomly selecting representative data for each class and calculating the distance between the datasets, then calculating the mean and standard deviation. The similarity between each class was calculated by measuring the distance between the datasets of two different classes and calculating the mean. The Euclidean Distance algorithm was used to measure the distance, as shown in Eq. (1).

$$\left(\sum_{i=1}^{48}\sum_{j=1}^{48}(|X_{(i,j)} - Y_{(i,j)}|)^2\right)^{\frac{1}{2}} \tag{1}$$

When looking at the Digit-Mnist [49] and Fashion-Mnist [50] datasets, we can see that the similarity between datasets belonging to the same class is relatively low due to the



The similarity within the datasets belonging to the same class

The similarity between each class

Fig. 1 Comparison of similarities

large distance between them. Additionally, the similarity between each class is spread out and if the absolute value is high, we can determine that they are not similar. Therefore, in terms of similarity, the Fashion-Mnist dataset exhibits a relatively wide distribution and higher values, resulting in the lowest similarity between classes. Following that is the Digit-Mnist dataset, which also has higher values. In contrast, the Foot Pressure dataset shows lower similarity values compared to the other two datasets. The reason for this is as follows: the foot pressure measurement system [34] used in this study selects intact foot pressure data that satisfies the conditions, rather than using all of the user's data, and generalizes the angle and position through a preprocessing process. Therefore, the similarity between datasets belonging to the same class is very low, and since everyone's feet do not differ much from each other, such as with numbers or types of clothing, the similarity between each class is also very low. Therefore, when evaluating the performance of the existing anomaly detection models using the low similarity Digit-Mnist and Fashion-Mnist datasets, they demonstrate high performance with an accuracy of 83%. However, when applied to the high similarity Foot Pressure dataset, the performance is significantly lower, with an accuracy of 61% or less, indicating poor performance. Additionally, this study had to use a model that utilizes a maximum of 1Mb of flash memory and 128 Kb of memory for distinguishes new users on the edge node, so the size of the model is limited. Therefore, this study proposes an anomaly detection model that can guarantee the accuracy of similar datasets on edge nodes to overcome this issue.

Overview of entire system

The following is the proposed anomaly detection system for automating the addition of users in this paper.

The flow chart of this paper is presented in Fig. 2. When a user steps on the footpad, their foot pressure data is measured. The collected data is then normalized and

