

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL**

LEONARDO DE MARQUI MARQUES

***BLENDING ENSEMBLE APLICADO EM RECONHECIMENTO DE
CONJUNTO ABERTO PARA CLASSIFICAÇÃO DE SÉRIES
TEMPORAIS***

DISSERTAÇÃO

**CURITIBA
2024**

LEONARDO DE MARQUI MARQUES

***BLENDING ENSEMBLE APLICADO EM RECONHECIMENTO
DE CONJUNTO ABERTO PARA CLASSIFICAÇÃO DE SÉRIES
TEMPORAIS***

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

Dissertação apresentado(a) como requisito para obtenção do título(grau) de Mestre em Engenharia Elétrica e Informática Industrial, do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. André Eugenio Lazaretti

Coorientador: Prof. Dr. Heitor Silvério Lopes

CURITIBA

2024



[4.0 Internacional](#)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es).

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



LEONARDO DE MARQUI MARQUES

**BLENDING ENSEMBLE APLICADO EM RECONHECIMENTO DE CONJUNTO ABERTO PARA
CLASSIFICAÇÃO DE SÉRIES TEMPORAIS.**

Trabalho de pesquisa de mestrado apresentado como
requisito para obtenção do título de Mestre Em Ciências da
Universidade Tecnológica Federal do Paraná (UTFPR). Área
de concentração: Engenharia De Computação.

Data de aprovação: 26 de Agosto de 2024

Dr. Andre Eugenio Lazzaretti, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Guilherme De Alencar Barreto, Doutorado - Universidade Federal do Ceará (Ufc)

Matheus Henrique Dal Molin Ribeiro, - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 26/08/2024.

RESUMO

MARQUES, Leonardo de Marqui. *Blending Ensemble aplicado em reconhecimento de conjunto aberto para classificação de séries temporais*. 2024. 74 f. Dissertação (Mestrado em Engenharia Elétrica e Informática Industrial) – Universidade Tecnológica Federal do Paraná. Curitiba, 2024.

Esta dissertação de mestrado investiga a aplicação de métodos de Open Set Recognition (OSR) na Classificação de Séries Temporais (TSC), abordando uma área pouco explorada na literatura existente. Enquanto a TSC envolve a classificação de instâncias de séries temporais, os métodos OSR são projetados para classificar simultaneamente amostras conhecidas e detectar as desconhecidas. Esse tema é relevante, pois permite que os modelos identifiquem padrões conhecidos e detectem padrões desconhecidos em ambientes dinâmicos e imprevisíveis, aumentando a robustez e a confiabilidade em aplicações do mundo real que estão repletas de dados de séries temporais. Apesar de seu potencial, os métodos OSR têm aplicação limitada na TSC em comparação com outras áreas, como classificação de texto ou imagem. Pesquisas recentes têm introduzido metodologias inovadoras para abordar essa lacuna, cada qual com seus pontos fortes e limitações. No entanto, os métodos OSR existentes na TSC muitas vezes ignoram o uso de "desconhecidos conhecidos" durante o treino, o que poderia aumentar a robustez do modelo. Além disso, a abordagem de transfer learning aplicado a vários modelos de rede neural requer melhorias, e os experimentos com conjuntos de dados de referência são limitados. Em resposta a esses desafios, esta pesquisa propõe uma abordagem inovadora que integra redes neurais, modelos que historicamente mostram boa performance em TSC, com camadas OpenMax, projetadas para dar às redes neurais a capacidade de trabalhar com OSR, fornecendo covariáveis para treinar um modelo de blending, e, ao mesmo tempo, incorporando instâncias "conhecidas desconhecidas" durante o treinamento. Para avaliar rigorosamente a eficácia do modelo proposto, são conduzidos experimentos abrangentes e extensos em várias configurações usando diversos conjuntos de dados de séries temporais de diferentes áreas, bem como comparando seu desempenho com trabalho anterior semelhante. Os resultados desses experimentos fornecem insights valiosos sobre o desempenho e a aplicabilidade da abordagem proposta em cenários do mundo real, contribuindo para o avanço tanto da Classificação de Séries Temporais quanto das metodologias de Reconhecimento de Conjuntos Abertos.

Palavras-chave: Conjunto aberto. Séries temporais. Classificação. Aprendizado profundo. Redes neurais.

ABSTRACT

MARQUES, Leonardo de Marqui. **Blending Ensemble applied to Open-Set Recognition for Time Series Classification.** 2024. 74 p. Dissertation (Master's Degree in Course Name) – Universidade Tecnológica Federal do Paraná. Curitiba, 2024.

This master thesis investigates the application of Open-Set Recognition (OSR) methods to Time Series Classification (TSC), addressing a gap in the existing literature. While TSC involves labeling time series instances, OSR methods are designed to simultaneously classify known samples and detect unknown ones. This is relevant because it enables models to identify known and detect unknown patterns in dynamic and unpredictable environments, enhancing robustness and reliability in real-world applications that have no shortage of time series data. Despite their potential, OSR methods have seen limited application in TSC compared to other domains such as text or image classification. Recent research has introduced innovative methodologies to address this gap, each with its strengths and limitations. However, existing OSR methods in TSC often overlook the utilization of "known unknowns" during training, which could enhance model robustness. Moreover, the transfer-learning approach across various neural network models requires improvement, and experiments with benchmark datasets are limited in scope. In response to these challenges, this research proposes a novel approach that integrates neural networks, models that have a proven record of high performance in TSC, with OpenMax layers, designed to make neural networks capable of working in open-set recognition, both providing features to train a blending model, while also incorporating known unknown instances during training. To rigorously evaluate the effectiveness of the proposed model, comprehensive and extensive experiments are conducted across various configurations using diverse time series datasets from different domains as well as comparing its performance against previous similar work. The results of these experiments provide valuable insights into the performance and applicability of the proposed approach in real-world scenarios and, contributing to the advancement of both Time Series Classification and Open-Set Recognition methodologies.

Keywords: Open set. Time series. Classification. Deep learning. Neural networks.

LIST OF ALGORITHMS

LIST OF FIGURES

Figure 1 – Demosntration of FCN	17
Figure 2 – ResCNN Residual Block	18
Figure 3 – ResNet Residual Block	19
Figure 4 – ResCNN Architecture	19
Figure 5 – LSTM-FCN architecture	21
Figure 6 – An overview of the issue with OSR	22
Figure 7 – Example of the EVT distributions	24
Figure 8 – Samples from different distributions. The black line marks the extreme values	25
Figure 9 – EVT distributions fitted to extreme values of the samples	26
Figure 10 – Simulated data	29
Figure 11 – Well Behaved Prediction Surface	30
Figure 12 – Not Well Behaved Prediction Surface	30
Figure 13 – Prediction Surfaces for Different Trials	31
Figure 14 – Euclidian Distance and Dynamic Time Wrapping	35
Figure 15 – Example of Shapelets applied to ECG data	36
Figure 16 – EVM 0.95 quantiles	38
Figure 17 – EVM Prediction Grid	39
Figure 18 – HiNoVa hidden layer activations	42
Figure 19 – Overview of training setup	47
Figure 20 – Blending Feature Extractor	48
Figure 21 – Data Set Split Overview	48
Figure 22 – Accuracy of blending and Split strategy	50
Figure 23 – Experiment Coverage	51
Figure 24 – Overview of Quality Metrics	56
Figure 25 – Scatter plot of metrics	59
Figure 26 – Correlation Metrics	60
Figure 27 – Violin plot of metrics	61
Figure 28 – Comparison overview between our novel method and the benchmark	63
Figure 29 – Increment comparison	64
Figure 30 – Increment Distribution	64
Figure 31 – Quality Metrics VS Openness	67

LIST OF TABLES

Table 1 – Time Series Metadata	45
Table 2 – Openness	52
Table 3 – F1 Score	54
Table 4 – Accuracy Score	55
Table 5 – Precision Score	56
Table 6 – Recall Score	57
Table 7 – Pearson Correlation	61
Table 8 – Kendall Correlation	62
Table 9 – Spearman Correlation	62
Table 10 – Compare Accuracy Score	65
Table 11 – Compare F1 Score	66

CONTENTS

1	INTRODUCTION	10
1.1	OBJECTIVES	12
1.1.1	General Objective	12
1.1.2	Specific Objectives	13
1.2	METHODOLOGICAL PROCEDURES	13
1.3	DOCUMENT STRUCTURE	14
2	THEORETICAL ASPECTS	15
2.1	TIME SERIES CLASSIFICATION	15
2.2	CONVOLUTIONAL NEURAL NETWORKS	16
2.2.1	Fully Convolutional Networks	16
2.2.2	InceptionTime	17
2.2.3	XceptionTime	17
2.2.4	ResNet	18
2.2.5	ResCNN	18
2.2.6	Omni-Scale Convolutional Neural Network (OmniScaleCNN)	19
2.2.7	Multilevel Wavelet Decomposition Network (mWDN)	20
2.3	LONG SHORT-TERM MEMORY NEURAL NETWORKS	20
2.3.1	LSTM-FCN	21
2.4	OPEN SET RECOGNITION	22
2.4.1	Open set recognition vs. Anomaly Detection	23
2.5	EXTREME VALUE THEORY	23
2.6	OPENMAX LAYER	27
2.6.1	Fundamentals	27
2.6.2	Prediction Surface	28
2.6.3	Limitations	30
2.7	BLENDINGS	32
2.7.1	XGBoost	32
3	RELATED WORK	34
3.1	TIME SERIES CLASSIFICATION	34
3.2	OPEN SET RECOGNITION	36
3.2.1	Extreme Value Machine	36
3.3	OPEN SET RECOGNITION ON TIME SERIES CLASSIFICATION	41
4	METHODOLOGY	44
4.1	DATA	44
4.2	MODELS	45
4.3	EXPERIMENTAL SETUP	46
4.3.1	Training Neural Networks	46
4.3.1.1	Hyperparamenters	46
4.3.2	Training Blendings	47
4.3.3	Performance Measures	48
4.3.4	Openess	51

5	RESULTS	53
5.1	OVERVIEW	53
5.2	BENCHMARK	62
6	CONCLUSION	68
	REFERENCES	70

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 (W.K., 2022). Such classification may be useful for comparing the behavior of different TS, making it an important problem in machine learning and, therefore, justifying efforts for its development and improvement.

