Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 22, Iss. 1, pp. 32-44, 2024

Blending Ensemble applied to Open-Set Recognition for Time Series Classification

Leonardo de Marqui Marques, André Eugenio Lazzaretti, Heitor Silvério Lopes
Universidade Tecnológica Federal do Paraná (UTFPR) / CPGEI, Curitiba, Brazil
leonardommarques@gmail.com, {lazzaretti,hslopes}@utfpr.edu.br

Resumo – Séries Temporais, incluindo preços de ações, temperaturas e marcadores de saúde, são monitorados o tempo todo e em diversos lugares. Eventos dependentes do tempo são frequentemente estudados no âmbito da previsão, onde os valores passados das séries são usados para prever os futuros. Por outro lado, a Classificação de Séries Temporais visa criar modelos que rotulem instâncias de séries temporais. As técnicas de Reconhecimento em Conjunto Aberto, projetadas para classificar amostras conhecidas e detectar instâncias desconhecidas simultaneamente, têm sido menos aplicadas no contexto de Classificação de Séries Temporais em comparação com outros campos, como classificação de imagens. Os métodos existentes têm limitações, como não usar o conhecido-desconhecido no estágio de treinamento, aprendizagem de transferência limitada entre modelos de redes neurais e experimentos com conjuntos de dados de referência com cobertura estreita. Este estudo apresenta uma nova abordagem Reconhecimento em Conjunto Aberto para Classificação de Séries Temporais, abordando as questões mencionadas e usando um conjunto combinado de redes neurais com uma camada OpenMax. Os resultados atestam o desempenho e potencial superioridade do modelo como alternativa aos métodos existentes em tarefas de reconhecimento de conjunto aberto.

Palavras-chave – Classificação de séries temporais, reconhecimento em conjunto aberto, aprendizado de máquina.

Abstract – Time Series (TS), including stock prices, temperatures, and health markers are monitored all the time and everywhere. Time-dependent events, and TS are frequently studied in the scope of forecasting, where past values of the TS are used to preview future ones. On the other hand, Time Series Classification (TSC) aims at creating models that label TS instances. Open Set Recognition (OSR) techniques, designed to classify known samples and detect unknowns simultaneously, have been less applied in TSC compared to other fields like image classification. Existing methods have limitations, such as not using known-unknown in the training stage, limited transfer learning across Neural Network models, and experiments with benchmark datasets with narrow coverage. This study introduces a novel OSR approach for TSC by addressing the mentioned issues and using a blending ensemble of Neural Networks with an OpenMax layer. The results vouch for the model's performance and potential superiority as an alternative to existing methods in open-set recognition tasks.

Keywords – Time Series Classification, Open Set Recognition, Machine Learning.

1 Introduction

When an event has time-dependent values, we can study it as a Time Series (TS). Stock prices, temperatures, and health markers are some of the everyday life examples of them. TS are frequently studied in the scope of forecasting, where past values of the TS are used to preview future ones. However, Time Series Classification (TSC) involves creating a model that labels a TS instance [1]. Such a classification may be useful for comparing the behavior of different TS, making it an important problem in machine learning, therefore justifying efforts for its development and improvement.

On the one hand, in general data classification, the classes of the data instances are previously known. On the other hand, Open-Set Recognition methods are models that can be created in such a way as to simultaneously classify the known samples and detect the samples of unknown classes. However, to date, they have limited application in TSC, when compared to other areas, such as text or image classification [2]. Nevertheless, this gap has been addressed by recent research introducing innovative methodologies. For instance, Open Set InceptionTime [3] uses barycenters to compute Dynamic Time Warping (DTW) distances and cross-correlations, identifying unknown samples based on threshold criteria. The MEROS method [4] focuses on preserving unique features in unknown classes by employing multi-feature extraction and integrating one-dimensional convolutional neural networks (1D-CNN) for a richer feature set. HiNoVa [5] applies a Convolutional Neural Network with Long Short-Term Memory (CNN+LSTM) model, generating features by aggregating hidden state values and using Kendall's correlation for classification.

The above-mentioned methods come with some theoretical and practical limitations. A notable gap is not using the "known unknowns" during training, which could enhance model robustness and generalization. Additionally, the application of the transfer-learning approach across various neural network models still needs to be improved [6]. Furthermore, experiments involving existing benchmark datasets have a narrow coverage.

In this research, we introduce a novel approach to applying Opens Set Recognition (OSR) to TSC by means of blending neural networks integrated with an OpenMax layer and incorporating known unknown instances during the training phase. Additionally, we conduct comprehensive experiments across various configurations, with several TS datasets from diverse domains. This

extensive experimental setup was designed to rigorously evaluate the effectiveness of the proposed model in a large range of application scenarios.

This paper is structured as follows. Sections 2 and 3 present the fundamental theory and related works, respectively. Section 4 presents a detailed description of the methodology, experiment setup, and the metrics used. Next, Sections 5 and 6, respectively, show the results and conclusions.

2 Theoretical Aspects

2.1 Time Series Classification

A real-world TS may have one or more values for each time step. Suppose a TS with just one measurement (channel) along the time. In this case, it is defined as a univariate TS. Conversely, it is considered a multivariate TS when it has more than one measurement at a time. In TSC, TS the measurements are taken as the inputs to a Machine Learning (ML) model, which outputs a probability estimate for each class. TSC is different from conventional classification because the attributes are, by nature, time-dependent [7]. Time series classification is crucial across fields, enabling pattern detection, trend analysis, and anomaly detection for prescriptive and predictive modeling. It supports decision-making by forecasting future events, categorizing past ones, and optimizing resources.