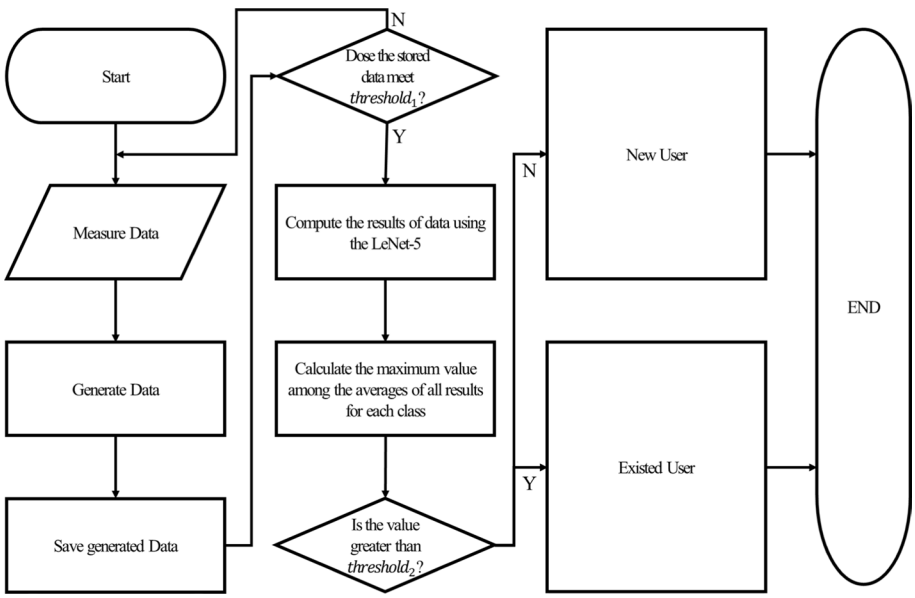


Fig. 2 System flow chart

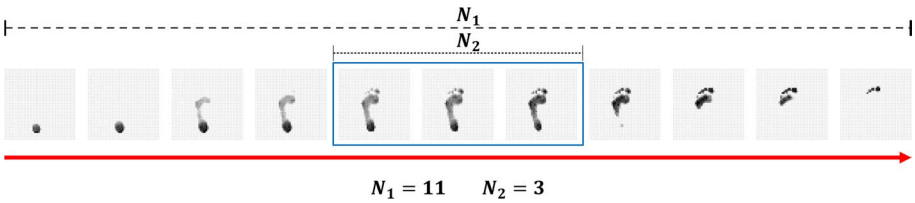


Fig. 3 Measurement of foot pressure as the user walks on the pad

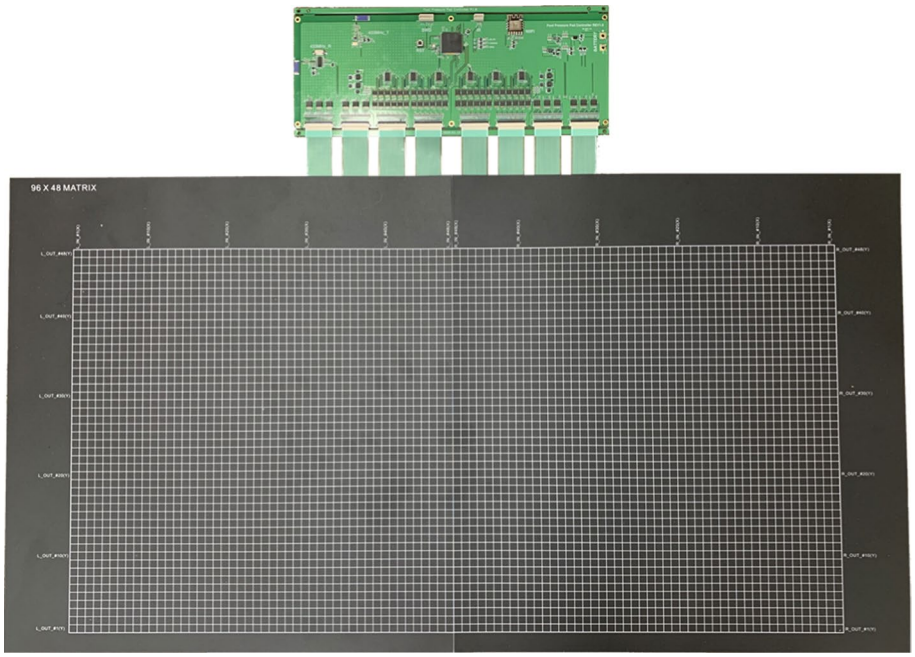


Fig. 4 Edge based foot pressure measure device

stored. If a sufficient number of normalized data, referred to as $threshold_1$, are gathered, the LeNet-5 model is utilized to calculate the accuracy of each data point. Subsequently, the average accuracy for each class is computed. If the maximum value among the average values for each class is greater than $threshold_2$, it means that all the data predicts the same person, indicating an existing user. On the other hand, if the maximum value is less than $threshold_2$, it means that one or more data points predict a different person, indicating a new user. The method for data normalization and setting $threshold_1$ is as follows.

This paper accumulates N_2 datasets, such as the blue box identified through the preprocessing process, from N_1 datasets measured from the moment a user enters the foot pad and until they exit, as shown in Fig. 3. The proposed system in this paper measures the user's foot pressure using an edge device in Fig. 4, and the preprocessing process [34] involves verifying intact data (presence of toes, front and rear centers of gravity, and a ratio between the distance of centers of gravity and the total measured cells being equal to or greater than the threshold), as well as generalizing angles and positions. Then, the preprocessed datasets are accumulated and input into the

LeNet-5 model, and the results are used for anomaly detection. The method for determining anomaly detection is described in Algorithm 1

Algorithm 1 Foot Pressure Anomaly Detection

```
procedure
    Collect Intact foot pressure dataset
    if Intact foot pressure dataset is bigger than threshold_1 then
        Select threshold_1 dataset
        for all Intact foot pressure dataset do
            Save user identification results for each class
        end for
        Calculate the Average of results for each class
        if The max value of each class's average is bigger than threshold_2 then
            return Existing User
        else
            return New User
        end if
    end if
end procedure
```

Algorithm 1 is the method for detecting anomaly data, where the detection begins when the accumulated number of intact data is equal to or greater than $threshold_1$ and restarts measurement when the number is less. The method for setting $threshold_1$ in this paper is as follows.

This paper compared the total number of measured data and the accumulated intact data during walking on the foot pad to set $threshold_1$. For instance, when User A walked on the foot pad as shown in Fig. 3, 11(N_1) pressure measurements were taken, and 3(N_2) intact data were accumulated. This paper conducted experiments by randomly selecting 100 data from the measured data of 20 users during walking, and found that on average, 12 measurements were taken and about 3 intact data were judged, as shown in Fig. 5. Therefore, since the system accumulates an average of 3 intact data, $threshold_1$ was set to 3, and if the number of data sets was less than that, the measurement was restarted. Then, user identification was performed using the three intact data. Finally, the accuracy of the user identification results was calculated as the average for each class, and if the highest average value was $threshold_2$ or higher, all three data were recognized as belonging to the same user with a high probability, and judged as an existing user. Otherwise, it was judged as a new user. $threshold_2$ was selected as the threshold value that maximizes the model's F1-score when varying the threshold.

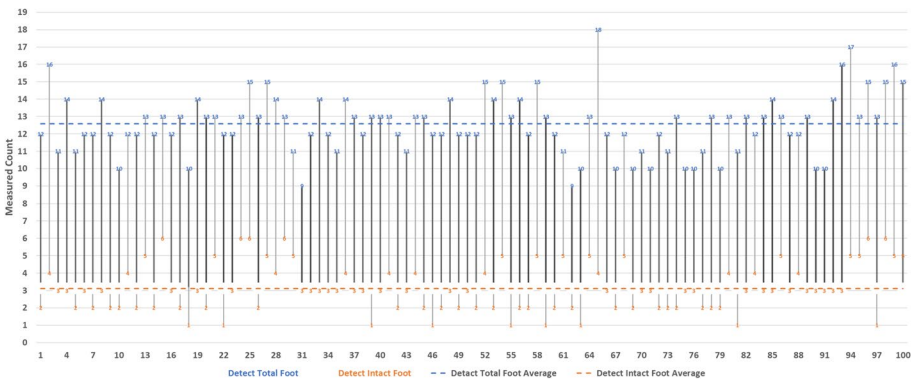


Fig. 5 The dataset collected from the moment the user steps onto the foot pad until they exit it, and the number of intact datasets determined from it

Anomaly detection experiment

Data construction

This paper used three datasets to compare the performance of the proposed model with the existing model on datasets with different similarities. Each dataset has a different level of similarity, and the foot pressure dataset consists of data collected directly from 20 users. The construction method of all datasets is the same, and the data construction method is described based on the foot pressure dataset. The foot pressure dataset was used to evaluate the performance of the anomaly detection model using data from 20 users. The dataset consists of intact data with generalized angles from the measured data while the user walked on the pad, and each user has over 1000 data. This paper randomly selected 1000 data from each user's data and combined them, and the dataset construction is shown in Fig. 6.

This paper divided 20 users into two groups: 10 known users and 10 unknown users, and created a dataset for model training using existing user datasets and a dataset for performance evaluation of the anomaly detection model consisting of data from all users. The dataset for model training was divided into Train dataset, Validation dataset, and Test dataset in a ratio of 7:1:2 using the data of 10 users. The anomaly detection model was trained using the train and validation datasets. Then, if the model was trained well, the test dataset of the training dataset was labeled as normal data, and the remaining dataset of the 10 users, randomly selecting 20%, was labeled as anomalous data to create a dataset for evaluation. Using this method, all datasets were separated, and the accuracy of the models was compared based on the data similarity.