For instance, Das (2024) introduced a class-balancing instance selection algorithm for TSC to integrate with Active Learning (AL) strategies, demonstrating this approach's effectiveness in tactile texture recognition in robotics and fault detection in synthetic fiber manufacturing. The Shapelet Transformer (ShapeFormer) proposed by Le *et al.* (2024) incorporates class-specific and generic transformer modules to capture discriminative shapelets from the training set. By introducing a Shapelet Filter, the model learns the differences between these shapelets and the input time series, effectively leveraging both types of features to enhance classification performance. Experiments demonstrate that ShapeFormer achieves the highest accuracy compared to state-of-the-art methods. In the realm of interval-based algorithms for TSC, Dempster *et al.* (2023) introduces Quant, a simplified interval method, which uses only quantile features, fixed intervals, and a standard classifier without requiring separate interval or feature selection, can match the accuracy of the most precise current interval methods. Experiments show that, unlike many state-of-the-art TSC methods that demand substantial computational resources, Quant is not only simpler but also offers a significant improvement in accuracy relative to computational cost. Romanova (2024) explores the application of Graph Neural Networks (GNN) for TSC. The approach involves transforming the TS into a graph and adding virtual nodes before employing the GNN. The research has yielded good accuracy results and highlights GNN's adaptability and sensitivity to graph topology, which, although initially concerning, proved effective for outlier detection.

On the one hand, in general data classification, as presented so far, the classes of the data instances are known beforehand. 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 compared to other areas, such as text or image classification. Nevertheless, this gap has been addressed by recent research introducing innovative methodologies. For instance, Open Set InceptionTime (FAWAZ *et al.*, 2020) uses barycenters to compute Dynamic Time Warping (DTW) distances and cross-correlations, identifying unknown samples based on threshold criteria. The MEROs method (OH; KIM, 2022) 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 (PUPPO *et al.*, 2023) applies a Convolutional Neural Network Long Short-Term Memory (CNN+LSTM) model, generating features by aggregating hidden state values and using Kendall's correlation for classification. In (LIU *et al.*, 2024), researchers explore open-set classification (OSC) for music genres using softmax thresholding and Openmax on two datasets, evaluating metrics like accuracy, precision, and recall, and analyzing threshold impacts. Traditional multi-label audio classification (AC), applied to urban sounds, everyday environments, and music, usually assumes a fixed class vocabulary. To address this, (SRIDHAR; CARTWRIGHT, 2023) introduces OSC using softmax thresholding and Openmax across five datasets, effectively identifying unknown sound events in polyphonic audio. Additionally, (YOU *et al.*, 2024) integrates deep learning with center loss and supervised contrastive loss, improving model performance for genre classification, vocal style differentiation, and acoustic scene classification and setting the stage for future improvements in unknown sample detection employing self-supervised learning.

The above-mentioned methods come with some theoretical and practical limitations. The most notable are:

- **Known Unknowns:** Incorporating “known unknowns” into the training process is crucial for improving model robustness and generalization. Known unknowns represent data points that are recognized as different from the training data but still within possible observations. By including these in training, models can learn the difference between the patterns of known and unknown instances more effectively, which improves the ability to handle real-world data variations and anomalies.
- **Blending Techniques:** The use of blending techniques proved to be a key aspect for optimizing the performance of machine learning models. By employing these techniques, we can combine the strengths of different models to get better prediction results. Coming up with a deterministic rule of how to jointly use the predictions of models is not a trivial task. With blendings, we can leverage the unique strengths of each model and benefit from

the collective knowledge to make the final prediction while reducing the risk of errors of any single model individually.

- **Transfer Learning:** Transfer learning is a good approach to using the power of advanced neural network architectures. Instead of training models from scratch, which can be computationally costly, transfer learning allows us to use pre-trained models that have already learned useful features from several datasets. This not only accelerates the training process but also improves the performance of the models by using neural network features learned from other tasks.
- **Benchmark Datasets:** Conducting experiments with existing benchmark datasets is fundamental for evaluating the performance of models and their ability to generalize. Benchmark real-world datasets provide a standardized way to test models against various data types and complexities. By doing so, we can ascertain the models' capability to handle different kinds of data and ensure their effectiveness in diverse real-world scenarios.

In order to advance the topics presented above as limitations of other works, this research introduces a novel approach to applying Opens Set Recognition (OSR) to TSC by 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.

1.1 OBJECTIVES

Given the limitations of the current approaches and the urge to contribute to the state-of-the-art TS classification, we have come up with the following objectives for this work.

1.1.1 General Objective

The general objective is to construct a model that can classify previously known time series and identify unknown ones.

1.1.2 Specific Objectives

- Extensively apply transfer learning across various neural networks using several Time Series data;
- Integrate the classification and detection capabilities of the models through blending ensembles;
- Incorporate known-unknown data into the machine learning engineering pipeline to test the potential improvements;
- Apply all the above using a wide and diverse set of Time Series Data in exhaustive experiments to evaluate the effectiveness of the proposed model rigorously;
- Compare the performance against a benchmark (AKAR *et al.*, 2022) to asses the performance of the proposed method;
- Conduct exploratory data analysis to assess the relation between metrics regarding the data and the model results.

1.2 METHODOLOGICAL PROCEDURES

This study begins with the comprehensive collection of Time Series data from various domains, ensuring a diverse representation of real-world scenarios. Following this, we craft the architecture of the Neural Networks employed in the subsequent analysis. Once the NN architectures are defined, a machine-learning pipeline is applied to each Time Series. This pipeline involves fitting the Neural Networks augmented with an OpenMax layer designed to enhance the model's ability to discriminate between known and unknown TS samples. Leveraging this pipeline, the last activations of the Neural Networks, alongside the corresponding OpenMax layer predictions, are extracted for each Time Series in the dataset, taking Time Series not used in training as known unknowns used in the test and open set splits. These extracted features serve as the features for constructing a robust blending model using the state-of-the-art XGBoost algorithm to harness the strengths of all Neural Networks, thereby improving the overall classification performance. The resulting predictions are analyzed and evaluated according to the chosen quality metrics. Furthermore, the performance of the proposed approach is rigorously

benchmarked against an alternative methodology, yielding insights about differences in efficacy and generalization capabilities.

1.3 DOCUMENT STRUCTURE

Following this introduction, the work is structured as follows: In Chapter 2, most of the main theoretical aspects used in this work are explained. Next, Chapter 3 presents some works directly related to the theme of Open-set recognition and Time Series Classification. Chapter 4 follows with the presentation of the methodology employed for the completion of the work. Subsequently, Chapter 5 presents the main results obtained in the research, and Chapter 6 presents conclusions discussed, as well as suggestions for future work, in addition to the citation of related works published during this research.

2 THEORETICAL ASPECTS

This chapter presents an overview of the theoretical aspects behind this research. It begins by defining and explaining TSC and the time dependency of the data. Next, it presents Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) Neural Networks with detailed subsections on the architectures employed, showing their versatility and application in feature extraction and classification tasks within TSC. The subsequent sections present Open Set Recognition and Extreme Value Theory, building the base for understanding the importance of OpenMax Layer in OSR. It also shows the fundamentals, prediction surface, and limitations of the OpenMax Layer. In the end, the chapter explores blending techniques, emphasizing XGBoost and demonstrating its effectiveness in improving prediction performance.

2.1 TIME SERIES CLASSIFICATION

When an event has values dependent on time, we can study it as a TS. Stock prices, temperatures, and health markers are some of the many everyday life examples of them. TSC involves building a model that assigns a label to an instance of a TS. TSC is different from conventional classification, because the attributes are, by nature, time-dependent (BAGNALL *et al.*, 2017). Below is a formal definitions that serves as the groundwork for understanding TSC (FAWAZ *et al.*, 2019):

- **Definition 1: Univariate Time Series:** A univariate time series, denoted as $X = [x_1, x_2, \dots, x_T]$, is an ordered sequence of real values. The length of this time series, T , corresponds to the total number of real values it comprises.
- **Definition 2: Multivariate Time Series (MTS):** An M -dimensional MTS, labeled as $X = [X_1, X_2, \dots, X_M]$, is consisted of M distinct univariate Time Series, with each $X_i \in \mathbb{R}^T$.
- **Definition 3: Dataset in TSC:** A dataset of N samples, represented as $D = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_N, Y_N)\}$, is a collection pairs, where X_i can either be a univariate or multivariate time series, and Y_i is the corresponding one-hot label vector, also known as dummy coding or contrast. In a dataset containing K classes, the one-hot label

vector Y_i has a length of K , where each element, denoted as $j \in [1, K]$, is assigned the value 1 if the class of X_i is j , and 0 otherwise.

TSC consists of training a classifier using a dataset D , with the aim of modeling the input variables (X) to a probability distribution over the class variable values (Y). This mapping enables the classifier to assign probabilities to the classes to classify the Time Series. Although this classifier can be of any nature we focus on DNN.