2.2 Open Set Recognition

In machine learning, OSR poses a unique challenge: models must classify known classes and detect unknown classes during testing. This reflects a more realistic scenario than the traditional closed-set classification. In closed-set classification, models are trained on a dataset with a predefined set of known classes $C = [c_1, c_2, ..., c_M]$. However, OSR extends this set to include unknown classes: $C' = [c_1, c_2, ..., c_M, c_{M+1}, \cdots, c_{M+\Omega}]$. Such an extension means that a given input data can belong either to a known class $c_i \in C$ or can be categorized as unknown. Figure 1(a) illustrates closed-set classification with decision boundaries defined by a Nearest Class Mean (NCM) [8] classifier for three known classes. Unknown classes, shown as stars, fall outside these boundaries. Figure 1(b) shows the original dataset distribution in the open space, highlighting the limitations of a closed three-class model. In contrast, OSR, shown in Figure 1(c), can distinguish between known and unknown samples, limiting decision-making to the areas covered by the training data thereby addressing misclassification of unknown observations in the open space.

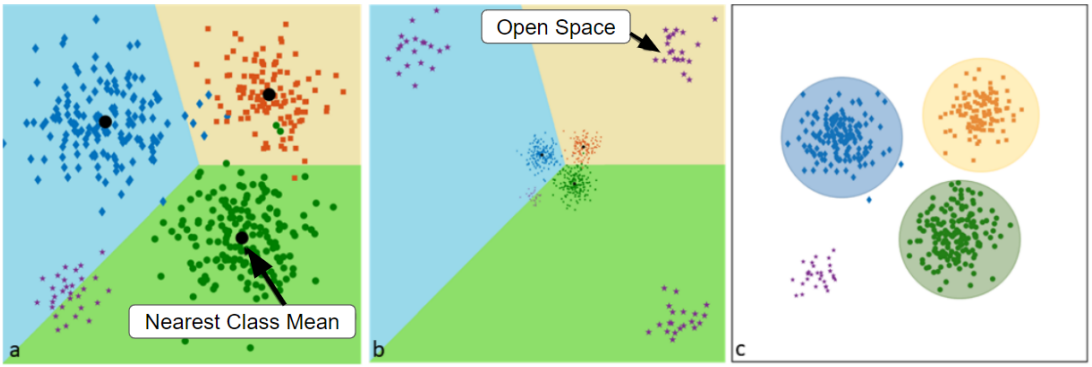

Figure 1: An overview of the issue with OSR.

The Extreme Value Machine (EVM) [9] applies the Extreme Value Theory (EVT) to calibrate the class boundaries of a Support Vector Machine [10] using a Probability Density Function (PDF). Data points falling outside these extreme boundaries are classified as “unknown”.

Another approach that uses the concept of EVT is OpenMax [11] which marked the inception of deep open-set classification without background samples. However, few reports of deep open-set classifiers appeared in the literature. While OpenMax primarily focuses on recognizing adversarial inputs, it also supports the rejection of fooling and unknown images. Rosza et al. [12] compared DNNs using the conventional Softmax layer and the OpenMax so as to evaluate their robustness against adversarial examples. OpenMax exhibited more resilience to adversarial examples than SoftMax and performed better than networks using the SoftMax threshold.

Actually, OpenMax uses the EVT model, constructed from positive training samples, to establish a class-specific Cumulative Activation Profile (CAP) model, allowing for the rejection of unknown inputs through appropriate thresholding. This process involves calculating the activation vector for each training instance, denoted as $V(x) = v_1(x), ..., v_N(x)$ for each class $C_j, j \in \{1, ..., N\}$. Additionally, a Mean Activation Vector (MAV) is computed for each class, considering only the correctly classified training examples using the NCM concept mentioned before. Subsequently, for each class C_l the η largest distances between the MAV_j and the correctly classified training instances are used to fit a Weibull distribution, which results in the parameter vector ρ_j.

Let $\rho_j = (\tau_j, \lambda_j, \kappa_j)$ denote the location, shape, and scale parameters of the Weibull distribution for the estimated EVT Meta-Recognition model for class j. Weights for the α largest activation classes are computed to scale the Weibull probability as the following:

$$\omega_{s(i)}(x) = 1 - \frac{\alpha - i}{\alpha} F_W(x - \text{MAV}_i)|\rho_i) \tag{1}$$

where $s(i) = argsort(vj)_i$ and F_W is the Weibull CDF.

The recalibrated OpenMax activations are computed in the testing phase, incorporating the probabilities derived from the Weibull distribution (Eq. 2). The activation for the unknown class, represented with the zero index, is estimated using Eq. (3). Subsequently, the Softmax layer is applied to calculate and adjust the class probabilities based on the values of the new activation vectors, using Eq. (4):

$$\hat{V}(x) = V(x)\omega(x), \tag{2}$$

$$\hat{v}_0(x) = \sum_i^n v_i(x)(1 - \omega_i(x)), \tag{3}$$

$$P(y = j|X) = \frac{e^{\hat{v}_j(x)}}{\sum_{i=0}^{N} e^{\hat{v}_i(x)}}. \tag{4}$$

OpenMax does not improve feature representation for better detecting unknowns. The class instances are not aligned around the MAVs, and feature engineering attempts did not improved results. Moreover, a recurring issue is to find a good trade-off for models: usually they are either good at detecting unknowns, or classifying knowns, but rarely both. Furthermore, finding a deterministic rule to combine a robust classifier with an effective detector is challenging.

A reliable OSR system has good classification and detection performance even when facing numerous unknown classes. For measuring that, an important metric for evaluating OSR is openness [13], which is defined as follows. Let C_{TA}, C_{TR}, and C_{Unk} represent, respectively, the set of classes to be recognized, the set of classes used in training, and the set of unknown classes used during testing. Then, the openness O of the corresponding recognition task is defined as:

$$O = 1 - \sqrt{\frac{2 \times |C_{TR}|}{2 * |C_{TA}| + |C_{Unk}|}} \tag{5}$$

where $|\cdot|$ denotes the number of classes in the corresponding set. The larger the openness, the more open the problems. When the problem is completely closed, the openness becomes 0.

2.3 Blending

"Stacking" [14], also known as "blending", is a method that uses the predictions from a group of models as input to a secondary model, which is trained to combine these predictions to a final output. This approach leverages the strengths of individual models and mitigates their weaknesses. Blending has been widely successful among machine learning professionals, often enhancing prediction quality to levels hardly achievable by single models.