Evaluation methodology

In this paper, we performed training of the anomaly detection model on a server and then optimized it by applying pruning and quantization using the TFLM library [51] for execution on an edge node. Consequently, on the training server, we evaluated the accuracy,

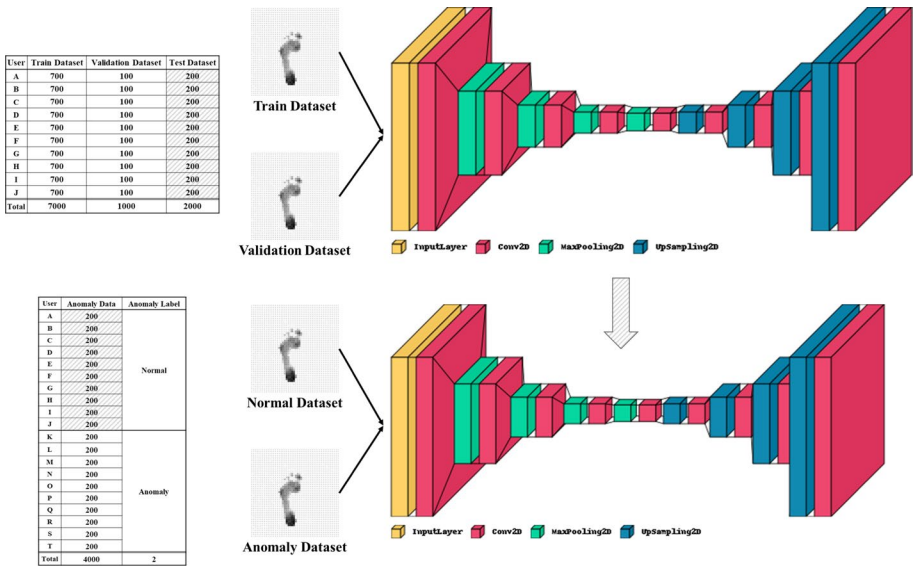


Fig. 6 The system train dataset and anomaly test dataset

ROC curves, and AUC of all models. The top two models including the ones proposed in this paper were further optimized and evaluated for execution speed, accuracy, and model size on the edge node, validating the excellence of this paper's approach. The following outlines the methodology used to evaluate the accuracy, ROC curves, and AUC of the models.

Accuracy

The accuracy of the model varies depending on the threshold, so the following method was used to determine the threshold. The threshold value is adjusted by evaluating the F1-score based on the results obtained from applying the anomaly test dataset to the model trained on the existing user dataset. The threshold value was set at the point where the F1-score is maximum, and the F1-score is calculated using Recall and Precision, which are determined as follows:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

Precision is the probability of correctly identifying positive data among the data predicted as positive. Recall is the probability of correctly identifying positive data among all actual positive data. The F1-score is calculated by multiplying the precision and recall, and then dividing the result by the sum of precision and recall. Finally, the result is multiplied by 2. Therefore, the F1-score is the harmonic mean of precision and recall. The F1-score ranges between 0 and 1, where higher values indicate better performance.

ROC curve

The ROC curve is a graphical representation that shows the performance of a binary classifier model, such as anomaly detection, across different thresholds. It allows us to visualize the performance of multiple models at a glance. The ROC curve is created by plotting the recall (true positive rate) on the y-axis against the false positive rate (1 − specificity) on the x-axis, as the threshold is varied.

$$ROC_x = \frac{False\ Positive}{False\ Positive + True\ Negative} \tag{5}$$

$$ROC_y = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{6}$$

AUC

AUC is a metric calculated from the area under the ROC curve. It provides a measure of the overall performance of a model. AUC ranges from 0 to 1, where a value closer to

1 indicates a better-performing model. Higher AUC values indicate a higher degree of separability between the classes and represent a more optimal classification model.

$$AUC = \int_0^1 ROC_x \, d(ROC_y) \tag{7}$$

Baseline anomaly detection algorithms

The model proposed in this paper is a semi-supervised learning-based model [25, 26] that utilizes the results of 3 user foot pressure data measurements taken as the user passes through the foot pad and input into LeNet-5. This paper compared the proposed model's performance with an anomaly detection model [13, 15, 17–19, 21] commonly used in recent Semi-Supervised Learning approaches to evaluate its performance.

Isolation forest

Isolation Forest [13, 14] is a method of finding anomaly data by specifying the depth of search as a threshold, because normal data can be found at the bottom of the decision tree, and abnormal data can be found at the top. Therefore, in this paper, the Isolation Forest model of scikit-learn [52] was used, with 100 decision trees, 256 features, and 0.1 contamination. As shown in Fig. 7 of the foot pressure dataset, the F1-score was highest at 0.01, so the threshold was set to 0.01. As a result of the test, the model showed an accuracy of 84.6% for the train dataset, 83% for the validation dataset, and 51.9% for the test dataset.

Support vector machine

This paper constructed a one-class SVM model [15, 16] based on Semi-Supervised Learning for anomaly detection. This model determines whether the input data is an anomaly by calculating the distance between the input data and the clustered training data. Therefore, this paper used the OneClassSVM model from scikit-learn [52], using rbf kernel for non-linear boundaries, setting gamma to auto, and nu to 0.1 to minimize

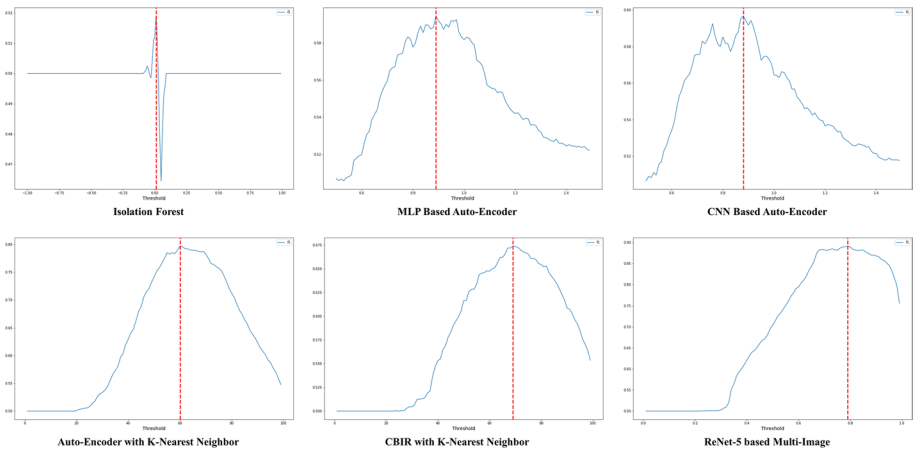


Fig. 7 Threshold for all algorithms

the training error rate. The test results using the user's foot pressure dataset showed an accuracy of 90.1% for the train dataset, 91.4% for the validation dataset, and 50.2% for the test dataset.