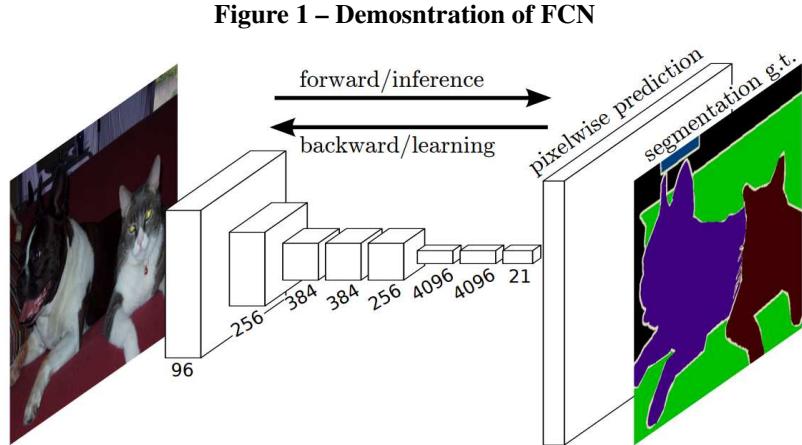
2.2 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs) (LI *et al.*, 2020) are one of the most significant advancements in the field of deep learning, especially in computer vision, image processing, and pattern recognition. They are inspired by the visual processing mechanisms of the human brain, which makes them capable of automatically learning and extracting spatial hierarchies of features from input data. Unlike conventional neural networks, CNNs use specialized layers, such as convolutional and pooling layers, to capture spatial dependencies and patterns within images. The consecutive application of convolutional kernels makes CNNs proficient at extracting increasingly abstract and complex features as activations propagate through the multiple layers of the neural network, enabling them to recognize patterns and shapes in images. Such capabilities led to their widespread adoption and success in various real-world applications, such as image classification and object detection to audio processing. The continuous evolution of CNN architectures, as well as advancements in computing power and training techniques, further improve their capabilities and broaden their applicability across various domains, reaffirming their role in machine learning. A short introduction to the CNNs employed in this work is presented as follows.

2.2.1 Fully Convolutional Networks

Over the years, Fully Convolutional Networks (WANG *et al.*, 2016) (SHELHAMER *et al.*, 2016) have proven to be a robust architecture for image semantic segmentation tasks. Essentially, its architecture is built so that it can make predictions at the pixel level, effectively assigning semantic labels to individual image regions, thus giving them the capability of delineation of object boundaries (figure 1 exemplifies how it works in segmentation context). FCNs can capture spatial relationships; this capability can be used to capture the inherent time dependency of Time Series. In the context of TSC, FCNs are employed as a tool to extract features that are

ultimately used on a softmax layer.



2.2.2 InceptionTime

The InceptionTime model, introduced by (FAWAZ *et al.*, 2020), is framed as the AlexNet equivalent for TSC, constructed as an ensemble of deep Convolutional Neural Network (CNN) models. Unlike traditional fully convolutional layers, Inception Network adopts Inception modules, followed by a Global Average Pooling (GAP) layer and a softmax layer. The Inception module serves as a dimensionality reduction block, leveraging a bottleneck layer to apply convolution with m filters of length 1 and stride 1, effectively reducing the time series dimension from M to m . This helps mitigate overfitting issues and reduces computational complexity. The subsequent series of filters with varying lengths stacked within the Inception module extract features of different resolutions and importance levels, enhancing the model’s predictive capacity. InceptionTime, tailored specifically for TSC, comprises an ensemble of Inception networks that demonstrate superior speed compared to the benchmark models used for evaluation.

2.2.3 XceptionTime

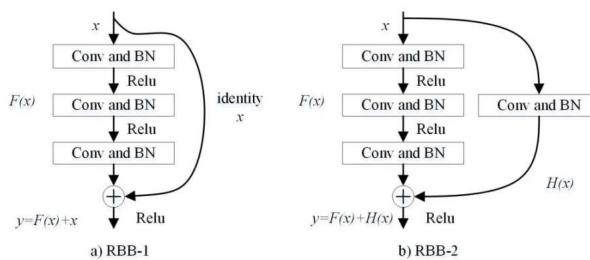
The XceptionTime module (RAHIMIAN *et al.*, 2019) distinguishes itself from the InceptionTime module by applying depthwise separable convolutions, which reduces the number of parameters and improves computational efficiency. Depthwise separable convolutions involve two stages: Depthwise Convolution, where each input channel is convolved independently, and Pointwise Convolution, which uses 1×1 convolutions to combine the convolved channels into a new depth. The module processes the input through two parallel paths: one with a bottleneck

layer followed by depthwise separable convolutions with kernel sizes of 11, 21, and 41, and another with a MaxPooling layer followed by a 1x1 convolution layer. This architecture allows for efficient feature extraction from the input time series data, balancing depth and computational efficiency.

2.2.4 ResNet

The Residual Network (HE *et al.*, 2015) architecture has significantly advanced deep learning by effectively handling the problem of vanishing gradient, which is a common challenge in training very deep neural networks. The architecture, also known as ResNet, address this problem by introducing a deep residual learning framework: Residual Block. Instead of directly learning a function that maps input to output, it makes these layers fit a residual mapping. Formally, if we denote the desired underlying mapping as $H(x)$, now the layers are designed to fit another mapping, $F(x) := H(x) - x$, recasting the original mapping as $F(x) + x$, see figure 3. This technique simplifies optimization by making it easier to push the residual to zero instead of fitting an identity mapping with nonlinear layers. The $F(x) + x$ formulation is implemented with shortcut connections which skip one or more layers, performing identity mapping. These connections add their outputs to the outputs of the stacked layers, introducing no extra parameters or computational complexity, while still being able to be trained using gradient descent with back propagation

Figure 2 – ResCNN Residual Block



2.2.5 ResCNN

ResCNN (WEN *et al.*, 2019) also uses the framework of Residual Blocks present in ResNet, but instead of having the identity function as the shortcut it applies a convolution and a batch normalization (see figure 3). Figure 4 shows the final architecture, which consists of concatenating N_{blocks} Residual Blocks, N_{blocks} being a hyperparameter of the model. ResCNN

still handles the problem of vanishing gradient while also yielding robust and reliable results especially when employed in Ensemble models.

Figure 3 – ResNet Residual Block

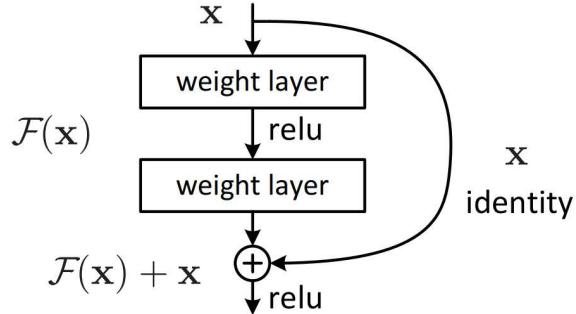
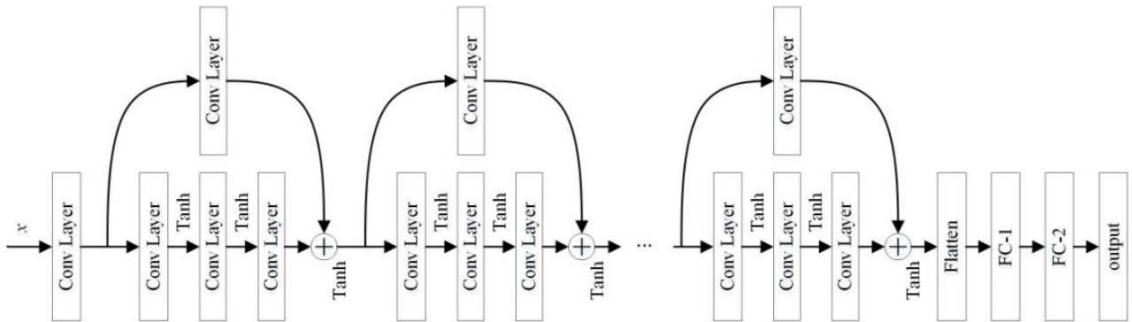


Figure 4 – ResCNN Architecture



2.2.6 Omni-Scale Convolutional Neural Network (OmniScaleCNN)

An important challenge in TSC is determining the optimal receptive fields (RF) for feature extraction. Selecting the most appropriate number and sizes of RFs (kernel sizes in CNNs) is not trivial, and hyperparameter fine-tuning is often impractical due to time and computational constraints. The Omni-Scale Convolutional Neural Network (TANG *et al.*, 2022) (OmniScaleCNN) handles the challenge of searching for the optimal RFs by using an architecture that applies various kernel sizes. This architecture captures a wide range of RFs, ensuring they cover frequencies across all time scales. By using multiple kernel sizes, the OmniScaleCNN meets requirements of different datasets, making it a flexible tool for time series analysis. The Omni-Scale block (OS-block) is the key component of this approach. The kernel sizes in the OS-block are determined by a rule that uses prime numbers based on the TS length. This allows the Neural Network to efficiently cover the best RFs size across different datasets without having to apply exhaustive grid searches. Experiment results show that Neural Networks that employ

OS-block perform similarly to those with the optimally searched receptive field size which demonstrates the effectiveness of the Omni-Scale CNN architecture in a variety of benchmark TS data.

2.2.7 Multilevel Wavelet Decomposition Network (mWDN)

Traditional TS methods seldom consider the frequency information of time series. Wavelet decompositions, known for their ability to capture features in both time and frequency domains, are typically used as preprocessing tools for feature extraction. Integrating wavelet transforms into deep learning frameworks is a significant challenge. The mWDN (WANG *et al.*, 2018) addresses this by decomposing a time series into sub-series with frequencies ranked from high to low by employing functions that have trainable parameters. This approach captures critical frequency factors and takes advantage of both wavelet decomposition and the learning capabilities of deep neural networks. Therefore, mWDN improves the model's capability to learn from the frequency domain, providing a powerful tool for TS analysis. Experiments on various TS datasets demonstrate that mWDN has superior performance when compared to other models tailored for the same task while also having the advantage of interpretability.