3 Related Works

3.1 Time Series Classification

Nearest-neighbor methods in supervised learning predict a new sample's target value based on similar samples. These algorithms often use the Euclidean distance to measure similarity. However, Euclidean distance faces challenges when applied to time series with different lengths, since resampling time series might decrease data representation of the features. Besides, the Euclidean distance compares the values of both time series independently, disregarding that the values are correlated. Sakoe and Chiba [15] introduced the Dynamic Time Wrapping (DTW) as an alternative measure to address these limitations of the Euclidean distance. In essence, DTW allows the stretch or compression of the time axes of the Time Series to find the optimal alignment, enabling a more accurate comparison. Despite the advantages over the Euclidean distance, DTW has relevant limitations. First, its high algorithmic complexity, which makes it computationally expensive for long time series. Also, it has substantial time warps, which might change the time series more than desirable. Furthermore, DTW is non-differentiable, which is a challenge for integration and research with machine learning algorithms that rely on gradient-based optimization [16]. Therefore, DTW is frequently employed alongside algorithms that rely on similarity metrics, such as SVM and K-Nearest Neighbors (KNN), as second option to be used instead of the Euclidean distance.

3.2 Open Set Recognition for Time Series Classification

Open-set recognition techniques have a well-established presence, although their application to time series classification has not been as extensive as observed in other domains, such as image classification. However, the relative scarcity of contributions in this area is counterbalanced by the quality of the research, where researchers came up with inventive and innovative methodologies to address this challenge.

Akar et al. [17] proposed the Open Set InceptionTime, which integrates barycenters in the model. When presented with novel samples, the model computes their DTW distance and cross-correlation with class-specific barycenters. If the DTW distance exceeds a specified threshold or the cross-correlation falls below its threshold, the sample is rejected and identified as unknown. This framework is the pioneering application of OSR in TSC and a reference for future endeavors. It was also used as the baseline for other works in the same domain and, therefore, it will be employed as a benchmark in the current study.

The MEROS (Multi-Feature Extraction and Reconstruction Learning for Open-Set Recognition) [4] method was aimed to preserve the unique features associated with unknown classes. This was done through multi-feature extraction applied to TS data. It also uses OpenMax for unknown detection. However, in contrast to other methods that predominantly depend only on features from the final activation layer, MEROS incorporates channel-wise one-dimensional convolutional neural networks (1D-CNN) to leverage a more diverse set of features, additionally integrating the main network, global 1D-CNN. Authors did experiments with several TS datasets, and demonstrate that MEROS was capable of achieving improved detection of unknown classes, while maintaining strong predictive performance.

HiNoVa [5] is another method that leverages activations of multiple hidden layers of a DNN. More precisely, it is an architecture that includes a Convolutional Neural Network, followed by a Long Short-Term Memory model (CNN+LSTM). The HiNoVa approach uses the hidden state values within a trained CNN+LSTM to generate features for each sample in the training set. This involves aggregating the values for each hidden layer node in the LSTM from all correctly classified samples during training for each known sample. Subsequently, a histogram with B bins is constructed for each known sample, capturing the distribution of hidden state values for each hidden layer in the LSTM. This matrix histogram serves as extracted features. During testing, the algorithm computes the Kendall's correlation τ between the new and all training samples. The highest correlation value is then compared to a predefined threshold, and if it is found to be below this threshold, the sample is detected as unknown.

4 Methods

In this Section we show the methods used in our experiments to investigate open-set recognition using a novel approach: the blendings of neural networks trained on the known and known-unknown TS data.

4.1 Datasets

In this work we used a diverse and comprehensive set of TS datasets provided by the UEA archive[1].This collection of datasets is usually the standard for research on time series as it represents several real-world scenarios. The time series metadata used in our experiments is shown in Table 1, and their main features are: length ranging from 8 to 2844, number of observations from 27 to 10992, dimensions from 1 to 963, and number of classes from 2 to 60. Some datasets are more complex than others. For instance, PEMS-SF, with 963 dimensions, brings significant challenges in handling high-dimensional data.Datasets like ShapesAll and PhonemeSpectra, which have 60 and 39 classes respectively, increase the risk of misclassification due to their large number of classes. Short time series, such as PenDigits with a length of 8, test the model's prediction power with limited data points. Datasets with a limited number of samples, such as StandWalkJump and AtrialFibrillation with 27 and 30 observations respectively, pose difficulties in maintaining robust performance. These diverse and demanding characteristics ensure that the models are tested under challenging, non-trivial conditions and validate their effectiveness and robustness in real-world scenarios.

4.2 Deep Neural Networks

Our extensive experiments leverage the state-of-the-art DNNs by applying transfer learning to nine architectures, as follows: FCN [18], InceptionTime [3], LSTM [19], LSTM-FCN [20], mWDN [21], OmniScaleCNN [22], ResCNN [23], ResNet [24], XceptionTime [25]. This set of DNNs range from widely adopted architectures and others not frequently used, while also incorporating models that exhibit strong performance in the domain of TSC, as well as those that are not specifically developed for such tasks.

After varying hyperparameters of the DNNs, we had up to 25 models for each time series. In our blending approach, the DNNs activations as well as the OpenMax probability estimates are concatenated to create the "blending features" that feed an ensemble model based on the XGBoost [26], which is responsible for the final classification. XGBoost[2] was chosen due to its robustness, scalability, and ease of use.

4.3 Experimental Setup

The experiments were done for each time series, following the next steps.

[1] https://www.timeseriesclassification.com/
[2] https://xgboost.readthedocs.io/en/stable/

Table 1: Time Series Metadata.

