RESEARCH ARTICLE

Multivariate Time Series Open-Set Recognition Using Multi-Feature Extraction and Reconstruction

HYEREYONG OH AND SEOUNG BUM KIM
School of Industrial and Management Engineering, Korea University, Seoul 02841, South Korea
Corresponding author: Seoung Bum Kim (sbkim1@korea.ac.kr)

This work was supported in part by the Brain Korea 21 FOUR, the Ministry of Science and ICT (MSIT) in Korea under the ITRC Support Program supervised by the Institute for Information Communication Technology Planning and Evaluation, under Grant IITP-2020-0-01749; and in part by the National Research Foundation of Korea Grant through the Korea Government under Grant RS-2022-00144190.

ABSTRACT In real-world classification tasks, deep neural networks show innovative performance in various fields. However, traditional classification methods are constructed based on a set of predefined classes and force unknown classes that determine their categorization into one of the predefined classes. This problem is addressed by the research field known as open-set recognition. Existing open-set recognition methods claim that the unique features of unknowns cannot be maintained by using only the final features. In other words, the various feature extraction methods should be considered to effectively reflect the characteristics of unknowns. In this study, we propose an open-set recognition model equipped with multi-feature extraction for multivariate time series data. The results of experiments with various multivariate time series datasets indicate that the proposed method shows improved capability to detect unknown classes while maintaining good predictive performance.

INDEX TERMS Open-set recognition, reconstruction error, multichannel signal, multisensor signal, multivariate time series.

I. INTRODUCTION
Advances in sensor technology allow the real-time collection of large amounts of time series data. Time series data collected from more than one sensor are termed multivariate time series (MTS). For example, human activity recognition data are collected from several sensors (e.g., accelerometers and gyroscopes) to classify different human activities. Their classification is one of the important problems in analyzing MTS data [1], [2]. MTS classification is a supervised learning problem designed to assign each MTS sample to predefined classes.

Time series classification is considered a difficult problem because temporal correlation must be considered. That is, the fact that past information affects the present makes time series data more difficult to handle than other data. Furthermore, MTS classification is more challenging than univariate classification because classifiers should consider several time series variables simultaneously [3]. Despite such difficulties, MTS classification has shown its usefulness across various fields, such as road environment monitoring [4], bearing fault detection [5], and human activity recognition [6]. Many approaches have attempted to achieve high performance for MTS classification problems. Among them, deep learning-based methods have achieved significant performance by their ability to extract features containing useful information from the input data [1]. A convolutional neural network (CNN) is considered a suitable structure for feature extraction for MTS because sliding filters can take into account the characteristics of multiple time series [2], [7].

Although deep learning algorithms have achieved high classification accuracies, they have an obvious limitation. Traditional classifiers are based on an assumption that the...
FIGURE 1. Comparison between a traditional classifier and open-set recognition when the predefined classes are A, B, and C.

training and testing data are drawn from the same distribution. As a result, novel classes absent in the training data are forced into categories belonging to the predefined classes [8]. In addition, because the features extracted from traditional deep learning models focus on differentiating between the known classes, these features may not contain aspects that facilitate the distinction between known and unknown classes [9].

To overcome these issues, open-set recognition algorithms allow the adaption of traditional classifiers for suitability to open-set scenarios. Figure 1 shows a comparison of a traditional classification and open-set recognition. The predefined classes are labeled A, B, and C. When a sample belonging to Class Z appears in the testing phase, a traditional classifier predicts this novel class is one of the known classes (here Class A). However, under open-set recognition, this sample can be identified as unknown. Samples detected as unknown are new patterns that were not present in the training data. Therefore, they can be used to update the model after relabeling by humans.

The state-of-the-art approaches of open-set recognition have proved their superiority mostly through experiments with image data, such as MNIST and CIFAR10 [10]. Despite the existence of the same open-set problems in MTS data, few efforts have been made to address them. For MTS classification, we need a proper structure to extract the features of MTS that are helpful for detection of unknowns. Some studies of the open-set recognition of MTS data have been proposed in specific areas including human activity recognition [11].

The main purpose of this study is to develop an effective open-set algorithm for analyzing general MTS data. Specifically, we propose to use multi-feature extraction and reconstruction-based thresholds to construct an open-set recognition algorithm for MTS data. The purpose of this study is to prove that more informative features and stricter criteria to detect unknowns can enhance performance in detecting these unknowns while also preserving predictive performance. The proposed method is equipped with various layers so as to reflect diverse information in the final features. This structure allows the final features to contain global features, channel-wise features, and latent representations so that these varied features that contain the unique characteristics of the unknowns can also be used for detection of unknowns. Moreover, for robust detection of unknowns, the maximum reconstruction errors of the training samples of each known class are used as criteria for this detection.

The main contributions of this study can be summarized as follows:

1) The proposed method can effectively extract the complex features of MTS data. We focus on how to extract their plentiful features and improve the usefulness of those features for robust detection of unknowns. The proposed algorithm is composed of manifold layers such as a global 1D CNN, a channel-wise 1D CNN, and latent representations. These structures provide abundant information to the final features, which are then used for detection of unknowns.

2) We define the thresholds of each known class tighten criteria for detection of unknowns. The proposed
method is trained to minimize the sum of the classification errors and reconstruction errors. In the process of detecting unknowns, the reconstruction errors of the training data are used to determine the thresholds of the known classes. By applying such reconstruction-based thresholds, the proposed method improves detection of unknowns while maintaining good predictive performance.