2.3 LONG SHORT-TERM MEMORY NEURAL NETWORKS

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) architecture that is specifically designed to capture and process sequential data while handling the problem of gradient vanishing or explosion usually encountered in traditional RNNs (GHOJOGH; GHODSI, 2023). LSTMs have memory cells that can maintain information over long periods, which enable them to learn and remember dependencies in sequential data with greater efficacy. Unlike standard RNNs, LSTMs utilize a gating mechanism that regulates the flow of information into and out of the memory cells, allowing them to selectively retain or discard information at each time step. This capability makes LSTMs well-suited for a wide range of sequential tasks. The relevance of LSTMs lies in their ability to effectively model and analyze sequential data with long-range dependencies, thereby facilitating advancements in various fields and applications, such as language translation, sentiment analysis, financial forecasting, and medical diagnostics, speech recognition, time series analysis, and more. As a result, LSTMs have become important in the domain of deep learning and have significantly contributed to the

advancement of machine learning research.

2.3.1 LSTM-FCN

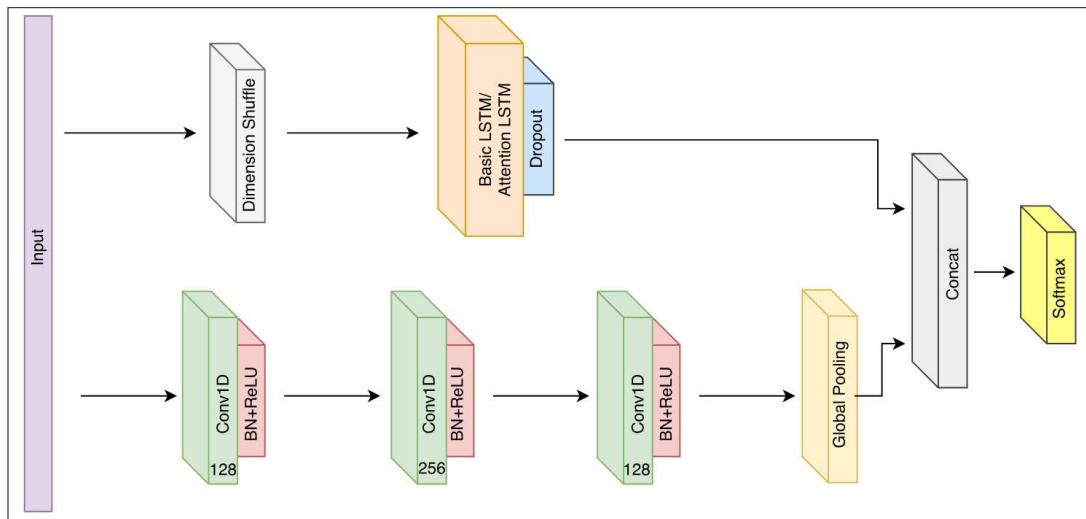
LSTM and FCN have demonstrated to be highly effective as feature extractors in TSC tasks. Building on this success, (KARIM *et al.*, 2018) improved the performance by combining both LSTM and FCN into a single neural network, leveraging the strengths of each approach. The architecture is presented in figure 5.

The FCN block includes three stacked temporal convolutional blocks with filter sizes of 128, 256, and 128, respectively. Each layer is then followed by batch normalization and a ReLU activation function, ending with a global average pooling after the final convolutional layer.

In addition to the FCN, the TS is transformed using a dimension shuffle layer, which converts a univariate TS of length N into a multivariate TS of length 1 and N dimensions. This shuffled TS is then fed into the LSTM block followed by a dropout layer. This approach significantly improves the performance of the LSTM-FCN architecture by reducing the overfitting of short TS and the underfitting in long ones.

The outputs from the FCN and the LSTM blocks are concatenated and then fed to a softmax classification layer. This combined architecture leverages the strengths of temporal convolutions for feature extraction and the LSTM's ability to capture temporal dependencies, making LSTM-FCN a powerful model that presents high performance on TSC benchmark datasets.

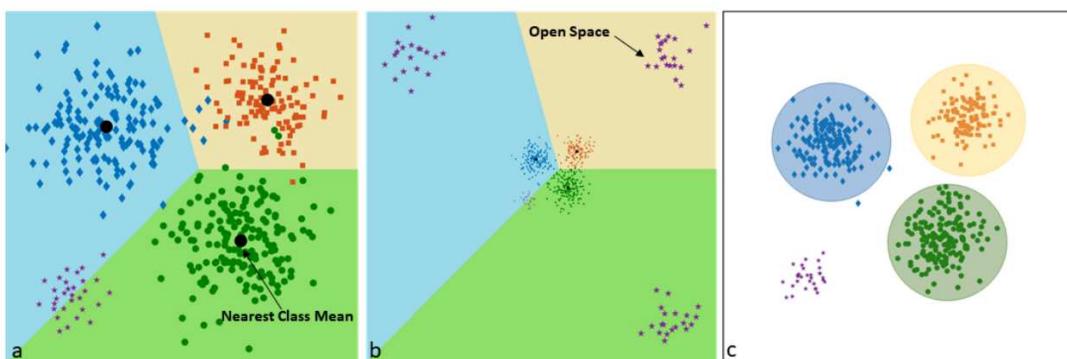
Figure 5 – LSTM-FCN architecture



2.4 OPEN SET RECOGNITION

Open-Set Recognition (OSR) presents a distinctive challenge in the field of machine learning, where traditional classification models confront the potential for unknown classes not present during their training phase, demanding the model not only to classify instances belonging to known classes but also to detect unknown classes during testing. By operating in the open set paradigm, OSR reflects a more realistic setting. In a closed-set scenario, the classifier is trained on a predefined set of M known classes denoted as $C = [c_1, c_2, \dots, c_M]$. In contrast, an OSR classifier operates within a scenario where the set of possible classes is extended to $C' = [c_1, c_2, \dots, c_M, c_{M+1}, \dots, c_{M+\Omega}]$, incorporating classes c_{M+1} through $c_{M+\Omega}$, as unknown classes. Consequently, an observation can be assigned to one of the known classes, denoted as $c_i \in C$, or categorized as unknown. This definition results in three class types: known known classes (KKC), known unknown classes (KUC), and unknown unknown classes (UUC). KKC are categories present in the training data, allowing for accurate classification. KUC are recognized as existing but are not included in the training set, while UUC are entirely novel and unforeseen. Figure 6(a) (MAHDAVI; CARVALHO, 2021) serves as an illustration of closed-set classification. The decision boundaries are established through the training of a conventional Nearest Class Mean (NCM) classifier on three known classes, symbolized by diamonds, circles, and squares; Unknown classes are represented as stars. Figure 1(b) visualizes the distribution of the original dataset within the open space, extending beyond the limits of the closed three-class model. In this context, where the classifier lacks knowledge of the complete set of potential classes, it extends class labels from the closed set across an unbounded region. Consequently, unknown observations within the open space are susceptible to misclassification. In contrast, OSR exhibits the ability to differentiate known samples and restrict decision-making to the domain supported by the training data, as depicted in Figure 1(c).

Figure 6 – An overview of the issue with OSR



2.4.1 Open set recognition vs. Anomaly Detection

Both Open set recognition (OSR) and anomaly detection identify unknown data, but they are different in terms of approach and usage. OSR is employed to classify data into known classes while detecting unknown ones. On the other hand, Anomaly detection identifies rare deviations and extreme values in data. While OSR extends classification models to detect unknown classes, anomaly detection focuses on finding outliers without defining new classes.

2.5 EXTREME VALUE THEORY

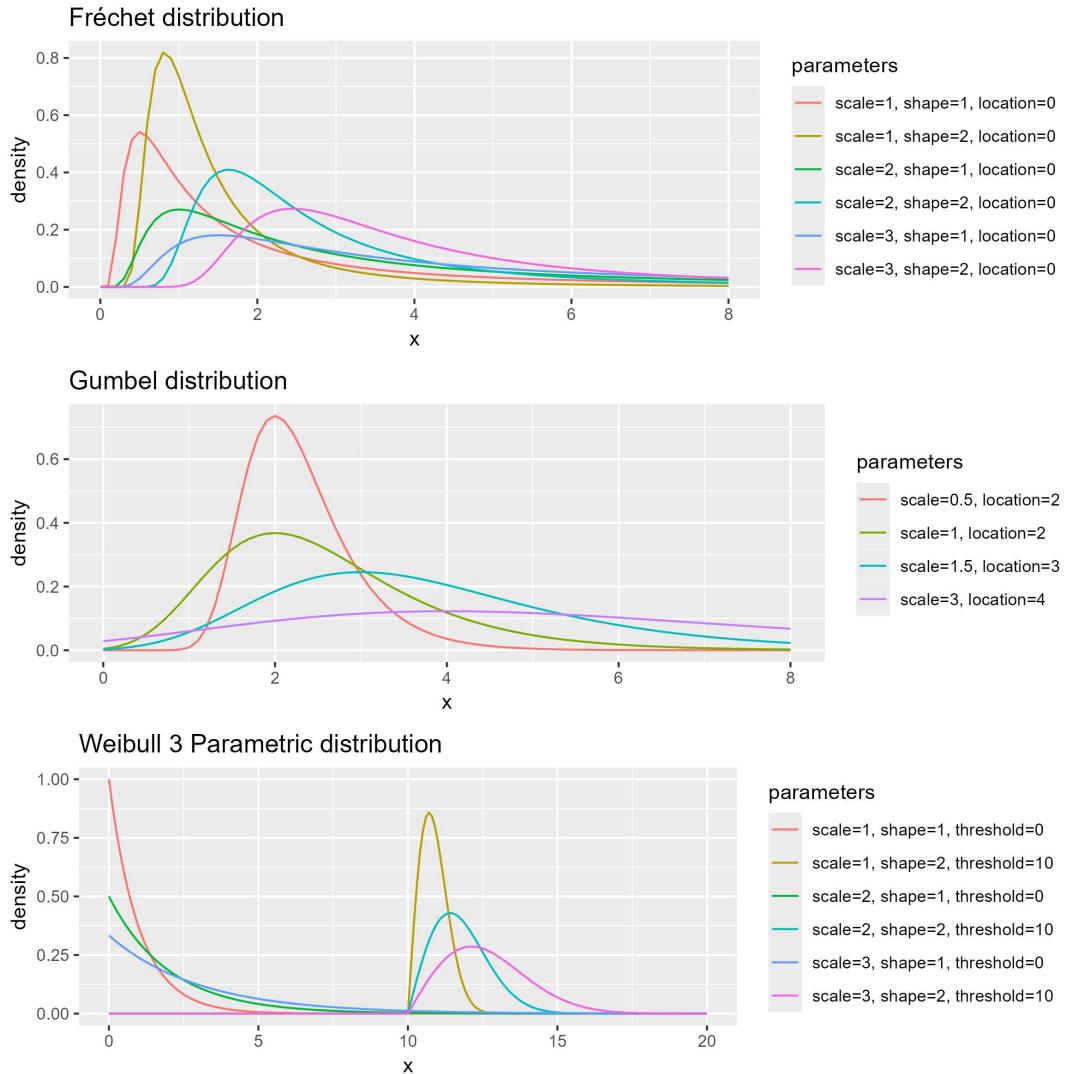
Extreme value theory (FISHER; TIPPETT, 1928) (EVT) is a statistical theory that studies extreme events or rare observations. By modeling the tail behavior of distributions, EVT makes it possible to perform risk assessment and prediction of observations beyond the range of sampled data, making it a valuable tool in several areas of research. EVT describes the possible distributions for maximum values for random variables. Let X_1, X_2, \dots, X_n be a sample of independent and identically distributed random variables. Let $M_n = \max(X_1, X_2, \dots, X_n)$ be the maximum value in the sample. If there is a sequence of pairs of real numbers (a_n, b_n) , $a_n > 0, b_n \in \mathbb{R}$ such that the limit