Auto-encoder

This paper used an Auto-Encoder model [17, 18] that produces an output data identical to the input data for anomaly detection. The detection method involves calculating the Euclidean Distance between the input data and the output data, and comparing it with the threshold value. Therefore, this paper compared the internal structure of the Auto-Encoder model using MLP and CNN methods, as shown in Fig. 8. The optimizer used was Adam, and the loss function used was Mean Squared Error. Additionally, experimental results using user foot pressure data showed that the MLP-based model had a threshold of 1.13, with 77% accuracy for the train dataset, 73.9% for the validation dataset, and 58.9% for the test dataset. The CNN-based model had a threshold of 1.09, with 77.6% accuracy for the train dataset, 75% for the validation dataset, and 56.7% for the test dataset.

Auto-encoder with K-nearest neighbor

Auto-Encoder with K-NN [19, 20] is a method that adds the K-NN algorithm to improve the traditional Auto-Encoder approach. It involves using the compressed hidden layer

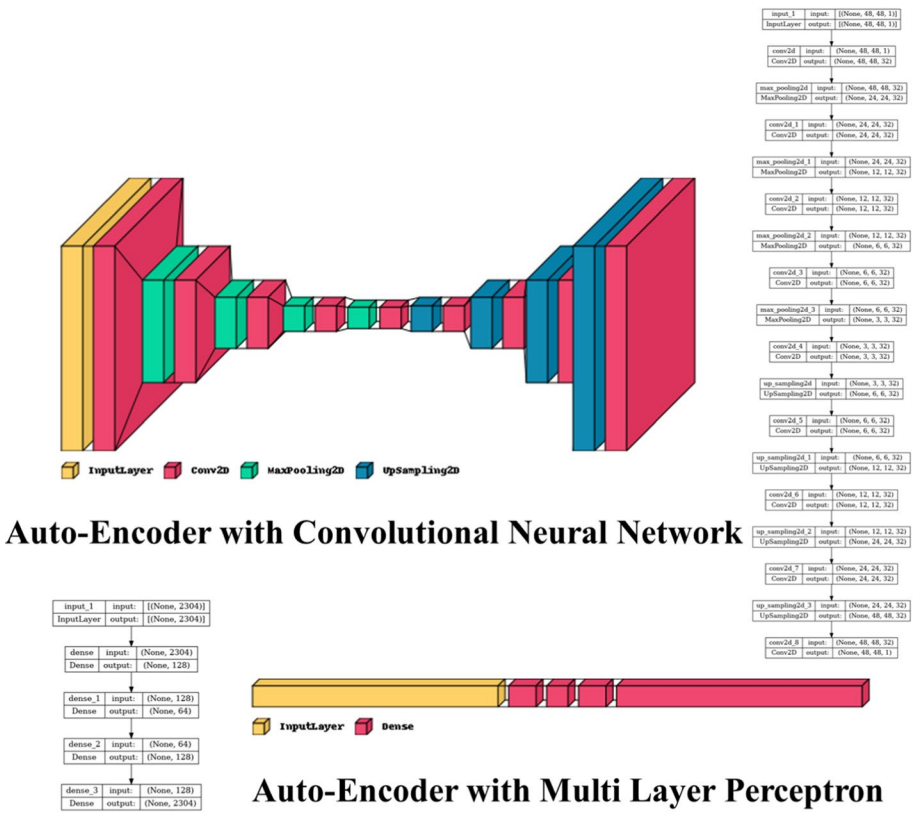


Fig. 8 Auto-encoder model using CNN and MLP

obtained while training the MLP-based Auto-Encoder approach as input data to the K-NN algorithm for anomaly detection. For performance evaluation, this paper constructed the entire model as shown in Fig. 9, and used the same Auto-Encoder model as the MLP-based model in Fig. 8. Additionally, instead of using the results directly from the Dense_2 layer, normalization was applied as shown in Eq. (8), and then additional training was performed using the K-NN algorithm with k=100 and Euclidean Distance. When the test data was input, the result from the Dense_1 layer was input into the K-NN algorithm that was trained, and if the maximum number of predicted labels was below the threshold, it was classified as an anomaly data. Using user foot pressure data for experimentation to find the optimal Threshold value, the value was set to 60 as shown in Fig. 7. The accuracy for the train dataset was 81.8%, for the validation dataset was 80.7%, and for the test dataset was 79.7%.

$$x_1, \ldots, x_{128} = \frac{x_1, \ldots, x_{128}}{(\sum_{i=1}^{128} |x_i|^2)^{\frac{1}{2}}} \tag{8}$$

CBIR with K-nearest neighbor

CBIR with K-NN [21, 22] is a method that adds the K-NN algorithm to the CBIR method used in existing image searches to detect anomaly data. The method involves training the K-NN algorithm with the results of the convolution layer, with the aim of detecting anomaly data. In this paper, the last convolutional layer of LeNet-5 was used to compare the results, as this allows for resource optimization and faster processing on edge node. The training method was the same as the Auto-Encoder with K-NN method, with the exception that the results of the convolutional layer were used as the input layer. Experimental results using user foot pressure data showed that the optimal threshold value was 69, as shown in Fig. 7, with an accuracy of 62.6% for the train dataset, 61.3% for the validation dataset, and 67.4% for the test dataset.

LeNet-5 based multi-image

The proposed model for anomaly detection in this paper is as follows. The collected foot pressure data of the users as they walk, as shown in Fig. 3, consists of an average

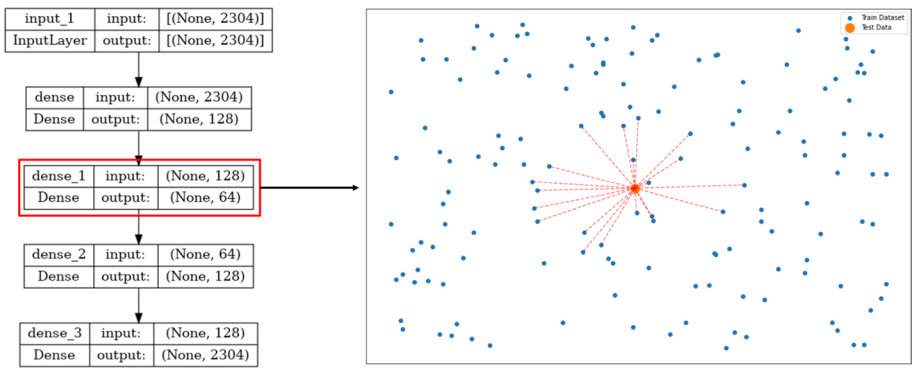


Fig. 9 Auto-encoder with K-nearest neighbor

of three complete data. These three complete data are preprocessed and normalized to determine whether the user is abnormal or not. The determination method is shown in Fig. 10. The obtained three data are individually input into the LeNet-5 model trained to recognize 10 users, resulting in obtaining results similar to Eq. (9).