The remainder of this paper is organized as follows. Section II reviews existing studies of time series classification and open-set recognition. Section III demonstrates the baseline model and our proposed method in detail. Section IV describes the experimental setups and the results of applying the proposed method to various MTS classification datasets. Finally, Section V presents our concluding remarks.

II. RELATED WORKS

A. TIME SERIES CLASSIFICATION

To conduct the time series classification, the characteristics of time series were considered when extracting the features to be used. Shapelets, which are local time series patterns containing a discriminative quality, are a popular feature extraction of time series data. However, identifying shapelets is time consuming. To speed identification, [12] introduced an approach using a number of random shapelets. To derive good performance while reducing time, shapelet-based decision trees were proposed based on random choices of instances and shapelets [13].

With advances in sensor data collection technology, analysis of MTS data has become important in various domains. For MTS classification, classifiers should consider the relationships of multiple attributes [14]. Deep learning models have been proposed to effectively extract complex time series features, and those models have exhibited good performance across a variety of fields, including human activity recognition [6], industrial equipment monitoring [15], and healthcare [16]. For time series classification, multilayer perceptrons (MLPs) were used for end-to-end learning. However, they have the drawback of losing time series information. To overcome this problem, CNNs have been used because they can capture sequential features by learning spatial filters [7], [17]. To extract effective features from MTS for failure detection in semiconductor manufacturing, a CNN model that consists of stacked structures with convolution, activation, and pooling layers was proposed [18]. Multi-domain feature fusion by CNN was proposed to improve monitoring efficiency and predictive accuracy in monitoring tool wear [19].

Recurrent neural networks (RNNs) are also common deep learning models for learning from sequential information. Shallow RNNs composed of several independent RNNs have been proposed to capture long time-series information. This approach achieved improved inference time over standard RNNs for time series classification tasks on tiny devices [20]. To alleviate the vanishing gradient problem of RNNs, an echo state network (ESN) was proposed. ESNs are based on a sparsely connected random RNN, and only the weights between the hidden layer and the output layer are trained [21]. The augmented structure of convolutional networks and long short-term memory achieved good performance in classifying time series sequences. Furthermore, adding an attention mechanism led to improved performance [22], [23]. However, although these classifiers show innovative predictive performance, they can achieve successful results only for the predefined classes presented in the training data. When these classifiers encounter situations in which unknown classes occur in the testing data, these unknown classes are forced into being classified into one of the known classes. In other words, existing MTS classification studies have focused on feature extraction to improve only the predictive performance for predefined classes to the exclusion of seeking to use feature extraction for detection of unknown classes.

B. DEEP NEURAL NETWORK-BASED OPEN-SET RECOGNITION

Deep neural network (DNN)-based open-set recognition approaches have been actively presented with the recent advent of deep learning [10]. As the first attempt at a DNN-based approach, OpenMax was proposed; it used the concept of nearest class mean to adapt the softmax layer and enable it to calculate the probability of an unknown [8]. To develop the distance-based measurement of OpenMax, a more effective representation was used to make samples from the same class closer to each other and push those from different classes away from each other [24]. Reconstruction learning was used during training to use latent representations and final features together for robust detection of unknowns [9]. For the open-set recognition of text documents, a one-versus-the-rest layer was used instead of the softmax layer. Moreover, the probability-based thresholds of each class were used as the criteria for detection of unknowns [25].

Apart from methods transforming the softmax layer, some algorithms proposed open-set frameworks by changing model structure or learning loss. Learning data placeholders and extra dummy classifiers address the drawbacks of threshold-based algorithms and effectively separate the known from the unknown [26]. To achieve detection of unknowns, extracting high-level features from a probabilistic ladder architecture and learning the conditional distributions have been proposed [27]. To gather samples of the same class together and push samples of different classes apart, a deep metric learning-based approach was proposed for robotic applications [28]. In an attempt to speed the process of detecting unknowns, an approach using a micro-clustering method and Jensen-Shannon (JS) divergence was proposed [29]. The effectiveness of these algorithms was proven primarily with image data. However, despite their usefulness, only few efforts have been made to address open-set recognition for MTS data. Recently, [30] proposed an open-set recognition algorithm referred to as Open Set InceptionTime (OS-InceptionTime) for MTS data using unknown-detection
criteria defined by dynamic time warping distance and cross-correlation. This algorithm requires both the raw input data and model output features for both classification and detection of unknowns. In the present study, we propose an open-set recognition algorithm that can differentiate the unknowns from the known by extracting useful MTS features.

III. PROPOSED METHOD

A. BASELINE MODEL

We used classification-reconstruction learning for open-set recognition (CROSIR) [9] as the baseline model for the proposed method. Although CROSIR was developed for image data, to accommodate MTS data we changed its structure in the baseline model to one-dimensional (1D) CNN. 1D CNN is commonly used for the analysis of MTS data because it can extract the features of time series by moving filters in one direction for time [7].

As shown in Figure 2, the baseline model uses two types of features together for detection of unknowns. One is the global features of the input samples. The other is their latent representations. To extract the global features, the baseline model has three convolutional blocks. Each convolutional block consists of two convolutional layers and a dropout. These convolutional layers are followed by batch normalization and a rectified linear unit (ReLU) activation function. The convolutional layers can extract features while maintaining the positioning information of time series data because their filters extract information for a certain length of time. Features coming from all the blocks pass through fully connected layers. This process yields the final features with a length equal to the number of predefined classes as the output of the model. These features are used for classification between known classes and detection of unknowns. The latent representations are formed at each end of the convolutional blocks. Their row-wise maximum values and the final features are concatenated to be provided to the unknown detector.