Time Series	Train Size	Test Size	Length	Dimensions	Classes
ArticularyWordRecognition	275	300	144	9	25
AtrialFibrillation	15	15	640	2	3
BasicMotions	40	40	100	6	4
CharacterTrajectories	1422	1436	182	3	20
Coffee	28	28	286	1	2
Cricket	108	72	1197	6	12
DuckDuckGeese	50	50	270	15	5
Epilepsy	137	138	206	3	4
ERing	30	270	65	4	6
EthanolConcentration	261	263	1751	3	4
FingerMovements	316	100	50	28	2
HandMovementDirection	160	74	400	10	4
Handwriting	150	850	152	3	26
Heartbeat	204	205	405	61	2
JapaneseVowels	270	370	29	12	9
Libras	180	180	45	2	15
LSST	2459	2466	36	6	14
NATOPS	180	180	51	24	6
PEMS-SF	267	173	144	963	7
PenDigits	7494	3498	8	2	10
PhonemeSpectra	3315	3353	217	11	39
RacketSports	151	152	30	6	4
RefrigerationDevices	375	375	720	1	3
Rock	20	50	2844	1	4
ScreenType	375	375	720	1	3
SelfRegulationSCP1	268	293	896	6	2
SelfRegulationSCP2	200	180	1152	7	2
ShapesAll	600	600	512	1	60
SmallKitchenAppliances	375	375	720	1	3
SpokenArabicDigits	6599	2199	93	13	10
StandWalkJump	12	15	2500	4	3
SwedishLeaf	500	625	128	1	15
TwoPatterns	1000	4000	128	1	4
UWaveGestureLibrary	120	320	315	3	8

4.3.1 Training the DNNs

The datasets used in this study were found in the above-mentioned repository already pre-segmented into training and testing subsets with diverse proportions for each split. We further partitioned the training set into training and validation sets (70% and 30%, respectively) to facilitate the training. After this partitioning process, the OpenMax layer was trained on the neural network architecture. Thus, for each model, we have the activations of the fitted neural network and the fitted OpenMax layer.

4.3.2 Training Blendings

To evaluate the performance of our approach, multiple neural networks and their respective OpenMax layers were used for all the time series. Therefore, the set of diverse models constituted the basis for our blending approach. The machine learning engineering pipeline applied, represented by Figure 2, is the following: For each model, the neural network activations (one for each class) and the OpenMax layer predictions (one for each class plus one extra for the unknown class) were extracted as features. These features were concatenated across all models, making a novel feature matrix for the known time series. These features were also extracted for the other time series that served as representatives of the known-unknowns and unknowns. Notably, both known-unknown and unknown time series instances inherently deviated in sequence length and number of dimensions of the known time series. We implemented a resampling technique to address this challenge, ensure compatibility, and enable consistent model evaluation across known and unknown classes.

4.3.3 Data Splits

Once we have the features we train our bending model, the XGboost. For this purpose the data was split into train, test, and open-set splits, as illustrated in Figure 4.3.3). The open-set split of our dataset has data exclusively from the unknown category. In contrast, the training and testing splits have a combined set of known and unknown (a.k.a known-unknowns) time series. It is important to underline that the three distinct data subsets are intentionally built so that they do not share instances of unknown time series. This partition strategy was adopted to reduce the risk of overfitting to the patterns in the known-unknown class.

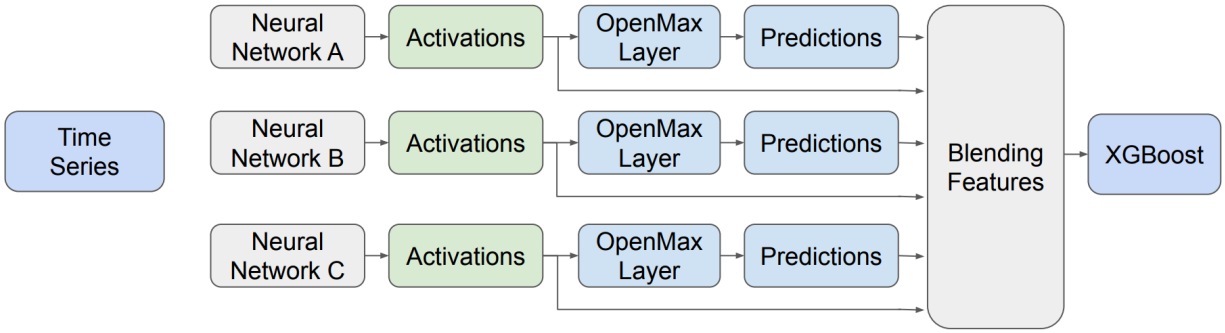

Figure 2: Overview of the Blending Approach

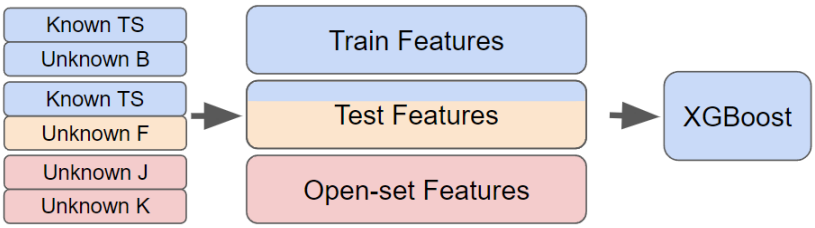

Figure 3: Blending Split Overview.

4.3.4 Hyperparamenters

While exploring several neural network architectures in our study, we systematically varied the hyperparameters to increase model diversity. The chosen architectures comprise both conventional and advanced models. Below is an overview of these variations:

- **FCN, ResNet, ResCNN, InceptionTime and OmniScaleCNN**: default settings;
- **LSTM_FCN**: Shuffle (True and False);
- **LSTM**: Number of layers (1, 2, or 3), Bidirectionality (False and True);
- **mWDN**: levels (2, 4, and 6);
- **XceptionTime**: Default setting, Number of filters (8 and 24), Adaptive sizes (10, 50, 90, 150, 200 and 300).

The neural networks were implemented using the TSAI [27] python library.

4.3.5 Quality Metrics

In this study we focus on the two quality metrics mostly applied in similar researches: Accuracy and F1-score. Accuracy is a measure of a classification model's overall correctness, calculated as the ratio of correct predictions, given by sum of the true positives (TP) and the true negatives (TN), divided by the number of total predictions. However, Accuracy is not adequate for unballanced datasets. In this case, the F1-score (F1) is more suitable, since it balances precision (proportion of correct positive predictions of total predicted positives) and recall (proportion of correct positive predictions of all actual positives). F1 is calculated as:

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{6}$$

in which FP is the false positives and FN the false negatives.