$$Output_i = x_{i,1}, x_{i,2}, x_{i,3}, ..., x_{i,10} \tag{9}$$

$$Average_j = \frac{1}{3}\sum_{k=1}^{3} x_{k,j} \tag{10}$$

In this paper, by inputting 3 images into the model trained to recognize 10 users, Eq. (9) is given with i = 1, 2, 3 and Eq. (10) is given with j = 1, 2,..., 10. In this process, the class averages are calculated using the results of the same class from the 3 image outputs as described in Equation x. If the maximum value among the class averages is equal to or above the threshold, it indicates that all 3 images have been predicted as belonging to the same class, suggesting that it corresponds to an existing user. Conversely, if the maximum value is below the threshold, it implies that there are predictions of different classes among the 3 images, indicating an new user. Experimental results using user foot pressure data showed that the threshold was 78%, with 97.7% accuracy for the train dataset, 92.7% for the validation dataset, and 89.2% for the test dataset.

Experimental results

Experimental results on the training server

This paper evaluated the proposed model and the existing model using three different datasets with varying similarities: Fashion-MNIST dataset, Digit-MNIST dataset, and Foot Pressure dataset, in that order of similarity. Each dataset was divided into Training Dataset, Validation Dataset, and Test Dataset. The results are presented in Table 1. The Fashion-MNIST dataset, which has the lowest similarity, showed good performance with an average accuracy of 83% and good AUC in the existing anomaly detection model. Furthermore, the Digit-Mnist Dataset, which exhibits a similarity similar to the Fashion-Mnist Dataset, showed a slight decrease in average accuracy and AUC. However, the Foot Pressure dataset, which has a high similarity, experienced a significant decrease in average accuracy by 22% and a sharp decline in AUC. Therefore, the existing model can guarantee good performance for datasets with low similarity, but not for datasets with high similarity [29, 30]. This is because lower similarity requires fewer features and can show better performance, while higher similarity requires more detailed and complex features. Therefore, it is possible to improve

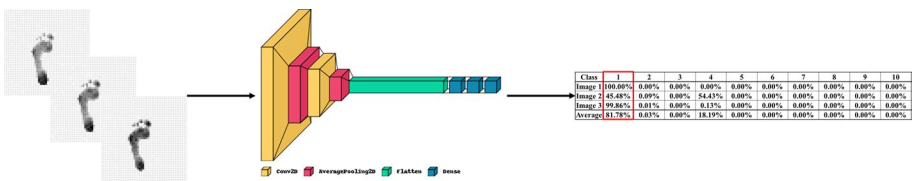


Fig. 10 LeNet-5 based multi-image

Table 1 Experiment result

Dataset	Algorithm	Threshold	Train accuracy (%)	Validation accuracy (%)	Test accuracy (%)	F1-Score	AUC (%)
Digits Mnist	Isolation forest	0.03	79.5	75.6	78.9	78.9	79
	SVM	–	89.8	88.2	78.6	78.6	79.5
	MLP-based Auto-Encoder	4.22	96.2	89.4	92.5	92.5	92.7
	CNN-based Auto-Encoder	3.79	99.7	94.8	95.2	95.2	95.2
	Auto-Encoder with K-Nearest Neighbor	63	79	76.4	72.3	72.3	72.5
	CBIR with K-Nearest Neighbor	95	80	80.6	79.6	79.6	79.6
	ReNet-5 based Multi-Image	88	100	95	92	92.7	92.7
Fashion Mnist	Isolation Forest	− 0.02	94.4	95.8	87.2	87.2	88.03
	SVM	–	90.2	91.4	83.9	83.9	84.3
	MLP-based Auto-Encoder	4.07	95.6	92.6	87.7	87.7	87.9
	CNN-based Auto-Encoder	3.77	95.9	94.8	87	87	87.3
	Auto-Encoder with K-Nearest Neighbor	64	81.2	78.8	78.3	78.3	78.3
	CBIR with K-Nearest Neighbor	89	61.8	59.6	73.3	73.3	75.4
	ReNet-5 based Multi-Image	66	92.2	85.4	84.7	84.7	84.7
Foot Pressure	Isolation Forest	0.01	84.6	83	51.9	51.9	53.6
	SVM	–	90.1	91.4	50.2	50.2	50.62
	MLP-based Auto-Encoder	1.13	77	73.9	58.9	58.9	59.8
	CNN-based Auto-Encoder	1.09	77.6	75	56.7	56.7	57.3
	Auto-Encoder with K-Nearest Neighbor	60	81.8	80.7	79.7	79.7	79.7
	CBIR with K-Nearest Neighbor	69	62.6	61.3	67.4	67.4	67.7
	ReNet-5 based Multi-Image	78	97.7	92.7	89.2	89.2	88.95

performance by adjusting the model size, as shown in Fig. 11, but the model size is limited by the computing resources due to the edge node with the user's foot pressure measuring system, not a high-performance server, which is used to judge new users.

However, the evaluation results using the proposed model in this paper show an average accuracy and AUC of 89% across all datasets, regardless of their similarity. Particularly, the Foot Pressure dataset, which has high similarity, exhibits an accuracy of 89% and AUC. Moreover, when examining the ROC curve Fig. 12 of the overall model generated using the high similarity Foot Pressure dataset, it shows superior performance compared to the baseline model. Therefore, we have confirmed that our proposed model can achieve high performance regardless of the dataset's similarity without increasing the model's size.