The baseline model uses the latent features not only for detection of unknowns but also for the reconstruction of input samples. In the classification task, final features are learned by focusing on being correctly classified as one of the known classes. However, it is hard to distinguish between knowns and unknowns based on these final features. Therefore, in open-set recognition, use of the classification and reconstruction of input samples together is useful for effective detection of unknowns. For reconstruction, latent representations are commonly used [9]. For a network consisting of a total of $T$ convolutional blocks, the process of reconstruction of an input sample $x$ is formulated as follows:

$$x_{t+1} = f_t(x_t),$$
$$z_t = g_t(x_{t+1}),$$
$$\tilde{x}_t = \begin{cases} F_t(\tilde{g}_t(z_t)) & \text{if } t = T \\ F_t(\tilde{x}_{t+1} + \tilde{g}_t(z_t)) & \text{otherwise,} \end{cases}$$

where $f_t$ represents the $t$-th convolutional block of the main network, and $g_t$ represents a convolutional layer for generating latent representation $z_t$, $\tilde{g}_t$ is a convolutional layer for the transformation of latent representation $z_t$ into the original dimension. $F_t$ is transposed convolution operation for the reconstruction sample $\tilde{x}_t$.

The training errors are measured as the sum of the cross-entropy loss and reconstruction errors. The loss function is formulated as follows:

$$L_{\text{train}}(y, \hat{y}, x, \tilde{x}) = -\sum_{i=1}^{C} y_i \log \hat{y}_i + \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (x_{ij} - \tilde{x}_{ij})^2,$$

where $y$ and $\hat{y}$ are the actual and predicted probabilities, respectively. $\tilde{x}$ is the reconstruction of an input sample $x$ whose shape is $(M, N)$. $C$ represents the number of known classes.

The baseline model adopts OpenMax [8], one of the most representative open-set approaches, to convert the classifier into an open-set classifier. To overcome the limitation of softmax which can only classify predefined classes, OpenMax recalibrates the final features based on extreme value theory. For this, a value representing the probability of being an unknown is added to the final features. This value is calculated using a Weibull distribution because statistical extreme-value theory considers it appropriate for this purpose [9].

A Weibull distribution is estimated for each class, and the parameters of the distribution are derived from outlier samples. The outlier samples denote training samples featuring the largest distance from the mean vector of the class; their Euclidean distances from the mean vector of the true class are used to estimate the Weibull distribution. To calculate these distances, the baseline model uses the concatenated features of the final features and the row-wise maximum values of the latent representations. When the number of known classes is $N$, $N + 1$ denotes an unknown class. After estimating a Weibull distribution $\text{Weibull}_c$ of known Class $c$, the process of detection of unknowns and the final prediction of a new input sample can be formulated as follows:

$$\hat{y}_c = \begin{cases} y_c(1 - w_c) & \text{if } c \neq N + 1 \\ \sum_{c=1}^{N} y_c w_c & \text{otherwise,} \end{cases}$$
$$y^* = \arg\max_{i \in \{1, \ldots, N+1\}} \hat{y}_i,$$
where \( w_c \) denotes an outlier score of Class \( c \) derived from \( \text{Weibull}_c \). \( y_c \) is the probability of belonging to Class \( c \) calculated after passing through a softmax layer. \( \hat{y}_c \) is the calibrated probability of Class \( c \) according to the OpenMax algorithm. The class with the maximum value in \( \hat{y} \) is the final prediction \( y^* \). That is, if the value corresponding to \( N + 1 \) is the largest, the sample is classified as unknown. In addition, OpenMax uses the following probability-based threshold \( \epsilon \) for detection of unknowns:

\[
y^* = \begin{cases} 
  \text{"unknown"} & \text{if } \max \hat{y} < \epsilon \\
  \arg\max_{l\in\{1,...,N\}} \hat{y}_l & \text{otherwise.}
\end{cases}
\] (7)

**B. MEROS**

We now present the proposed open-set recognition framework called multi-feature extraction and reconstruction learning for open-set recognition (MEROS). MEROS is trained by the same loss function as the baseline model and adopts OpenMax for detection of unknowns. However, in contrast with the baseline model, MEROS is equipped with many more layers so that the final features that are the input of OpenMax can contain more useful information. Furthermore, reconstruction-based thresholds are additionally used in MEROS for robust detection of unknowns. Figure 3 demonstrates the overall structure of the proposed method. After feature-extraction, the final features are fed into the unknown detector that performs both classification and unknown detection. Because the training loss includes the cross entropy (Equation (4)), the input samples are classified into the class with the highest probability value. Further, the classification results are updated through the unknown-detection process to detect unknowns.
FIGURE 4. Description of multi-feature extraction of the proposed MEROS compared with the baseline model.

1) MULTI-FEATURE EXTRACTION

Deep learning models can perform well when they capture informative features. Therefore, it is crucial to make the structure of a model conducive to the extraction of useful features. Especially for open-set recognition, it is necessary to extract not only features for known-class classification but also features that can be used to distinguish between knowns and unknowns. The proposed MEROS is equipped with numerous layers for effective feature extraction so that useful information is available for detection of unknowns. Figure 4 compares the baseline model and MEROS equipped with multi-feature extraction. For an input MTS sample whose dimension is \( D \), \( x_t \) denotes the features extracted by the \( t \)-th convolutional block of global 1D CNN. \( x^{d}_{t} \) represents the univariate-specific features from the \( t \)-th convolutional block of the \( d \)-th channel-wise 1D CNN. \( z_t \) is the \( t \)-th latent representations, and \( \tilde{x}_t \) is the reconstructed features of \( x_t \) from \( z_t \). \( y \) is the final features used for detection of unknowns.