5 Results

In this Section, we delve into the results of our study on open-set recognition.

5.1 Overview

Tables 2 and 3 present a detailed overview of the quality metrics obtained in our experiments. Results show that, for a significant portion (76.2%) of the time series tested, the metrics exceeded the 0.9 mark, with the F1 Score averaging 0.892. These results suggest the effectiveness of the proposed method in delivering reliable classification results in open-set scenarios.

A deeper analysis reveals that, for certain datasets, including Coffee, HandMovementDirection, and SwedishLeaf, there is a consistent exhibition of high performance across all three data splits (training, test, and open-set). This consistency in achieving robust metrics validates the model's proficiency for accurately identifying known classes and detecting unknown instances.

Datasets such as ERing, Heartbeat, and MotorImagery, a relevant variance in performance was noticed. Here, the test or open-set metrics are observed to be lower than those achieved during training, suggesting an overfitting tendency of the model. This performance disparity, particularly in the context of distinguishing between known and unknown classes, brings concerns regarding the model's generalization capabilities. However, future work focused towards Machine Learning Engineering, covering aspects like feature engineering, model tuning, and regularization techniques, will significantly enhance the model's performance metrics, mitigating overfitting issues and improving the model's ability to generalize across datasets.

Table 2: For each time series and data split, the table shows the average and standard deviation of the F1-score, presented separately by "±".

Time Series	train	test	openset
ArticularyWordRecognition	1.00 ± 0.01	0.89 ± 0.32	0.78 ± 0.44
AtrialFibrillation	0.97 ± 0.11	0.79 ± 0.34	0.83 ± 0.33
BasicMotions	0.98 ± 0.07	0.86 ± 0.22	0.87 ± 0.26
CharacterTrajectories	1.00 ± 0.01	0.93 ± 0.24	1.00 ± 0.00
Coffee	0.97 ± 0.11	0.96 ± 0.12	1.00 ± 0.00
Cricket	0.99 ± 0.03	0.94 ± 0.17	1.00 ± 0.00
DuckDuckGeese	0.97 ± 0.09	0.91 ± 0.21	0.81 ± 0.43
Epilepsy	0.98 ± 0.07	0.85 ± 0.23	0.66 ± 0.29
ERing	0.99 ± 0.03	0.45 ± 0.39	1.00 ± 0.00
EthanolConcentration	0.97 ± 0.10	0.97 ± 0.10	1.00 ± 0.00
FingerMovements	1.00 ± 0.01	0.59 ± 0.54	1.00 ± 0.00
HandMovementDirection	0.99 ± 0.02	0.99 ± 0.02	1.00 ± 0.00
Handwriting	0.99 ± 0.04	0.43 ± 0.42	0.80 ± 0.39
Heartbeat	0.88 ± 0.21	0.38 ± 0.34	1.00 ± 0.00
Libras	0.99 ± 0.02	0.99 ± 0.02	1.00 ± 0.00
MotorImagery	0.93 ± 0.16	0.72 ± 0.26	0.33 ± 0.00
NATOPS	1.00 ± 0.00	0.94 ± 0.17	1.00 ± 0.00
PEMS-SF	0.98 ± 0.05	0.88 ± 0.30	1.00 ± 0.00
PenDigits	0.99 ± 0.03	0.90 ± 0.26	1.00 ± 0.00
PhonemeSpectra	0.96 ± 0.09	0.85 ± 0.27	1.00 ± 0.00
RacketSports	0.98 ± 0.06	0.77 ± 0.31	1.00 ± 0.00
RefrigerationDevices	0.97 ± 0.09	0.84 ± 0.26	0.60 ± 0.46
Rock	0.98 ± 0.08	0.73 ± 0.40	0.88 ± 0.28
ScreenType	0.97 ± 0.10	0.83 ± 0.29	0.62 ± 0.25
SelfRegulationSCP1	0.98 ± 0.09	0.80 ± 0.35	0.88 ± 0.29
SelfRegulationSCP2	0.97 ± 0.10	0.84 ± 0.28	0.76 ± 0.38
ShapesAll	0.99 ± 0.03	0.79 ± 0.40	1.00 ± 0.00
SmallKitchenAppliances	0.97 ± 0.09	0.78 ± 0.37	0.74 ± 0.30
SpokenArabicDigits	0.99 ± 0.03	0.82 ± 0.30	0.87 ± 0.25
StandWalkJump	0.97 ± 0.10	0.92 ± 0.23	1.00 ± 0.00
SwedishLeaf	0.99 ± 0.03	0.94 ± 0.17	1.00 ± 0.00
TwoPatterns	0.98 ± 0.07	0.74 ± 0.28	0.69 ± 0.38
UWaveGestureLibrary	0.99 ± 0.04	0.90 ± 0.24	1.00 ± 0.00

5.2 Benchmark

In this Section a comparative analysis is accomplished to evaluate the effectiveness of our novel open-set recognition method. By means of a detailed comparison with an established benchmark proposed by [17], our objective is to highlight the unique strengths and significant contributions of our approach in OSR. Notice that a direct comparison between our approach and the cited benchmark is only partially possible, and this is due to certain inherent dissimilarities in the experimental design. A notable distinction lies in the composition of the test split. First, the benchmark study uses only two unknown time series, and our approach employs multiple unknown instances. Second, we introduce the additional open-set split. Notwithstanding, both approaches share a common methodology for computing the overall quality metrics. Specifically, both methods calculate these metrics by computing the mean of the individual metrics assigned to each time series across the respective splits.

Tables 4 and 5 compare our proposed method and the benchmark. We notice the second dissimilarity between the methods because they do not share the same time series. This is a minor issue, as we have enough intersections to compare. The average

Table 3: For each time series and data split, the table shows the average and standard deviation of the Accuracy, presented separately by "±".