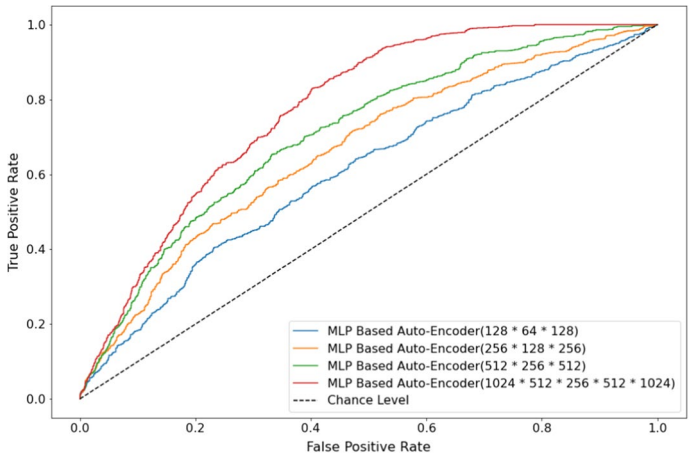


Fig. 11 ROC curve for MLP-based auto-encoder with varying sizes of the hidden layer with foot pressure dataset

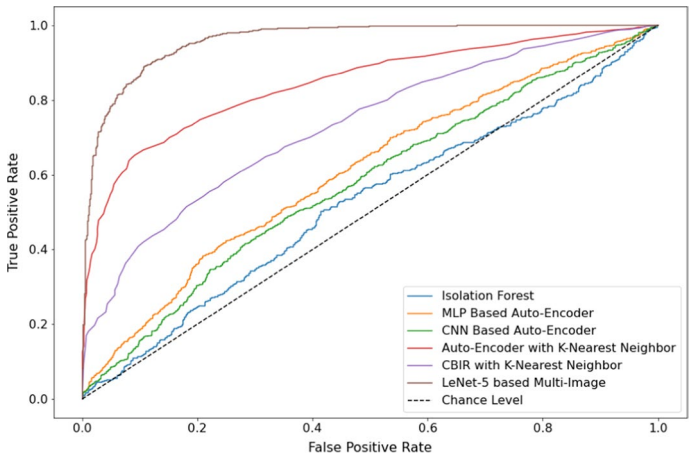


Fig. 12 ROC curve for all models with foot pressure dataset

Experimental results on the edge node

To evaluate the performance from the perspective of the edge node, we selected the proposed model and the top two models based on their average test accuracy across all datasets. Then, we trained each model using the Foot Pressure dataset. Subsequently, we performed pruning and quantization using the TFLM library to create compressed 8-bit integer weights. These compressed weights were loaded onto the flash memory of the edge node, and experiments were conducted directly on the edge node. The MCU (STM-32F207ZG) used in the experiment has 128 Kb of SRAM and 1Mb of flash memory, but since the user's foot pressure measurement system is installed, the model's flash memory must be below 869 Kb and the required memory during operation must be below 90 Kb. Table 2 shows the experimental results, where the MLP-based Auto-Encoder requires low operating memory of about 2 Kb but requires about 304 Kb of flash memory, while the CNN-based Auto-Encoder requires low flash memory of about 78 Kb but requires 90 Kb of operating memory. Due to the limitations of computing resources, it is not

Table 2 Accuracy comparison on STM32F207ZG

Algorithm	Threshold	Train accuracy (%)	Validation accuracy (%)	Test accuracy (%)	TFLM accuracy(%)	Model size (Kb)	Used memory (Kb)	Inference speed (Ms)
MLP-based Auto-Encoder	1.13	77	73.9	58.9	58.7	304.1953125	2.3125	198
CNN-based Auto-Encoder	1.09	77.6	75	56.7	56.2	78.1953125	90	66415
ReNet-5 based Multi-Image	78	97.7	92.7	89.2	85.7	206.390625	16.875	2742

possible to significantly scale up both models, and therefore, the accuracy cannot be improved. If we increase the model size by using more weights, it would result in inference times exceeding 66 s, similar to the Auto-Encoder CNN model. However, it was confirmed that the proposed model using intact data and LeNet-5 allows for anomaly detection within 3 s using 204 Kb of flash memory and 16 Kb of operating memory. Furthermore, by performing Pruning and Quantization for execution on the edge node, there was a loss of approximately 3% in accuracy. However, it demonstrated a good performance of 86% accuracy within 2.7 s, which is considered acceptable. Therefore, the proposed model in this paper requires less memory compared to existing anomaly detection models and shows a performance improvement of 53% in datasets with high similarity. Thus, it can be concluded that this model guarantees high accuracy in similar datasets and is considered the optimal model for detecting new users on the edge node.

Conclusion

This paper proposes a system capable of detecting new users even in datasets with high similarity, such as foot pressure datasets. The system is designed to ensure predictability and enable real-time predictions by being executed on edge nodes with user foot pressure measurement systems, rather than relying on high-performance servers. Therefore, to guarantee high data similarity, it is recommended to avoid using Auto Encoder-based anomaly detection models, which have been widely used recently. Furthermore, due to the execution on edge nodes, the inference speed of models with a large number of weights increases significantly, limiting the possibility of increasing the model size. Therefore, this paper conducted anomaly detection based on the LeNet-5 algorithm, which is an image classification algorithm with a small number of weights. It satisfied the resource limitations of edge nodes, which include flash memory of 869 Kb or less and memory of 90 Kb or less. Also, the paper demonstrated the reliability of the experiments by applying the proposed method to various datasets, not limited to a single dataset. Additionally, the use of the F1 score to select the threshold reduced the risk of decreased accuracy. The experimental results showed that the proposed model achieved an accuracy of over 89% not only on low similarity datasets such as digit MNIST and fashion MNIST but also on a high similarity dataset like foot pressure. By optimizing the model through pruning and quantization using TFLM, there was a slight decrease of approximately 3% in

accuracy. However, the model could make predictions within 2.7 s on an edge node, utilizing 204 Kb of flash memory and 16 Kb of memory. As a result, compared to the existing anomaly detection models, the proposed model achieved a significant reduction of approximately 530% in memory usage and around 40% in flash memory usage on the high similarity foot pressure dataset. Moreover, the accuracy was improved by approximately 53%. The proposed system has the capability to detect abnormal data in real-time on edge nodes. This opens up various possibilities for providing services using foot pressure data, such as non-contact authorization based on foot pressure, or sending alerts to administrators when abnormal users are detected in unauthorized areas. Furthermore, it can be used as a model for detecting new users in a system aimed at automating the addition of new users in future development. This system detects new users at the edge node and sends their data to the training server. The training server performs transfer learning to incorporate the data of the new users and further train the model. The optimized model weights are then transmitted back to the edge node, enabling real-time user addition functionality in the system. To achieve that, it is necessary to improve the inference speed, which is currently a limitation of the existing anomaly detection model. To improve the inference speed, it is necessary to reduce the number of model weights, which may result in a decrease in accuracy. Therefore, in order to improve the inference speed of the model in the future, we plan to enhance the preprocessing system by compressing images and reducing their size, aiming to reduce the model size. We will compare the accuracy and inference speed to evaluate the improvements.