Some MTS classification studies have used univariate-specific features despite input data composed of several dimensions. For example, to consider the features of each dimension, an ensemble algorithm has been proposed using models independently created for each dimension [31]. Extracting univariate-specific features has the drawback that the unique features of each dimension can be extracted without consideration of their correlations. To take advantage of more diverse features, MEROS is equipped with channel-wise 1D CNN in addition to the main network, global 1D CNN.

Global 1D CNN captures the overall features that consider the relationships between the dimensions. Meanwhile, channel-wise 1D CNN extracts the features of each dimension to use univariate-specific features effectively. Each output layer of the channel-wise 1D CNN is linked at the end. Then, the concatenated features of the final layers of both global 1D CNN and channel-wise 1D CNN are provided for detection of unknowns.

Table 1 shows the detailed hyperparameters of the proposed model. It is worth noting that these hyperparameters can be used to general MTS data, but some adjustments are required when the length of data is short. For setting the parameters, we examined different values of the parameters and found one that yielded the best performance of validation data from known classes.

The usage of reconstructive latent representations in the process of detecting unknowns allows a detector of unknowns to exploit broad features [9]. Both the proposed and the baseline models use latent representations when detecting unknowns. However, although the baseline model uses the concatenated features of latent representations and final features, the proposed method is transformed to make the final features contain as much latent information as possible.

We contend that providing as many latent representations as possible to the final features improves unknown-detection performance. Thus, we add max pooling layers with small window size for latent representations. Max pooling means that the filters move through particular sections and select the maximum value within the range called window size [32]. Many deep learning algorithms use max pooling because it
can extract important features while reducing the dimension of feature maps. As shown in Figure 5, row-wise maximum values of latent representations are extracted in the baseline model. That is, if the height of latent representations is \( k \), only \( k \) values are selected, and the rest are discarded. However, the proposed MEROS is equipped with max pooling with much smaller window size than the baseline model. This allows MEROS to extract \( l \) values of latent representations, which are much larger than \( k \), as shown in Figure 5. Therefore, more information can be contained in the final features. In brief, the multi-feature extraction of MEROS consists of channel-wise 1D CNN and max pooling latent representations. Both components are aimed at forming useful features for detection of unknowns. Many parameters extract the features of input samples in different ways, and all these features are considered when the final features are created. The final features contain various information coming from multiple layers including channel-wise features, global features, and latent representations. Thus, these features can be useful for the intra-class classification of known classes as well as the distinction between knowns and unknowns.

2) UNKNOWN DETECTION FOR MEROS

For tighter criteria allowing more effective unknown detection, we used reconstruction-based thresholds immediately before final predictions. Reconstruction errors are commonly used in unsupervised approaches, especially in anomaly detection. Networks that are well trained to reconstruct only normal samples produce more reconstruction errors in reconstructing anomalies than they produce with normal data reconstruction [33]. Thus, anomalies can be detected by applying certain thresholds estimated from reconstruction errors, and these criteria can be used for detection of unknowns. In image segmentation, thresholds based on reconstruction errors have been used to define which pixels are known and which are unknown [34]. Similar to this concept, we set thresholds for each known class. If the reconstruction error of the input sample exceeds the threshold of the predicted class in the detection of unknowns phase, its final prediction becomes unknown. Note that only training data are used to define the thresholds. For an original input sample \( x \) whose shape is \((M, N)\) and a reconstructed sample \( \tilde{x} \), the threshold \( \delta_c \) of the known Class \( c \) is derived as follows:

\[
\delta_c = \max\left\{ \frac{1}{M \times N} \sum_{i}^{M} \sum_{j}^{N} (x_{ij} - \tilde{x}_{ij})^2 | x \in c \right\}. \tag{8}
\]

We defined the maximum reconstruction error of each class as its threshold so as to minimize the number of known samples misclassified as unknown. The process for detection of unknowns for a new input sample whose reconstruction error is \( r \) can be summarized in the following steps: The final prediction Class \( c \) is determined by the OpenMax algorithm using Equations (5), (6), and (7). Then, one more detection process for unknowns is conducted using the reconstruction-based threshold of Class \( c \). After these two steps for detection of unknowns, the final prediction \( y^* \) is obtained as follows:

\[
y^* = \begin{cases} 
c & \text{if } r \leq \delta_c 
\text{"unknown"} & \text{otherwise.}
\end{cases} \tag{9}
\]

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

For our experiments, we used MTS classification datasets from the University of East Anglia (UEA) archive [35] and the University of California, Irvine (UCI) Machine Learning Repository. Datasets with more than three classes were used because at least one class had to be set as unknown. We set an equal ratio of known classes to unknown classes. All models were programmed in Python using PyTorch. All experiments were conducted on a personal computer equipped with an Intel Core i7-10700 CPU @ 2.90GHz, 32-GB DDR4 RAM 3200 MHz, and NVIDIA GeForce RTX 2060 SUPER. OpenMax is used for detection of unknowns in both the baseline and proposed models. Therefore, hyperparameters of OpenMax such as threshold and tail size were set the same for both models. The threshold value, which is the unknown-detection criterion compared with the probability of belonging to each known class, was set to zero. That is, when the probability of being unknown was greater than the probabilities belonging to other known classes, the new sample was detected as unknown. The tail size for the number of outlier samples to estimate the parameters of the Weibull distribution was set to 20, which is the value defined in the original paper. However, for datasets with 30 or fewer training samples per known class, the tail size was set to three because using too many samples to estimate the extreme distribution tended to degrade performance. In the proposed model, a reconstruction-based threshold for tighter criteria of detecting unknowns (as described in Section 3.2.2) is required. This threshold must be defined for each known class. We defined the threshold based on the maximum value
TABLE 2. Descriptions of MTS classification datasets.