$$\lim_{z \rightarrow \infty} P\left(\frac{M_n - b_n}{a_n} < z\right) = F(z)$$

converges to a non-degenerate distribution function, then F belongs to the Gumbel, Fréchet or the Weibull family (RUDD *et al.*, 2018a). One of the key methods in EVT is the generalized extreme value (GEV) distribution, which serves as a fundamental model for extreme values. The GEV distribution combines the Gumbel, Fréchet, and negative Weibull distributions into a single framework, allowing for flexible modeling of extreme events across different scenarios. Figure 7 shows examples of probability density functions of the three EVT distributions. One can see that they are a set of flexible functions well suited to model extreme values.

In the specific context of EVT applied to neural network activations, (BENDALE; BOULT, 2016) states that these activations adhere to a Weibull distribution. To verify this assertion, we conducted an experimental investigation using data sampled from various distributions, including Gamma, Normal, Poisson, Weibull, and tri parametric Weibull. Figure 8 showcases histograms of these data sets, where the vertical black line represents the 10% highest values used for fitting EVT distributions (Gumbel, Fréchet, and Weibull). Subsequently, in Figure 9,

Figure 7 – Example of the EVT distributions



we present the histogram of extreme values alongside the probability density functions of three EVT functions fitted to these extreme values. Notably, across all instances, it is evident that the three-parameter Weibull distribution consistently provides the most accurate fits for extreme values. This phenomenon can be attributed to the flexibility and predictive power gained by the threshold parameter of the Weibull distribution. These observations attest the efficacy of the Weibull distribution in modeling extreme values when applying EVT to neural network activations.

Figure 8 – Samples from different distributions. The black line marks the extreme values

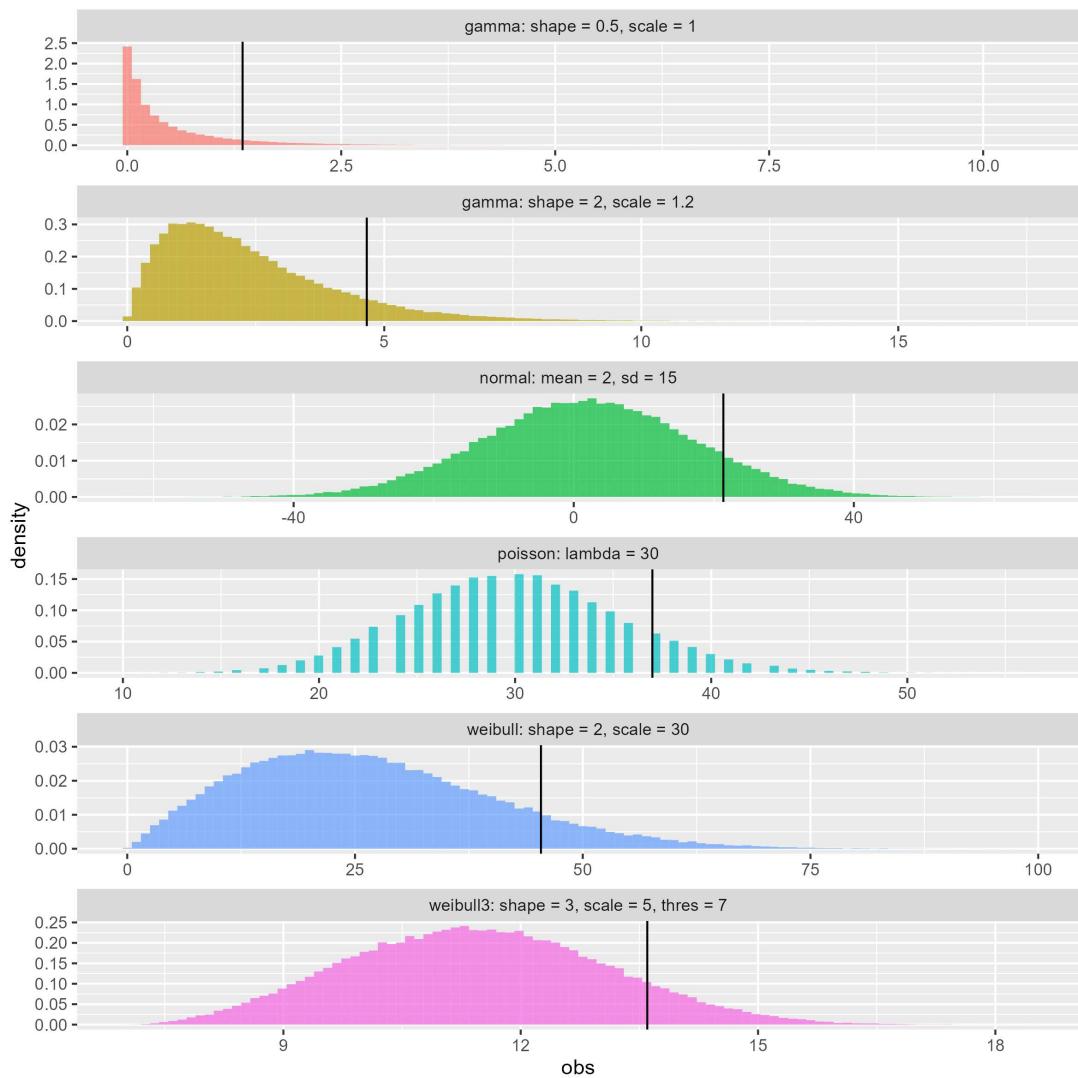
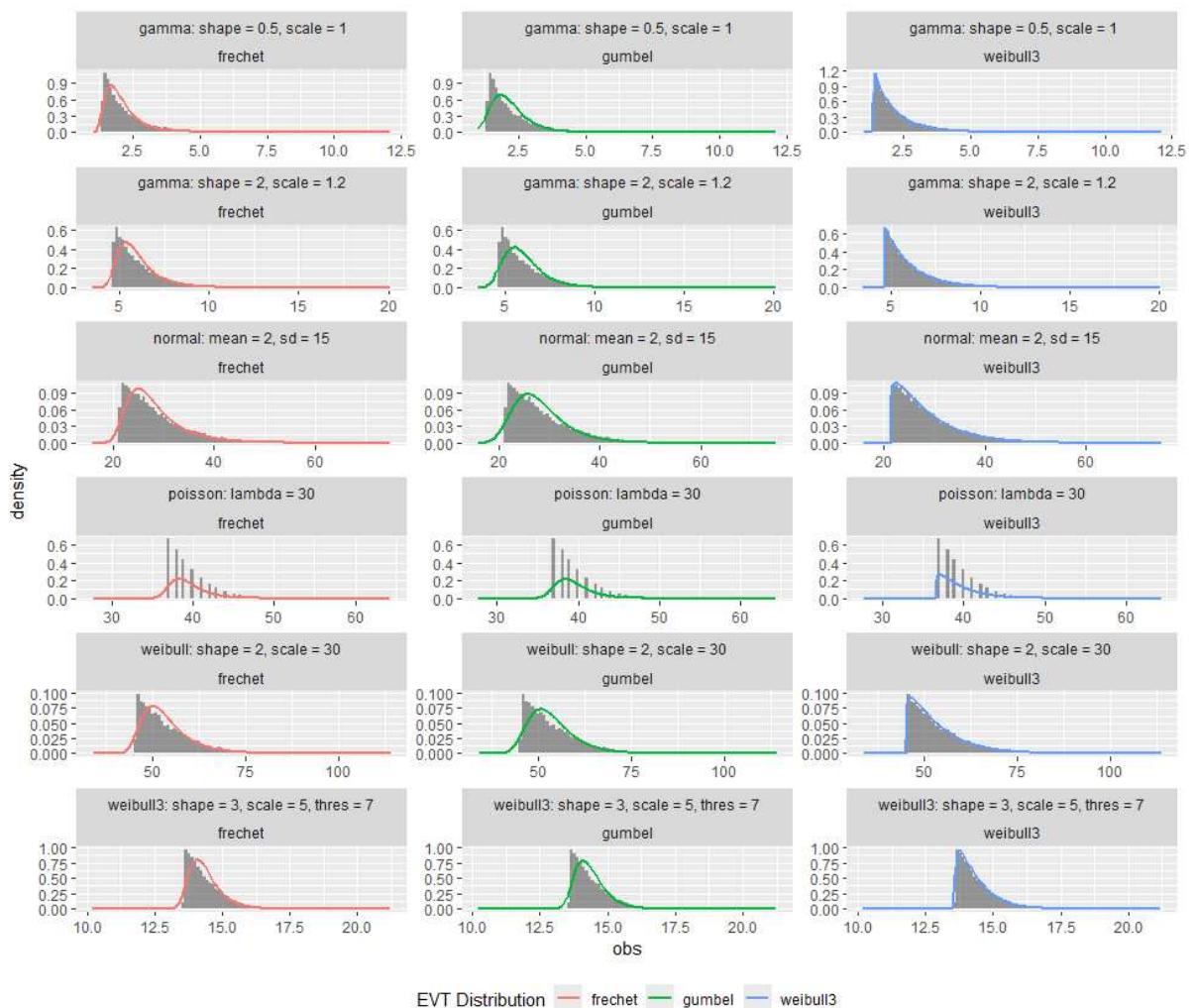