Time Series	train	test	openset
ArticularyWordRecognition	1.00 ± 0.00	0.92 ± 0.23	0.93 ± 0.13
AtrialFibrillation	0.99 ± 0.04	0.91 ± 0.22	1.00 ± 0.00
BasicMotions	1.00 ± 0.01	0.99 ± 0.02	0.98 ± 0.03
CharacterTrajectories	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Coffee	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.00
Cricket	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
DuckDuckGeese	0.99 ± 0.05	0.98 ± 0.05	0.83 ± 0.39
Epilepsy	1.00 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
ERing	1.00 ± 0.01	0.91 ± 0.10	1.00 ± 0.00
EthanolConcentration	0.98 ± 0.05	0.98 ± 0.05	1.00 ± 0.00
FingerMovements	1.00 ± 0.01	0.59 ± 0.54	1.00 ± 0.00
HandMovementDirection	0.99 ± 0.02	0.99 ± 0.02	1.00 ± 0.00
Handwriting	0.99 ± 0.03	0.82 ± 0.22	0.93 ± 0.13
Heartbeat	0.99 ± 0.02	0.66 ± 0.57	1.00 ± 0.00
Libras	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.00
MotorImagery	0.99 ± 0.03	0.96 ± 0.05	0.97 ± 0.00
NATOPS	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
PEMS-SF	1.00 ± 0.01	0.92 ± 0.22	1.00 ± 0.00
PenDigits	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
PhonemeSpectra	0.96 ± 0.09	0.95 ± 0.09	1.00 ± 0.00
RacketSports	1.00 ± 0.01	0.92 ± 0.22	1.00 ± 0.00
RefrigerationDevices	1.00 ± 0.01	0.99 ± 0.01	0.84 ± 0.20
Rock	0.99 ± 0.03	0.87 ± 0.27	0.97 ± 0.06
ScreenType	0.99 ± 0.02	0.97 ± 0.07	0.98 ± 0.02
SelfRegulationSCP1	1.00 ± 0.00	0.89 ± 0.29	0.96 ± 0.10
SelfRegulationSCP2	0.99 ± 0.02	0.97 ± 0.05	0.90 ± 0.21
ShapesAll	0.99 ± 0.03	0.97 ± 0.05	1.00 ± 0.00
SmallKitchenAppliances	1.00 ± 0.01	0.88 ± 0.30	0.95 ± 0.09
SpokenArabicDigits	1.00 ± 0.00	0.99 ± 0.03	1.00 ± 0.00
StandWalkJump	0.98 ± 0.08	0.94 ± 0.16	1.00 ± 0.00
SwedishLeaf	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.00
TwoPatterns	1.00 ± 0.00	1.00 ± 0.01	0.98 ± 0.03
UWaveGestureLibrary	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.00

F1 score of the benchmark is 0.66, whereas our method achieved 0.82. The average accuracy for Benchmark and our model are 0.90 and 0.94, respectively.

To compare the metrics more effectively, we adopted a methodology that calculates the discrepancy between the metrics obtained from our method and that from the benchmark. Specifically, we compute the increase in metric value by implementing our approach and then calculate the differences: $\text{Metric}_{OurTest}$ - $\text{Metric}_{BenchmarkTest}$, as well as $\text{Metric}_{OurOpenset}$ - $\text{Metric}_{BenchmarkOpenset}$. Resulting values were arranged in descending order, what facilitates ranking the time series according to the magnitude of their metric differences.

Fig. 4 illustrates that the increments in metric values tend to exceed the decrements in magnitude. Even with a noticeable concentration of increments around zero, the F1 score and Accuracy increment averages are 0.152 and 0.006, respectively. Additionally, the presence of outliers in the decrements suggest that a deeper investigation may bring some improvement. These findings imply that our novel approach performs equally with or surpasses the benchmark, indicating its effectiveness and potential superiority in OSR tasks.

Fig. 5 and table 6 show that the quality metrics of the alternative method is less impacted by the time series's degree of openness than the benchmark. Furthermore, it is evident that our alternative method consistently produces quality metrics that are either superior or similar with those achieved by the benchmark for all splits. These findings suggest the efficacy and resilience of the proposed approach in time series open-set recognition, particularly in the face of varying levels of dataset openness, what can be found in real-world scenarios.

Table 4: F1-Score comparative analysis of our method against the benchmark proposed by [17]. The benchmark uses only two unknown time series, whereas our approach includes multiple unknown instances and introduces an open-set split. Both methods compute overall quality metrics by averaging individual metrics across splits

Time Series	[17]	Our Test	Our Open-set
ArticularyWordRecognition	**0.96**	0.89	0.78
AtrialFibrillation	0.18	**0.79**	0.83
BasicMotions	0.82	**0.86**	0.87
CharacterTrajectories	**0.96**	0.93	1.00
Coffee		**0.96**	1.00
Cricket	0.68	**0.94**	1.00
DuckDuckGeese	0.64	**0.91**	0.81
EigenWorms	**0.85**		
Epilepsy	0.82	**0.85**	0.66
ERing	**0.86**	0.45	1.00
EthanolConcentration	0.38	**0.97**	1.00
FaceDetection	**0.54**		
FingerMovements	**0.62**	0.59	1.00
HandMovementDirection	0.42	**0.99**	1.00
Handwriting	0.43	**0.43**	0.80
Heartbeat	**0.54**	0.38	1.00
InsectWingbeat	**0.65**		
JapaneseVowels	**0.95**		
Libras	0.80	**0.99**	1.00
LSST	**0.36**		
MotorImagery	0.53	**0.72**	0.33
NATOPS	0.89	**0.94**	1.00
PEMS-SF	0.87	**0.88**	1.00
PenDigits	**0.95**	0.90	1.00
PhonemeSpectra	0.37	**0.85**	1.00
RacketSports	**0.85**	0.77	1.00
RefrigerationDevices		**0.84**	0.60
Rock		**0.73**	0.88
ScreenType		**0.83**	0.62
SelfRegulationSCP1	0.46	**0.80**	0.88
SelfRegulationSCP2	0.54	**0.84**	0.76
ShapesAll		**0.79**	1.00
SmallKitchenAppliances		**0.78**	0.74
SpokenArabicDigits	**0.98**	0.82	0.87
StandWalkJump	0.17	**0.92**	1.00
SwedishLeaf		**0.94**	1.00
TwoPatterns		**0.74**	0.69
UWaveGestureLibrary	0.79	**0.90**	1.00