| Dataset                 | Number of observations | Number of classes | Number of dimensions | Length | Number of observations per class of training data |
|-------------------------|------------------------|-------------------|----------------------|--------|-----------------------------------------------|
| Epilepsy                | 275                    | 4                 | 3                    | 206    | 27                                           |
| MHEALTH                 | 3,355                  | 12                | 12                   | 100    | 189                                          |
| RacketSports            | 303                    | 4                 | 6                    | 30     | 30                                           |
| SpokenArabicDigits      | 8,798                  | 10                | 13                   | 93     | 528                                          |
| UCI HAR                 | 10,299                 | 6                 | 9                    | 128    | 1,000                                        |
| UWaveGestureLibrary     | 440                    | 8                 | 3                    | 315    | 12                                           |

TABLE 3. Comparison of F1 scores, known classification accuracy, and unknown class accuracy among the baseline, OS-InceptionTime and proposed methods for six datasets.

| Dataset                 | F1 score | Known class accuracy | Unknown class accuracy | F1 score | Known class accuracy | Unknown class accuracy | F1 score | Known class accuracy | Unknown class accuracy |
|-------------------------|----------|----------------------|-----------------------|----------|----------------------|-----------------------|----------|----------------------|-----------------------|
| Epilepsy                | 0.62     | 0.79                 | 0.51                  | 0.63     | 0.79                 | 0.54                  | 0.72     | 0.76                 | 0.70                  |
| (0.15)                  | (0.10)   | (0.28)               | (0.15)                | (0.08)   | (0.29)               | (0.15)                | (0.09)   | (0.25)               | (0.12)                |
| MHEALTH                 | 0.56     | 0.68                 | 0.67                  | 0.51     | 0.89                 | 0.4                   | 0.61     | 0.64                 | 0.80                  |
| (0.06)                  | (0.09)   | (0.12)               | (0.11)                | (0.09)   | (0.16)               | (0.09)                | (0.11)   | (0.12)               | (0.12)                |
| RacketSports            | 0.56     | 0.83                 | 0.35                  | 0.48     | 0.87                 | 0.25                  | 0.61     | 0.44                 | 0.86                  |
| (0.20)                  | (0.09)   | (0.30)               | (0.19)                | (0.09)   | (0.29)               | (0.09)                | (0.13)   | (0.19)               | (0.12)                |
| SpokenArabicDigits      | 0.56     | 0.94                 | 0.41                  | 0.48     | 0.98                 | 0.24                  | 0.69     | 0.89                 | 0.63                  |
| (0.07)                  | (0.03)   | (0.11)               | (0.06)                | (0.01)   | (0.10)               | (0.01)                | (0.04)   | (0.15)               | (0.15)                |
| UCI HAR                 | 0.69     | 0.95                 | 0.38                  | 0.77     | 0.92                 | 0.58                  | 0.76     | 0.93                 | 0.53                  |
| (0.13)                  | (0.05)   | (0.27)               | (0.12)                | (0.05)   | (0.27)               | (0.12)                | (0.04)   | (0.29)               | (0.29)                |
| UWaveGestureLibrary     | **0.65** | **0.52**             | **0.88**             | **0.64** | **0.66**            | **0.72**            | **0.61** | **0.46**            | **0.91**            |
| (0.06)                  | (0.08)   | (0.10)               | (0.12)                | (0.10)   | (0.19)               | (0.05)                | (0.07)   | (0.07)               | (0.07)               |

The macro-averaged F1 score is calculated as the average F1 score of each class so that the unknown class is taken into account along with the known classes when computing the metrics. In other words, the sum of each F1 score is divided by N + 1 when the number of known classes is N.

of the reconstruction errors of each class as the threshold as shown in Equation (8).

B. EVALUATION METRICS

Most open-set recognition studies use the macro-averaged F1 score for evaluation [9], [27]. In general, the model exhibiting the larger F1 score can be considered the better model. However, for open-set recognition, it is important to retain the predictive performance of known classes while simultaneously improving the unknown-detection performance because there is a trade-off between them. In this study, to ensure that the improvement in the F1 score is driven by the improvements in unknown-detection performance, the accuracy of both unknown classes and known classes was considered. The known-class accuracy is the proportion of samples that are correctly predicted as true known class of the samples belonging to known classes. The unknown-class accuracy is the proportion of samples that are correctly detected as unknown of the samples belonging to unknowns. The F1 score is calculated as follows:

\[ F1 - score = \frac{2 \times precision \times recall}{precision + recall}, \]  

where \( precision \) indicates the proportion of samples that are actually true among the samples predicted as true (i.e., \( precision = true \, positive \, / \, (true \, positive \, + \, false \, positive) \)). \( recall \) is the proportion of samples that are correctly predicted as true of the samples whose classes are actually true (i.e., \( recall = true \, positive \, / \, (true \, positive \, + \, false \, negative) \)). The macro-averaged F1 score is calculated as the average F1 score of each class so that the unknown class is taken into account along with the known classes when computing the metrics. In other words, the sum of each F1 score is divided by \( N + 1 \) when the number of known classes is \( N \).