Abbreviations

SVM	Support Vector Machine
K-NN	K-Nearest Neighbor
MLP	Multi-Layer Perceptron
CNN	Convolutional Neural Network
CBIR	Content-Based Image Retrieval
ROC	Receiver Operating Characteristic
AUC	Area Under the ROC Curve
MCU	Micro Controller Unit
TFLM	TensorFlow Lite for Microcontrollers

Acknowledgements
This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF2018R1A6A-1A03025109).

Author contributions
DHH proposed a novel approach, developed the entire system, extracted the experimental results, and wrote the manuscript. SHP assisted with the system development, particularly in the porting and execution on the MCU. SJK provided technical expertise and guidance throughout the research project, serving as a constant guide from start to finish. The author read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 22 March 2023 Accepted: 1 June 2023
Published online: 12 June 2023

References

1. Huh M, Agrawal P, Efros A.A. What makes imagenet good for transfer learning? 2016; arXiv preprint arXiv:1608.08614
2. Zhu F, Zhang X-Y, Wang R-Q, Liu C-L. Learning by seeing more classes. IEEE Trans Pattern Anal Mach Intell. 2022. https://doi.org/10.1109/TPAMI.2022.3225117
3. Powers S, Keselman L. Introspective neural networks
4. Kuzborskij I, Orabona F, Caputo B. From n to n+1: Multiclass transfer incremental learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013;
5. Zhang H, Ding H. Prototypical matching and open set rejection for zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021; pp. 6974–6983
6. Henrydoss J, Cruz S, Rudd E.M, Gunther M, Boult T.E. Incremental open set intrusion recognition using extreme value machine. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017; pp. 1089–1093 https://doi.org/10.1109/ICMLA.2017.000-3
7. John P, Brazzale A.R, Süveges M. Margin-free classification and new class detection using finite dirichlet mixtures. 2021; arXiv preprint arXiv:2103.14138
8. Din SU, Shao J, Kumar J, Mawuli CB, Mahmud SH, Zhang W, Yang Q. Data stream classification with novel class detection: a review, comparison and challenges. Knowl Inf Syst. 2021;63:2231–76.
9. Roh Y, Heo G, Whang SE. A survey on data collection for machine learning: a big data—AI integration perspective. IEEE Trans Knowl Data Eng. 2021;33(4):1328–47. https://doi.org/10.1109/TKDE.2019.2946162.
10. Mahlamäki K, Nieminen M. Analysis of manual data collection in maintenance context. J Qual Maint Eng. 2020;26(1):104–19.
11. Cardoni M, Pau DP, Falaschetti L, Turchetti C, Lattuada M. Online learning of oil leak anomalies in wind turbines with block-based binary reservoir. Electronics. 2021;10(22):2836.
12. Pau D, Khiari A, Denaro D. Online learning on tiny micro-controllers for anomaly detection in water distribution systems. In: 2021 IEEE 11th International Conference on Consumer Electronics (ICCE-Berlin), 2021; pp. 1–6 . https://doi.org/10.1109/ICCE-Berlin53567.2021.9720009
13. Chabchoub Y, Togbe MU, Boly A, Chiky R. An in-depth study and improvement of isolation forest. IEEE Access. 2022;10:10219–37. https://doi.org/10.1109/ACCESS.2022.3144425.
14. Tokovarov M, Karczmarek P. A probabilistic generalization of isolation forest. Inf Sci. 2022;584:433–49.
15. Pang J, Pu X, Li C. A hybrid algorithm incorporating vector quantization and one-class support vector machine for industrial anomaly detection. IEEE Trans Ind Inf. 2022;18(12):8786–96. https://doi.org/10.1109/TII.2022.3145834.
16. Ji Y, Lee H. Event-based anomaly detection using a one-class svm for a hybrid electric vehicle. IEEE Trans Veh Technol. 2022;71(6):6032–43. https://doi.org/10.1109/TVT.2022.3165526.
17. Chow JK, Su Z, Wu J, Tan PS, Mao X, Wang Y-H. Anomaly detection of defects on concrete structures with the convolutional autoencoder. Adv Eng Inf. 2020;45: 101105.
18. Liu J, Song K, Feng M, Yan Y, Tu Z, Zhu L. Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection. Opt Lasers Eng. 2021;136: 106324.
19. Guo J, Liu G, Zuo Y, Wu J.: An anomaly detection framework based on autoencoder and nearest neighbor. In: 2018 15th International Conference on Service Systems and Service Management (ICSSSM), 2022; pp. 1–6 (2018). https://doi.org/10.1109/ICSSSM.2018.8464983
20. Wang X, Zheng Q, Zheng K, Sui Y, Cao S, Shi Y. Detecting social media bots with variational autoencoder and k-nearest neighbor. Appl Sci. 2021;11(12):5482.
21. Sampathila N, Martis RJ. Computational approach for content-based image retrieval of k-similar images from brain mr image database. Expert Syst. 2022;39(7):12652.
22. Praveena HD, Guptha NS, Kazemzadeh A, Parameshachari B, Hemalatha K. Effective cbmir system using hybrid features-based independent condensed nearest neighbor model. J Healthc Eng. 2022;2022:3297316.
23. Ioannou C, Vassiliou V. Classifying security attacks in iot networks using supervised learning. In: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), 2019; pp. 652–658 https://doi.org/10.1109/DCOSS.2019.00118
24. Jia W, Shukla R.M, Sengupta S. Anomaly detection using supervised learning and multiple statistical methods. In: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019; pp. 1291–1297 https://doi.org/10.1109/ICMLA.2019.00211
25. Wang X, Yang I, Ahn S-H. Sample efficient home power anomaly detection in real time using semi-supervised learning. IEEE Access. 2019;7:139712–25.
26. Zhang S, Ye F, Wang B, Habetler T.G. Semi-supervised learning of bearing anomaly detection via deep variational autoencoders. 2019; arXiv preprint arXiv:1912.01096
27. Schlegl T, Seeböck P, Waldstein SM, Langs G, Schmidt-Erfurth U. f-anogan: fast unsupervised anomaly detection with generative adversarial networks. Medical Image Anal. 2019;54:30–44.
28. Audibert J, Michiardi P, Guyard F, Marti S, Zuluaga M.A. Usad: Unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020; pp. 3395–3404
29. Roady R, Hayes TL, Kemker R, Gonzales A, Kanan C. Are open set classification methods effective on large-scale datasets? PLoS ONE. 2020;15(9):0238302.