Figure 9 – EVT distributions fitted to extreme values of the samples



2.6 OPENMAX LAYER

2.6.1 Fundamentals

OpenMax (BENDALE; BOULT, 2016) marks the inception of deep open-set classification that did not rely on background samples. Subsequently, there have been a few reports of deep open-set classifiers. While OpenMax primarily focuses on recognizing adversarial inputs, it also supports the rejection of fooling and unknown images. (ROZSA *et al.*, 2017) conducted a comparison between deep neural networks (DNNs) using the conventional Softmax layer and OpenMax to evaluate their robustness against adversarial examples. OpenMax exhibits greater resilience to adversarial examples compared to Softmax and performs better than networks with thresholding SoftMax. OpenMax utilizes the Extreme Value Theory (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, described in algorithm 1, involves calculating the activation vector for each training instance, denoted as $V(x) = v_1(x), \dots, v_M$ for each class, $y = 1, \dots, M$. Additionally, a Mean Activation Vector (MAV) is computed for each class separately, considering only correctly classified training examples using the Nearest Class Mean (NCM) concept (MENSINK *et al.*, 2013). Subsequently, a Weibull distribution is fitted to each class, on the η largest distances between the MAV for the class and positive training instances that were correctly classified. Finally, the parameters of the Weibull distribution for each class are estimated. Let $\rho_j = (\tau_j, \lambda_j, \kappa_j)$ denote the location, shape, and scale parameters of the Weibull distribution for the estimated Extreme Value Theory (EVT) Meta-Recognition model for class j . Weights for the α largest activation classes are computed and used to scale the Weibull probability as the following:

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

where $s(i) = \text{argsort}(v_j)_i$ and F_W is the Weibull CDF.

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

$$\hat{V}(x) = V(x) \odot \omega(x) \quad (2)$$

$$\hat{v}_0(x) = \sum_i v_i(x)(1 - \omega_i(x)) \quad (3)$$

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

This process generates a vector that has probabilities associated with the known classes, alongside the probability of being categorized as unknown. Consequently, this approach eliminates the necessity for implementing thresholding techniques.

Algorithm 1 – OpenMax probability estimation with rejection of unknown or uncertain inputs.

Require: Activation vector for $V(x) = v_1(x), \dots, v_N(x)$

Require: Means μ_j and parameters $\rho_j = (\tau_j, \lambda_j, \kappa_j)$

Require: α , the number of “top” classes to revise

1: Let $s(i) = \text{argsort}(v_j(x))$; Let $\omega_j = 1$

2: **for** $i = 1, \dots, \alpha$ **do**

3:

$$\omega_{s(i)}(x) = 1 - \frac{\alpha - i}{\alpha} e^{\left(-\frac{\|x - \tau_{(i)}\|}{\lambda_{(i)}}\right)^{\kappa_{(i)}}}$$

4: **end for**

5: Revise activation vector $\hat{V}(x) = V(x) \odot \omega(x)$

6: Define

$$\hat{v}_0(x) = \sum_i v_i(x)(1 - \omega_i(x))$$

7: Define

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

8: Let $y^* = \text{argmax}_j P(y = j|x)$

9: Reject input if $y^* == 0$ or $P(y = y^*|x) < \epsilon$

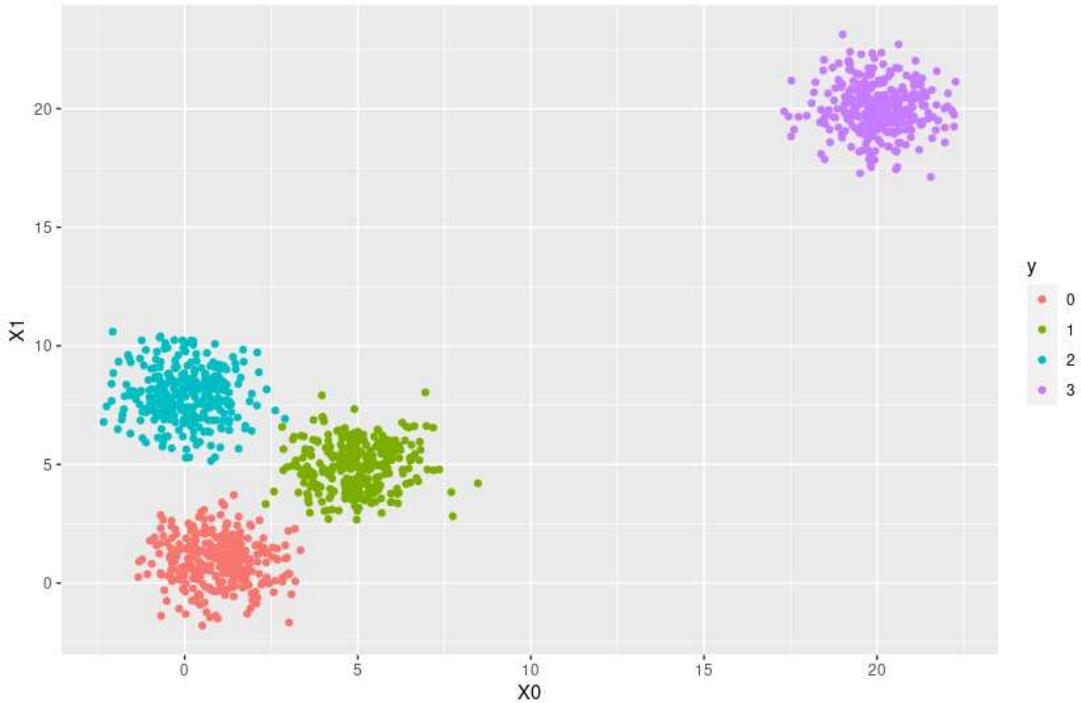
2.6.2 Prediction Surface

The OpenMax methodology operates by delineating an acceptance region for known classes and estimating the probability of unknown. To illustrate, we can apply this technique to a simulated scenario. In Figure 10 we present simulated data that contains four distinct classes, with Class 3 designated to represent unknown data. These classes are generated from data such that both x_0 and x_1 conform to a normal distribution:

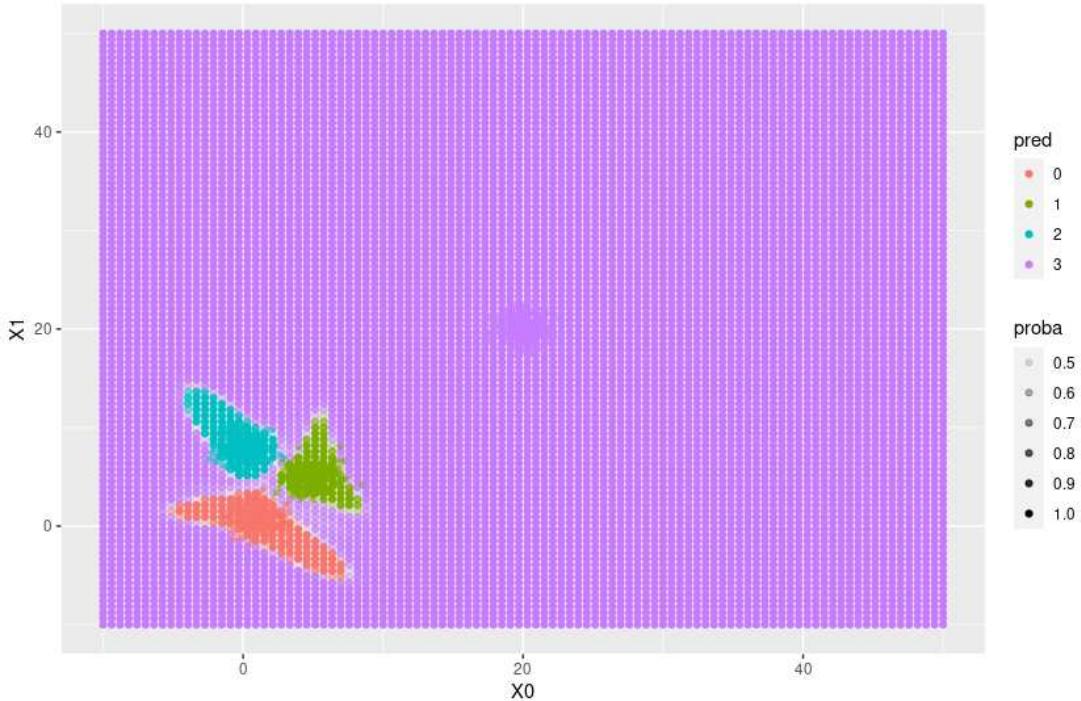
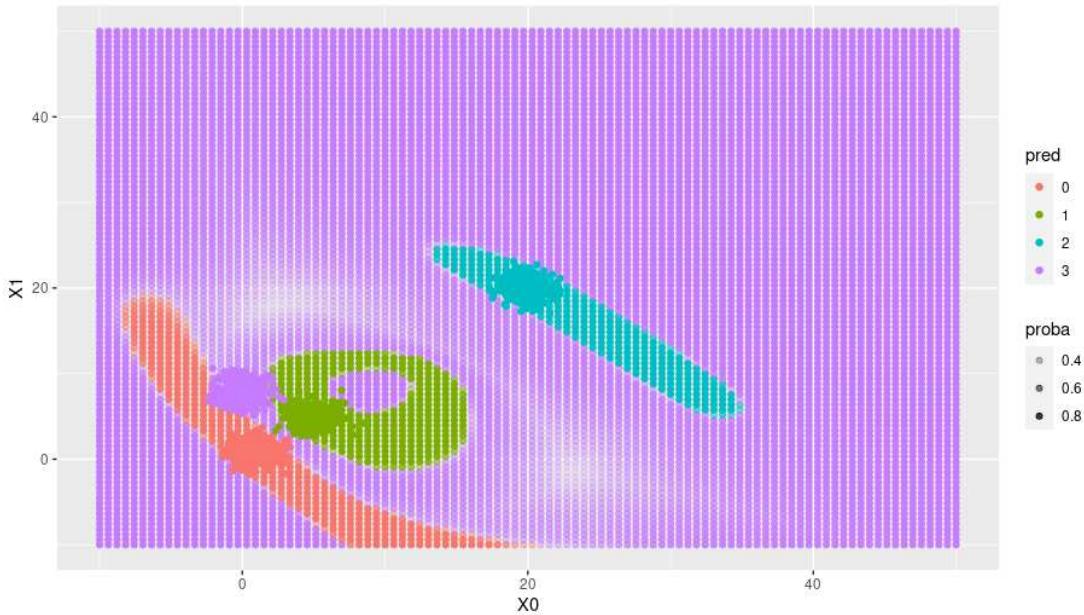
$$(x_0, x_1)_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