C. RESULTS AND DISCUSSION

We compared the proposed MEROS with the baseline model and OS-InceptionTime [30]. OS-InceptionTime uses an open-set algorithm on general MTS data. This algorithm uses InceptionTime [36] as a feature extractor for classification of known classes. Unlike to the proposed MEROS, detection of unknowns is based on thresholds defined by dynamic time warping distance and cross correlation. That is, OS-InceptionTime detects unknowns based on raw input samples instead of the features from the model.

We evaluated the proposed method by conducting experiments with various MTS classification datasets. Table 2 contains an overview of the datasets used. Each experiment was repeated 20 times. Table 3 shows the average performances with standard deviations given in parentheses. The numbers highlighted in bold indicate the best performance of F1 scores. The proposed MEROS outperformed the compared models in terms of F1 scores for all datasets except UCI HAR and UWaveGestureLibrary. As for unknown-class accuracy, the proposed method outperformed the compared models for all datasets except UCI HAR. It is
TABLE 4. Comparison of results with UCI HAR data in terms of F1 scores, known-class accuracy, and unknown-class accuracy.

| Method                           | F1 score | Known class accuracy | Unknown class accuracy |
|----------------------------------|----------|----------------------|------------------------|
| Baseline Model                   | 0.69     | 0.95                 | 0.38                   |
|                                  | (0.13)   | (0.05)               | (0.27)                 |
| Baseline Model + Multi-Feature Extraction | 0.72     | 0.93                 | 0.46                   |
|                                  | (0.14)   | (0.04)               | (0.32)                 |
| Baseline Model + Reconstruction-Based Thresholds | 0.72     | 0.95                 | 0.44                   |
|                                  | (0.11)   | (0.05)               | (0.24)                 |
| MEROS (Proposed)                 | 0.76     | 0.93                 | 0.53                   |
|                                  | (0.12)   | (0.04)               | (0.29)                 |

TABLE 5. Evaluation of the proposed method in terms of intra-class accuracy of known classes by one of repeated experiments. KC denotes known class.

| Dataset      | KC 1 | KC 2 | KC 3 | KC 4 | KC 5 | KC 6 |
|--------------|------|------|------|------|------|------|
| Epilepsy     | 0.70 | 0.89 | -    | -    | -    | -    |
| MHEALTH      | 0.50 | 0.50 | 0.77 | 0.36 | 1.00 | 0.70 |
| RacketSports | 0.37 | 0.47 | -    | -    | -    | -    |
| SpokenArabicDigits | 0.95 | 0.88 | 0.84 | 0.92 | 0.93 | -    |
| UCI HAR      | 0.91 | 0.92 | 0.96 | -    | -    | -    |
| UWaveGestureLibrary | 0.30 | 0.50 | 0.53 | 0.50 | -    | -    |

TABLE 6. Comparison of the computational time between baseline and proposed methods for six datasets.

| Dataset      | Training time (second) | Inference time (second) |
|--------------|------------------------|-------------------------|
|              | Baseline | MEROS (Proposed) | Baseline | MEROS (Proposed) |
| Epilepsy     | 1.24     | (0.23)            | 0.34      | (0.03)            |
| MHEALTH      | 24.47    | (3.29)            | 4.1       | (1.99)            |
| RacketSports | 4.3      | (0.20)            | 0.29      | (0.04)            |
| SpokenArabicDigits | 57.93 | (0.50)            | 11.55     | (0.27)            |
| UCI HAR      | 116.99   | (7.39)            | 6.1       | (1.40)            |
| UWaveGestureLibrary | 1.22 | (0.20)            | 0.65      | (0.03)            |

FIGURE 6. Probability distribution of the predictions of unknown samples as unknown in terms of feature extractions.

worth noting that UWaveGestureLibrary has a very small number of samples per known class compared with other datasets. In these situations, the proportion of outliers to the entire samples can be large. We thought that the proposed method was less tolerant of this problem than the baseline model. Nevertheless, the experimental results demonstrated the usefulness of the proposed MEROS using multi-feature extraction and thresholds based on reconstruction errors.

To examine the respective importance of multi-feature extraction and reconstruction-based thresholds, we conducted further experiments. The human activity recognition dataset (UCI HAR), the dataset that led to the best performance of the proposed method, was used. Table 4 indicates performance depending on whether the components of the proposed method were applied. The results show that, when all components are used, the degree to which the accuracy of the unknown classes has improved is much greater than the degree to which the accuracy of the known classes has declined. For this reason, MEROS has the highest F1 score. Even when each component is used one by one, MEROS performs better than the baseline model. Multi-feature extraction enables it to extract features far more informative than those produced by the baseline model. In this way, unknown-detection performance is improved. Reconstruction-based thresholds have further significant effects. By using the thresholds, unknowns were detected more accurately than with the baseline model while maintaining the performance of known-class classification.

Figure 6 shows the probability distribution in which unknown samples are predicted as unknown. The probability distributions differ depending on the way features are extracted. As more varied features are extracted, the number of unknown samples with a high probability of being unknown increases. Consequently, we confirmed that the richer the information contained in the final features, the more informative the characteristics provided regarding unknowns. Thus, the number of unknown samples that were correctly predicted increases.

Experimental results show that the MEROS shows good unknown class accuracy while maintaining comparable known class accuracy leading to the highest F1 score. To further examine the classification performance for known classes of the proposed MEROS, we evaluated their intra-class accuracy. Table 5 shows that some variations in intra-class classification performance exist. Future research effort could be devoted to balancing intra-class accuracy of known classes while maintaining good unknown-detection performance.