30. Linderman R, Zhang J, Inkawhich N, Li H, Chen Y. Fine-grain inference on out-of-distribution data with hierarchical classification. 2022; arXiv preprint arXiv:2209.04493
31. Wang W, Yang Y, Xiong Z, Niyato D. Footstone of metaverse: a timely and secure crowdsensing. IEEE Network 2023;
32. Hu Y, Li H, Chang Z, Han Z. End-to-end backlog and delay bound analysis for multi-hop vehicular ad hoc networks. IEEE Trans Wirel Commun. 2017;16(10):6808–21.
33. Geetha R, Suntheya A, Srikanth GU. Cloud integrated iot enabled sensor network security: research issues and solutions. Wirel Pers Commun. 2020;113:747–71.
34. Heo KH, Jeong SY, Kang SJ. Real-time user identification and behavior prediction based on foot-pad recognition. Sensors. 2019;19(13):2899.
35. Wei G, Li G, Zhao J, He A. Development of a lenet-5 gas identification cnn structure for electronic noses. Sensors. 2019;19(1):217.
36. Hsu C-Y, Chien J-C. Ensemble convolutional neural networks with weighted majority for wafer bin map pattern classification. J Intell Manuf. 2022;33(3):831–44.
37. Shaheen M, Khan R, Biswal RR, Ullah M, Khan A, Uddin MI, Zareei M, Waheed A. Acute myeloid leukemia (aml) detection using alexnet model. Complexity. 2021;2021:1–8.
38. Xu X, Ding Y, Hu SX, Niemier M, Cong J, Hu Y, Shi Y. Scaling for edge inference of deep neural networks. Nat Electron. 2018;1(4):216–22.
39. Liang T, Glossner J, Wang L, Shi S, Zhang X. Pruning and quantization for deep neural network acceleration: a survey. Neurocomputing. 2021;461:370–403.
40. Geng C, Huang S-J, Chen S. Recent advances in open set recognition: a survey. IEEE Trans Pattern Analy Mach Intell. 2020;43(10):3614–31.
41. Xie H, Du Y, Yu H, Chang Y, Xu Z, Tang Y. Open set face recognition with deep transfer learning and extreme value statistics. Int J Wavelets Multiresolut Inf Process. 2018;16(04):1850034.
42. Khandelwal P. Which algorithm takes the crown: Light gbm vs xgboost? Analytics Vidhya 2017; 12
43. Koleini M. Performance improvement of XGBoost and LightGBM when deploying on AWS Graviton3 2022; https://community.arm.com/arm-community-blogs/b/infrastructure-solutions-blog/posts/xgboost-lightgbm-aws-graviton3
44. SHUKLA L. Battle of the Boosting Algos: LGB, XGB, Catboost 2019; https://lavanya.ai/2019/06/01/battle-of-the-boosting-algorithms/
45. Moon A, Zhuo X, Zhang J, Son S.W, Jeong Song Y. Anomaly detection in edge nodes using sparsity profile. In: 2020 IEEE International Conference on Big Data (Big Data), 2020; pp. 1236–1245 https://doi.org/10.1109/BigData50022.2020.9377757
46. Jiang J, Liu F, Liu Y, Tang Q, Wang B, Zhong G, Wang W. A dynamic ensemble algorithm for anomaly detection in iot imbalanced data streams. Comput Commun. 2022;194:250–7.
47. Liu H, Yu C, Wu H, Duan Z, Yan G. A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting. Energy. 2020;202: 117794.
48. Shao Z, Zhang Z, Wei W, Wang F, Xu Y, Cao X, Jensen C.S. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. 2022; arXiv preprint arXiv:2206.09112
49. Deng L. The mnist database of handwritten digit images for machine learning research. IEEE Signal Process Mag. 2012;29(6):141–2.
50. Xiao H, Rasul K, Vollgraf R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. 2017; arXiv preprint arXiv:1708.07747
51. David R, Duke J, Jain A, Janapa Reddi V, Jeffries N, Li J, Kreeger N, Nappier I, Natraj M, Wang T. Tensorflow lite micro: embedded machine learning for tinyml systems. Proc Mach Learn Syst. 2021;3:800–11.
52. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–30.

Publisher's Note

Dong Hyuk Heo is currently pursuing a Ph.D. in Electronic Engineering from Kyungpook National University, Daegu, Republic of Korea. His field of research is "AI Embedded System" Other areas of specialization edge artificial intelligence, include Internet of Things, Real-time System, and Deep Learning. Mr. Dong Hyuk Heo has publications in SCI indexed journals and Conference.

Sung Ho Park received the Ph.D. degrees in School of Electronics Engineering from the Kyungpook National University, Daegu, Republic of Korea, in 2011. He is currently a research professor at a Korea Government-Funded Next-Generation Software Platform Research Center, called Center of Self-Organizing Software. His research interests include operating systems, embedded systems, real-time systems, wireless sensor networks, IoT, edge AI.

Soon Ju Kang (Member, IEEE) received the Ph.D. degree in computer science from the Republic of Korea Advanced Institute of Science and Technology, Daejon, South Korea, in 1995. From 1985 to 1996, he was with the Korea Atomic Energy Research Institute, Daejeon, as a Member of Research Staff and a Head of the Computing and Information Research Department. Since 1996, he has been a Faculty Member with the IT College, Kyungpook National University, Daegu, South Korea. Meanwhile, he was also with the University of Pennsylvania, Philadelphia, PA, USA, as a Visiting Research Faculty from 2000 to 2001 and in 2007. Since

2020, he has been serving as the Dean of the IT College, Kyungpook National University. He is the Director of the Korea Government-Funded Next-Generation Software Platform Research Center, called Center of Self-Organizing Software Platform (http://www.csosp.org), Kyungpook National University. He wrote seven books related to embedded real-time systems as a Coauthor and published about 200 technical papers in regular international journals and related conferences. His current research interests include the self-organizing software platform for embedded real-time systems and edge artificial intelligence, Internet of Things, distributed object technology, and software engineering for embedded real-time systems. Dr. Kang is currently a member of ACM.