where $j \in (0, 1, 2, 3)$ is the class index. Here, $\mathcal{N}(\mu, \Sigma)$ represents a bivariate normal distribution with mean vector μ and covariance matrix $\Sigma = I_2$, the order 2 identity matrix. Upon fitting the model and incorporating the OpenMax layer, the resulting prediction surface, depicted in Figure 11, shows a grid that extends beyond both the simulation space as well as the sample data. The analysis shows that OpenMax has delineated acceptance regions surrounding the observed data points, while classifying any points lying outside this boundary as unknown.

Figure 10 – Simulated data



In a second experiment, we strategically positioned the unknown class closer to two known classes, while deliberately putting the remaining class at a distance from these clusters. Figure 12 evidentiates that OpenMax faced greater challenges in delineating well-defined regions around the classes in this scenario. However, despite this increased difficulty, the quality of the fit remains satisfactory.

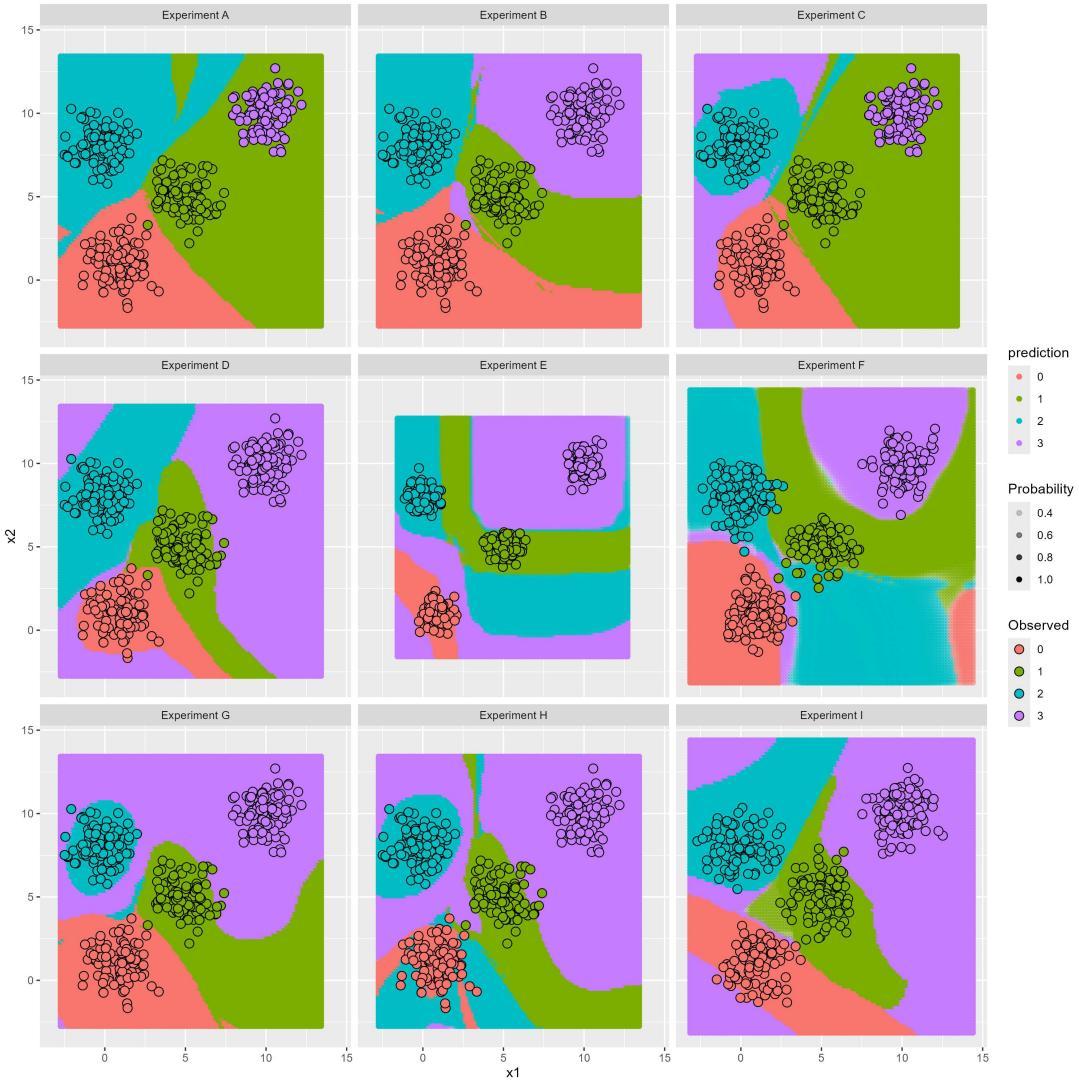
Figure 11 – Well Behaved Prediction Surface**Figure 12 – Not Well Behaved Prediction Surface**

2.6.3 Limitations

Ensuring consistent and reliable prediction surfaces with OpenMax is a challenging task due to its instability. To empirically demonstrate this phenomenon, we conducted an experiment where we exclusively manipulated the seed parameter while keeping all other hyperparameters constant. This experiment is presented in Figure 13. We see a considerable variability observed in the resulting prediction surfaces, emphasizing the sensitivity of OpenMax to minor changes

in model initialization. Such findings emphasize the necessity for cautious interpretation and robustness testing when utilizing OpenMax in open-set predictive modeling as well as studies to add robustness to models dependent on it.

Figure 13 – Prediction Surfaces for Different Trials



OpenMax does not improve the feature representation to facilitate more effective unknown detection. The class instances are not projected around the MAVs, and attempts at feature engineering have not yielded improvements in this context. Additionally, as the testing distance function is not applied during the training phase, there is no guarantee that it is the appropriate distance function for the given space. A recurrent challenge is the inherent trade-off where models excel either in detecting unknowns or in classification of knowns, but seldom in both simultaneously. Also establishing a deterministic rule for effectively combining a robust classification model with a proficient detector has proven to be a more challenging task than initially anticipated.

2.7 BLENDINGS

”Stacking” (WOLPERT, 1992), also known as ”blending” are methods designed to boost predictive accuracy by combining the predictions of multiple machine learning models. These predictions are combined and used as features for a secondary model to effectively output the final prediction. Historically practitioners in the field of machine learning have found success with blending, as they have the potential to synergistically improve prediction accuracy beyond what any single model achieves alone. For example, let’s consider two predictive models for diagnosing a rare disease: Model A and Model B. Model A may demonstrate better results when applied to patients with a well-established medical history and an extensive set of diagnostic tests. However, in scenarios where patients have limited medical data or ambiguous symptoms, Model B might prove to be more effective in providing accurate diagnoses. The secondary model learns to effectively combine the predictions from each individual model, assigning appropriate importance to each, in order to produce the final prediction. Unlike traditional ensemble techniques like bagging or boosting, which aggregate predictions through voting or weighted averaging, blending uses a meta-model to optimize the combination of model predictions. This way blendings can capture more complex relationships between the outputs of models, potentially improving performance in cases where models complement each other’s strengths and weaknesses.

2.7.1 XGBoost

Gradient Boosting Machines are effective machine learning techniques, with XGBoost (CHEN; GUESTRIN, 2016) being an implementation widely used in the industry and machine learning competitions. XGBoost combines weak learners, models slightly better than random, into a strong learner through an iterative process, enhancing predictive accuracy. It incorporates randomization techniques such as random subsamples for training individual trees and column subsampling at tree and node levels to reduce overfitting and increase training speed.

Additionally, XGBoost focuses on building less complex trees, which mitigates overfitting and reduces computational costs by reducing processing load and storage requirements. To further improve training efficiency, XGBoost employs methods to reduce the computational complexity of finding the best split, which is typically the most time-consuming task in decision tree algorithms. It uses a compressed column-based structure that makes it necessary to sort the

data only once and facilitates parallel processing of split evaluations. Instead of scanning all possible splits, XGBoost tests a subset of candidate splits based on percentiles, computing their gain using aggregated statistics. This approach significantly lowers computational demands while maintaining high model accuracy, making XGBoost a powerful model that can be implemented even in low-performance computers.