Table 6 shows that training and inference times between the baseline and proposed methods. We found that the proposed method requires more times in training because...
it involves multi-feature extraction. Nevertheless, because the unknown-detection performance of the proposed method is high, the proposed algorithm is useful for open-set recognition of MTS data. The inference time is comparable between the baseline and proposed methods.

We also conducted further experiments on our proposed reconstruction-based thresholds. We set the reconstruction-based thresholds to the maximum value of the reconstruction error distribution of each class for unknown-detection criteria, as shown in Equation (8). To examine the change of F1 score of the proposed method according to changes in percentile value of reconstruction error distribution, experiments were conducted by changing the percentile value. As shown in Figure 7, there is no significant difference in the performance of our proposed method when the reconstruction-based thresholds are changed. Therefore, we found that our proposed reconstruction-based thresholds are not sensitive to changes in percentile values.

As a result of these experiments, we determined that providing varied features for detection of unknowns leads to improved performance. In addition, we found that applying reconstruction-based thresholds as criteria has the beneficial effect of distinguishing between knowns and unknowns.

V. CONCLUSION

In this study, we proposed an open-set recognition algorithm for MTS data. To facilitate more informative features that contain the unique characteristics of unknowns, the proposed MEROS is equipped with multi-feature extraction, specifically channel-wise 1D CNN and max pooling latent representations. In addition, reconstruction-based thresholds are used to achieve much stricter criteria for detection of unknowns. Therefore, the model can reject input samples whose reconstruction errors exceed the thresholds of the known classes. Our experiments demonstrated that, because the features extracted by the various layers are added to the final features, abundant information is available for utilization in detection of unknowns. Furthermore, the reconstruction-based criteria help boost performance in detecting unknowns while maintaining known-classification performance. We believe that using diverse features and more restrictive unknown-detection criteria is more effective than using the state-of-the-art structure of image data as it is.

Although the proposed method shows promising results, it has the limitation of high computational cost. Because of the additional channel-wise 1D CNN that extracts the features of each dimension separately, the computational complexity can increase, depending on the number of dimensions of input data. For a more efficient way of overcoming this problem, an attention mechanism can be considered. Such a mechanism allows models undergoing training to focus on the important segments of the input. An attention mechanism can learn useful channel-wise features by focusing on important channels. We believe that incorporating an attention mechanism can achieve good performance while reducing computational costs compared with building a 1D CNN for each channel.

ACKNOWLEDGMENT

The authors would like to thank the editor and reviewers for their useful comments and suggestions, which were greatly helpful in improving the quality of the article.