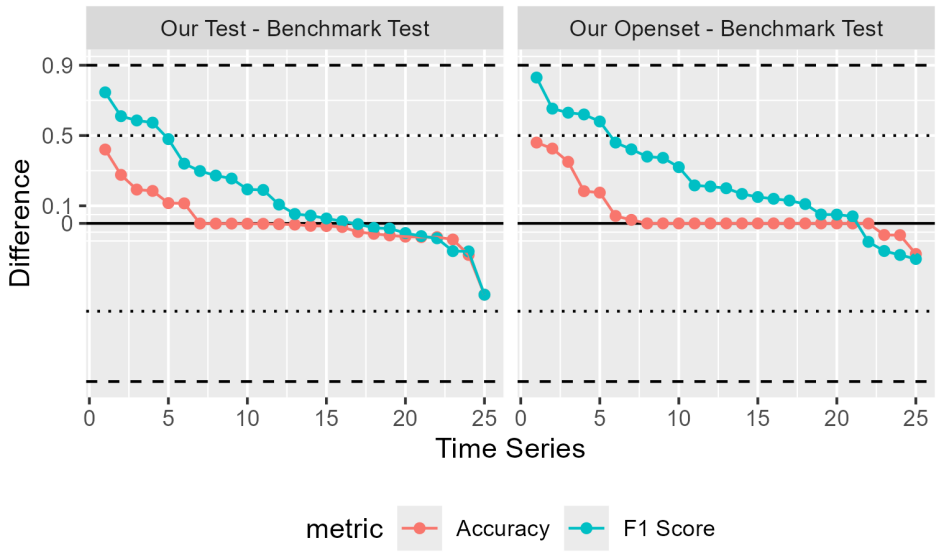

Figure 4: Comparison of metric increment between our method and the benchmark, showing $\text{Metric}_{OurTest}$ - $\text{Metric}_{BenchmarkTest}$ and $\text{Metric}_{OurOpenset}$ - $\text{Metric}_{BenchmarkOpenset}$, arranged in descending order.

Table 5: Accuracy Score comparative analysis of our method against the benchmark proposed by [17]. The benchmark uses only two unknown time series, whereas our approach includes multiple unknown instances and introduces an open-set split. Both methods compute overall quality metrics by averaging individual metrics across splits

Time Series	[17]	Our Test	Our Open-set
ArticularyWordRecognition	**1.00**	0.92	0.93
AtrialFibrillation	**1.00**	0.91	1.00
BasicMotions	0.80	**0.99**	0.98
CharacterTrajectories	**1.00**	**1.00**	1.00
Coffee		**1.00**	1.00
Cricket	**1.00**	**1.00**	1.00
DuckDuckGeese	**1.00**	0.98	0.83
EigenWorms	**1.00**		
Epilepsy	0.81	**0.99**	0.99
ERing	**0.98**	0.91	1.00
EthanolConcentration	**1.00**	0.98	1.00
FaceDetection	**1.00**		
FingerMovements	**1.00**	0.59	1.00
HandMovementDirection	**1.00**	**0.99**	1.00
Handwriting	**1.00**	0.82	0.93
Heartbeat	0.54	0.66	1.00
InsectWingbeat	0.92		
JapaneseVowels	**1.00**		
Libras	**1.00**	**1.00**	1.00
LSST	0.00		
MotorImagery	0.54	**0.96**	0.97
NATOPS	**1.00**	**1.00**	1.00
PEMS-SF	**1.00**	0.92	1.00
PenDigits	**1.00**	**1.00**	1.00
PhonemeSpectra	**1.00**	**0.95**	1.00
RacketSports	**1.00**	0.92	1.00
RefrigerationDevices		**0.99**	0.84
Rock		**0.87**	0.97
ScreenType		**0.97**	0.98
SelfRegulationSCP1	0.61	**0.89**	0.96
SelfRegulationSCP2	**0.86**	**0.97**	0.90
ShapesAll		**0.97**	1.00
SmallKitchenAppliances		**0.88**	0.95
SpokenArabicDigits	**1.00**	**0.99**	1.00
StandWalkJump	**1.00**	0.94	1.00
SwedishLeaf		**1.00**	1.00
TwoPatterns		**1.00**	0.98
UWaveGestureLibrary	**1.00**	**1.00**	1.00

Figure 5: Quality Metrics vs. Openness. Loess curves [28] applied to smooth values.

Table 6: Openness of data splits for each time series in the experiment

Original Ts	Benchmark Test	Our Train	Our Test	Our Openset
ArticularyWordRecognition	0.14	0.46	0.46	1.00
AtrialFibrillation	0.29	0.81	0.73	1.00
BasicMotions	0.33	0.75	0.75	1.00
CharacterTrajectories	0.38	0.50	0.53	1.00
Cricket	0.07	0.49	0.65	1.00
DuckDuckGeese	0.00	0.72	0.66	1.00
Epilepsy	0.65	0.55	0.78	1.00
ERing	0.11	0.48	0.55	1.00
EthanolConcentration	0.40	0.77	0.67	1.00
FingerMovements	0.00	0.81	0.55	1.00
HandMovementDirection	0.27	0.75	0.64	1.00
Handwriting	0.17	0.44	0.48	1.00
Heartbeat	0.50	0.71	0.68	1.00
Libras	0.09	0.58	0.60	1.00
MotorImagery	0.50	0.70	0.61	1.00
NATOPS	0.07	0.65	0.67	1.00
PEMS-SF	0.14	0.73	0.54	1.00
PenDigits	0.00	0.57	0.68	1.00
PhonemeSpectra	0.06	0.31	0.26	1.00
RacketSports	0.31	0.69	0.80	1.00
SelfRegulationSCP1	0.53	0.81	0.85	1.00
SelfRegulationSCP2	0.29	0.83	0.80	1.00
SpokenArabicDigits	0.00	0.53	0.58	1.00
StandWalkJump	0.45	0.81	0.78	1.00
UWaveGestureLibrary	0.48	0.72	0.65	1.00

6 Conclusion

In this study we introduced a new methodology for Open Set Recognition in time series classification, employing a blending of various Deep Neural Networks equipped with an OpenMax layer. This approach demonstrated a performance comparable to or exceeding that of the benchmark used, achieving an average F1 Score of 0.82, which is 0.16 higher. Furthermore, the proposed model is more robust against varying degrees of openness and offers a more simplified and straightforward implementation. The model's consistent/superior overall quality supports its choice for openset recognition tasks.