REFERENCES

[1] F. Karim, S. Majumdar, H. Darabi, and S. Harford, “Multivariate LSTM-FCNs for time series classification,” Neural Netw., vol. 116, pp. 237–245, Aug. 2019.
[2] X. Cheng, P. Han, G. Li, S. Chen, and H. Zhang, “A novel channel and temporal-wise attention in convolutional networks for multivariate time series classification,” IEEE Access, vol. 8, pp. 212247–212257, 2020.
[3] G. Wu, H. Zhang, Y. He, X. Bao, L. Li, and X. Hu, “Learning Kullback–Leibler divergence-based Gaussian model for multivariate time series classification,” IEEE Access, vol. 7, pp. 139580–139591, 2019.
[4] A. Gupta, H. P. Gupta, B. Biswas, and T. Dutta, “An early classification approach for multivariate time series of on-vehicle sensors in transportation,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 12, pp. 5316–5327, Dec. 2020.
[5] R. Yuan, Y. Lv, H. Li, and G. Song, “Robust fault diagnosis of rolling bearings using multivariate intrinsic multiscale entropy analysis and neural network under varying operating conditions,” IEEE Access, vol. 7, pp. 130804–130819, 2019.
[6] A. Gumaei, M. M. Hasan, A. Aclaawi, and H. Alsalman, “A hybrid deep learning model for human activity recognition using multimodal body sensing data,” IEEE Access, vol. 7, pp. 99152–99160, 2019.
[7] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, “Deep learning for time series classification: A review,” Data Mining Knowl. Discovery, vol. 33, no. 4, pp. 917–963, Mar. 2019.
[8] A. Bendale and T. E. Boult, “Towards open set deep networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1563–1572.
[9] R. Yoshiihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, “Classification-reconstruction learning for open-set recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4016–4025.
[10] C. Geng, S.-J. Huang, and S. Chen, “Recent advances in open set recognition: A survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3614–3631, Oct. 2021.
[11] M. Tonmoy, S. Mahmud, A. M. Rahman, M. A. Amin, and A. A. Ali, “Hierarchical self attention based autoencoder for open-set human activity recognition,” in Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining, Cham, Switzerland: Springer, 2021, pp. 351–363.
[12] M. Wistuba, J. Grabocka, and L. Schmidt-Thieme, “Ultra-fast shapelets for time series classification,” 2015, arXiv:1503.05018.
[13] I. Karlsson, P. Papapetrou, and H. Bostrom, “Generalized random shapelet forests,” Data Mining Knowl. Discovery, vol. 30, no. 5, pp. 1053–1085, Sep. 2016.
[14] Z. Zhang, G. Zhang, Z. Zhang, G. Chen, Y. Zeng, B. Wang, and E. R. Hancock, “Structural network inference from time-series data using a generative model and transfer entropy,” Pattern Recognit. Lett., vol. 125, pp. 357–363, Jul. 2019.
15] F. De Vita, D. Brunace, and S. K. Das, “On the use of a full stack hardware/software infrastructure for sensor data fusion and fault prediction in industry 4.0,” Pattern Recognit. Lett., vol. 138, pp. 30–37, Oct. 2020.
16] R. Khasha, M. M. Sepehri, and N. Taherkhani, “Detecting asthma control level using feature-based time series classification,” Appl. Soft Comput., vol. 111, Nov. 2021, Art. no. 107694.
17] R. Chandra, S. Goyal, and R. Gupta, “Evaluation of deep learning models for multi-step ahead time series prediction,” IEEE Access, vol. 9, pp. 83105–83123, 2021.
18] C.-Y. Hsu and W.-C. Liu, “Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing,” J. Intell. Manuf., vol. 32, no. 3, pp. 823–836, Mar. 2021.
19] Z. Huang, J. Zhu, J. Lei, X. Li, and F. Tian, “Tool wear predicting based on multi-domain feature fusion by deep convolutional neural network in milling operations,” J. Intell. Manuf., vol. 31, no. 4, pp. 953–966, 2019.
20] D. K. Dennis, D. Acar, V. Mandikal, V. Sadasivan, H. Simhadri, V. Saligram, and P. Jain, “Shallow rnns: A method for accurate time-series classification on tiny devices,” in Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS), Vancouver, BC, Canada, 2019, pp. 8–14.
21] C. Gallicchio and A. Micheli, “Deep echo state network (DeepESN): A brief survey,” 2017, arXiv:1712.04323.
22] F. Karim, S. Majumdar, H. Darabi, and S. Chen, “LSTM fully convolutional networks for time series classification,” IEEE Access, vol. 6, pp. 1662–1669, 2017.
23] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu, “TapNet: Multivariate time series classification with attentional prototypical network,” in Proc. AAAI Conf. Artif. Intell., vol. 34, no. 4, 2020, pp. 6845–6852.
24] M. Hassen and P. K. Chan, “Learning a neural-network-based representation for open set recognition,” in Proc. SIAM Int. Conf. Data Mining, Philadelphia, PA, USA: SIAM, 2020, pp. 154–162.
25] L. Shu, H. Xu, and B. Liu, “DOC: Deep open classification of text documents,” 2017, arXiv:1709.08716.
26] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, “Learning placeholders for open-set recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 4401–4410.
27] X. Sun, Z. Yang, C. Zhang, K.-V. Ling, and G. Peng, “Conditional Gaussian distribution learning for open set recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 13480–13489.
28] B. J. Meyer and T. Drummond, “The importance of metric learning for robotic vision: Open set recognition and active learning,” in Proc. Int. Conf. Robot. Autom. (ICRA), May 2019, pp. 2924–2931.
29] L. Chambers and M. M. Gaber, “DeepStreamOS: Fast open-set classification for convolutional neural networks,” Pattern Recognit. Lett., vol. 154, pp. 75–82, Feb. 2022.
30] T. Akar, T. Werner, V. K. Yalavarthi, and L. Schmidt-Thieme, “Open set recognition for time series classification,” in Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining. Cham, Switzerland: Springer, 2022, pp. 354–366.
31] A. Bagnall, M. Flynn, J. Large, J. Lines, and M. Middlehurst, “On the usage and performance of the hierarchical collective of transformation-based ensembles version 1.0 (hive-cote v1.0),” in Proc. Int. Workshop Adv. Anal. Learn. Temporal Data. Cham, Switzerland: Springer, 2020, pp. 3–18.
32] M. Azizjon, A. Jumabek, and W. Kim, “1D CNN based network intrusion detection with normalization on imbalanced data,” in Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC), Feb. 2020, pp. 218–224.
33] T. Amarbayasgalan, V. H. Pham, N. Theera-Umpon, and K. H. Ryu, “Unsupervised anomaly detection approach for time-series in multi-domains using deep reconstruction error,” Symmetry, vol. 12, no. 8, p. 1251, Jul. 2020.
34] I. Nunes, M. B. Pereira, H. Oliveira, J. A. dos Santos, and M. Poggi, “Conditional reconstruction for open-set semantic segmentation,” 2022, arXiv:2203.01368.
35] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, “The UEA multivariate time series classification archive, 2018,” 2018, arXiv:1811.00075.
36] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Müller, and F. Petitjean, “Inceptiontime: Finding alexnet for time series classification,” Data Mining Knowl. Discovery, vol. 34, pp. 1936–1962, Sep. 2019.
37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 1–11.

HYEREUNG OH received the B.S. degree in 2019. She is currently pursuing the M.S. degree with the School of Industrial and Management Engineering, Korea University, Seoul, South Korea. Her research interests include multivariate time series data analysis, open-set recognition, and their industrial applications.

SEOUNG BUM KIM received the M.S. degree in industrial and systems engineering, the M.S. degree in statistics, and the Ph.D. degree in industrial and systems engineering from the Georgia Institute of Technology, in 2001, 2004 and 2005, respectively. From 2005 to 2009, he was an Assistant Professor at the Department of Industrial and Manufacturing Systems Engineering, University of Texas at Arlington. He is a Professor with the School of Industrial and Management Engineering, Korea University. His research utilizes machine learning algorithms to create new methodologies for various problems appearing in engineering and science. He has expertise in semi-supervised learning, self-supervised learning, and uncertainty quantification in deep neural networks. He has published more than 170 internationally recognized journals and refereed conference proceedings.

* * *