Future works could leverage the activations of intermediate layers to explore further potential improvements, test other models to be used as the blending, and more extensive exploration of the impact of number of known-unknown time series in the overall quality. A further challenging research is to train the neural networks from the scratch, or just the last layer, using the OpenMax as the activation function, aiming to optimize it for open set recognition.

REFERENCES

[1] W. K. Wang, I. Chen, L. Hershkovich *et al.*. "A Systematic Review of Time Series Classification Techniques Used in Biomedical Applications". *Sensors*, vol. 22, no. 20, 2020.

[2] M. Gutoski, A. E. Lazzaretti and H. S. Lopes. "Unsupervised open-world human action recognition". *Pattern Analysis and Applications*, vol. 26, pp. 1753–1770, 2023.

[3] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller and F. Petitjean. "InceptionTime: Finding AlexNet for time series classification". *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[4] H. Oh and S. B. Kim. "Multivariate Time Series Open-Set Recognition Using Multi-Feature Extraction and Reconstruction". *IEEE Access*, vol. 10, 2022.

[5] L. Puppo, W.-K. Wong, B. Hamdaoui and A. Elmaghbub. "HiNoVa: A Novel Open-Set Detection Method for Automating RF Device Authentication". In *Proc. IEEE Symposium on Computers and Communications (ISCC)*, pp. 1122–1128, 2023.

[6] M. Romero, M. Gutoski, L. T. Hattori, M. Ribeiro and H. S. Lopes. "A study of the influence of data complexity and similarity on soft biometrics classification performance in a transfer learning scenario". *Learning and Nonlinear Models*, vol. 18, no. 2, pp. 56–65, 2020.

[7] A. Bagnall, J. Lines, A. Bostrom, J. Large and E. Keogh. "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances". *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[8] T. Mensink, J. Verbeek, F. Perronnin and G. Csurka. "Distance-based image classification: Generalizing to new classes at near-zero cost". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2624–2637, 2013.

[9] E. M. Rudd, L. P. Jain, W. J. Scheirer and T. E. Boult. "The Extreme Value Machine". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 762–768, 2018.

[10] M. Pal. "Support vector machines/relevance vector machine for remote sensing classification: A review". *arXiv preprint 1101.2987*, 2011.

[11] A. Bendale and T. E. Boult. "Towards Open Set Deep Networks". In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1563–1572, 2016.

[12] A. Rozsa, M. Günther and T. E. Boult. "Adversarial robustness: Softmax versus openmax". *arXiv preprint 1708.01697*, 2017.

[13] C. Geng, S.-J. Huang and S. Chen. "Recent Advances in Open Set Recognition: A Survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3614–3631, 2021.

[14] J. Sill, G. Takacs, L. Mackey and D. Lin. "Feature-Weighted Linear Stacking". *arXiv preprint 0911.0460*, 2009.

[15] H. Sakoe and S. Chiba. "Dynamic programming algorithm optimization for spoken word recognition". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[16] J. Faouzi. "Time Series Classification A review of Algorithms and Implementations". In *Machine Learning Emerging Trends and Applications*, edited by K. Kotecha. Proud Pen, 2022.

[17] T. Akar, T. Werner, V. K. Yalavarthi and L. Schmidt-Thieme. "Open Set Recognition for Time Series Classification". In *Advances in Knowledge Discovery and Data Mining*, edited by J. Gama, T. Li, Y. Yu, E. Chen, Y. Zheng and F. Teng, pp. 354–366. Springer, 2022.

[18] J. L. E. Shelhamer and T. Darrell. "Fully Convolutional Networks for Semantic Segmentation". In *Proc. IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 3431–3440, 2015.

[19] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] F. Karim, S. Majumdar, H. Darabi and S. Chen. "LSTM Fully Convolutional Networks for Time Series Classification". *IEEE Access*, vol. 6, pp. 1662–1669, 2018.

[21] J. Wang, Z. Wang, J. Li and J. Wu. "Multilevel Wavelet Decomposition Network for Interpretable Time Series Analysis", 2018.

[22] W. Tang, G. Long, L. Liu, T. Zhou, M. Blumenstein and J. Jiang. "Omni-Scale CNNs: a simple and effective kernel size configuration for time series classification". In *Proc. International Conference on Learning Representations*, pp. 1–7, 2022.

[23] L. Wen, Y. Dong, and L. Gao. "A New Ensemble Residual Convolutional Neural Network for Remaining Useful Life Estimation". *Mathematical Biosciences and Engineering*, vol. 16, no. 2, pp. 862–880, 2019.

[24] K. He, X. Zhang, S. Ren and J. Sun. "Deep Residual Learning for Image Recognition". In *Proc. IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 770–778, 2015.

[25] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif and A. Mohammadi. "Xceptiontime: independent time-window xceptiontime architecture for hand gesture classification". In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1304–1308, 2020.

[26] T. Chen and C. Guestrin. "XGBoost: A Scalable Tree Boosting System". In *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 785–794, 2016.

[27] I. Oguiza. "tsai - A state-of-the-art deep learning library for time series and sequential data". Github, 2023.

[28] M. D. Cattaneo, M. Jansson and X. Ma. "Local Regression Distribution Estimators", 2021.