3 RELATED WORK

This chapter presents an overview of previous Open Set Recognition (OSR) research applied to Time Series Classification (TSC). It begins with a detailed discussion on TSC, addressing the unique challenges of time-dependent data and introducing several methods and algorithms in this field. It then transitions to OSR, exploring techniques for detecting samples from unknown classes and classifying known ones. Following this, the chapter shows research focusing on integrating OSR with TSC in various fields. The chapter provides a background demonstrating some novel OSR techniques that can enhance TSC by effectively handling and recognizing unknown instances.

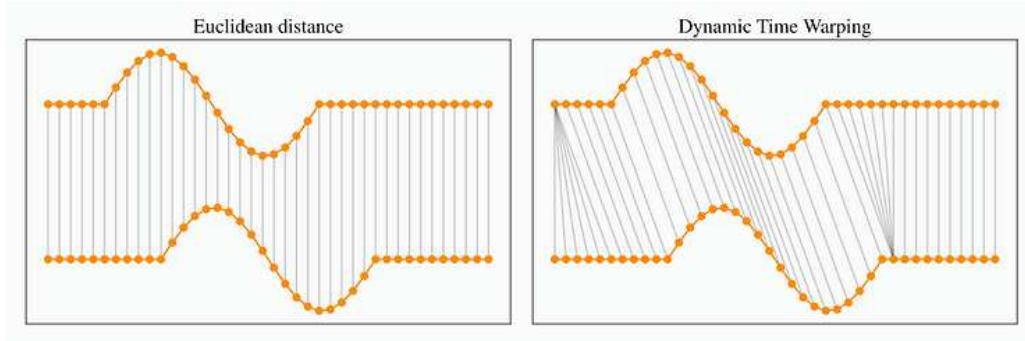
3.1 TIME SERIES CLASSIFICATION

When an event has its values dependent on time we can study it as a time series. Stock prices, temperatures and health markers are some of the everyday life examples of them. TSC involves build a model that assigns a label to an instance of a TS (W.K., 2022). If the instance is consisted of just one metric along the time it is defined as a univariate TS and, conversely, it is considered a multivariate TS when it has more than one metric along the time. In TSC the observed metrics are taken as the inputs of the ML model, which outputs a vector of probabilities that the instance belongs to the corresponding class.

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 having different lengths, as resampling time series might decrease data representation of the features. Besides, Euclidean distance compares the values of both time series independently, disregarding the fact that the values are correlated. (SAKOE; CHIBA, 1978) provides the DTW as an alternative measure to address these limitations of Euclidian distance. In essence, DTW allows you to stretch or compress the time axes of the Time Series to find the optimal alignment, enabling a more accurate comparison. Figure 14 presents an example of how DTW differs from Euclidian distance. Despite the advantages over the Euclidean distance, DTW has relevant limitations, such as high algorithmic complexity, which makes it computationally expensive for long Time Series; substantial time warps, which might change the Time Series more than desirable; and

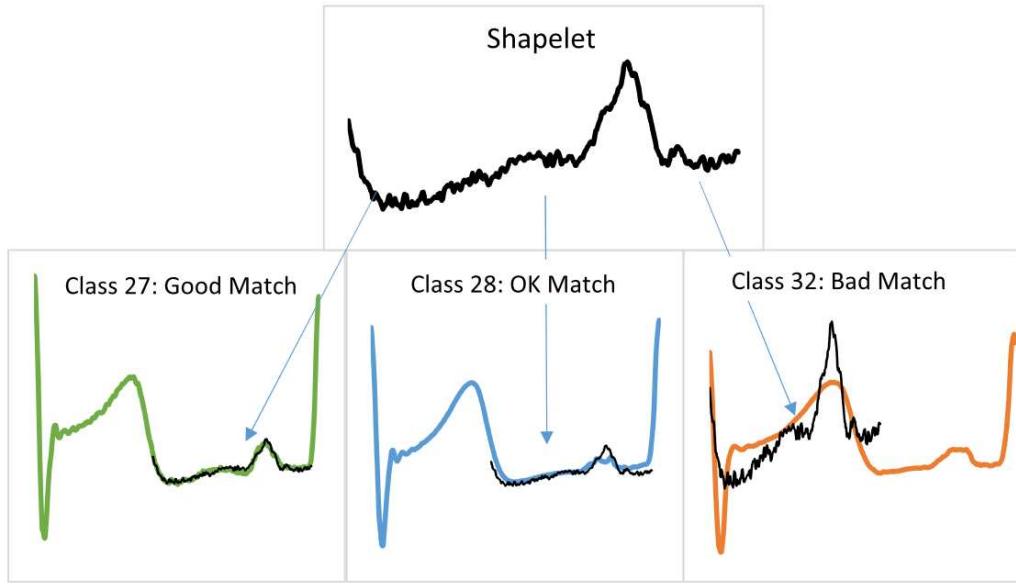
its non-differentiability, which is a challenge for integration and research with machine learning algorithms that rely on gradient-based optimization (FAOUZI, 2022). DTW is frequently employed alongside algorithms that rely on similarity metrics, such as SVM and KNN, as an alternative to Euclidean distance.

Figure 14 – Euclidian Distance and Dynamic Time Wrapping



Phase-independent shapelets (YE; KEOGH, 2011) are relevant in scenarios where patterns within a time series define a class, regardless of their specific location (BAGNALL *et al.*, 2017). For instance, abnormal electrocardiogram (ECG) measurements may exhibit specific patterns that occur sporadically during the measurement. Shapelets representing subseries capturing such characteristics enable the identification of phase-independent localized similarities within the same class (see figure 15). Methods based on the full Time Series often perform poorly, as informative anomalies may occur in varied locations, while interval-based methods offer improvement, but still rely on discriminatory features appearing approximately in the same location. The original shapelet algorithm identifies shapelets through exhaustive enumeration, employing the best shapelet as the splitting criterion in a decision tree. The algorithm, however, is computationally expensive and when applied to said decision trees does not yield remarkably accurate classifiers. Significant efforts to reduce computational cost of shapelet discovery have been applied using techniques that involve discretising and approximating the shapelets using symbolic aggregate approximations (LIN *et al.*, 2007), finding the top k shapelets on a single run (HILLS *et al.*, 2014) and the adoption heuristic gradient descent shapelet search algorithm rather than exhaustive enumeration (GRABOCKA *et al.*, 2014).

Figure 15 – Example of Shapelets applied to ECG data



3.2 OPEN SET RECOGNITION

OSR systems can be divided into two categories. The first one is more concerned with detecting instances from unknown classes. This approach is explored in research such as Kernel Null Space Methods for Novelty Detection (BODESHEIM *et al.*, 2013) and the “1-vs-Set Machine” (SCHEIRER *et al.*, 2013). In the second category, OSR refers to the task of distinguishing among the known classes, in which the system can detect unknowns and appropriately label instances as one of the known classes (GE *et al.*, 2017; JAIN *et al.*, 2014; BENDALE; BOULT, 2016). Accurately classifying known instances while effectively detecting unknown ones is a challenge faced by OSR systems.

3.2.1 Extreme Value Machine

The Extreme Value Machine (EVM), proposed by (RUDD *et al.*, 2018b) is derived from statistical Extreme Value Theory (EVT). In EVM each known class is represented by a set of extreme vectors, each associated with a Probability of Sample Inclusion (PSI). Let x_1, x_2, \dots, x_n be training samples and $y_i \in C$ be the class label for x_i . Given x_j , where $\forall j, y_j \neq y_i$, the closest instance of another class, The margin estimates for the τ closest points are calculated as $m_{ij} = \frac{\|x_i - x_j\|}{2}$. The distributional form of the margin distances is estimated utilizing EVT, and results in $\Psi_i(x) = 1 - F_W(x, \theta_i)$, where F_W is the Weibull cumulative distribution function (CDF). Thus each training instance x_i has its own model Ψ_i , and each class C_l is associated with

a set of models Ψ_i where $y_i = C_l$. The probability that an observation x_j is associated with class C_l is $\hat{P}(y_j = C_l | x_j) = \arg \max_{i:y_i=C_l} \Psi_i(x_j)$. Let δ be the threshold probability that defines the boundary of sample inclusion, the decision function in OSR is given by:

$$\hat{y}_j = \begin{cases} \arg \max_{l \in \{1, \dots, M\}} \hat{P}(C_l | x_j), & \text{if } \hat{P}(C_l | x_j) \geq \delta \\ "unknown", & \text{otherwise} \end{cases} \quad (5)$$

where M is the number of known classes. If an observation x_j does not fall within the inclusion boundary of any x_i defined by the threshold δ then it is detected as unknown. To illustrate this concept, we use the following example: we randomly sample 15 observations from four classes $C_l, l \in 0, 1, 2, 3$, that are normally distributed with parameter vector θ_i , $(x_{0i}, x_{1i}) \sim \mathcal{N}(\theta_i)$, class 3 being used as the unknown. Each observation x_i is then fitted with a Weibull distribution based on its extreme values, resulting in a unique set of parameters θ_i . The circle around the i-th observation is determined by the 0.95 quantile of the Weibull distribution associated with the parameters of the given observation θ_i . Figure 16 has a visual representation of this experiment. It becomes apparent that observations situated farther away from those of other classes tend to have larger extreme values, consequently resulting in Weibull parameters that define a bigger circle. Conversely, observations closer to those of other classes have smaller extreme vectors, leading to the definition of a smaller circle. This visualization effectively highlights the relationship between the spatial distribution of observations and the size of the acceptance region defined by Weibull parameters. In Figure 17, we see the prediction surface from our experiment. The Extreme Value Machine has drawn areas where it can predict the known classes and identify the unknown ones, using the rule in 5. Keeping all Ψ models results in more computationally expensive OSR systems. To address this issue, EVM also includes a model reduction technique, that strategically discards Ψ models that do not significantly compromise classification performance.

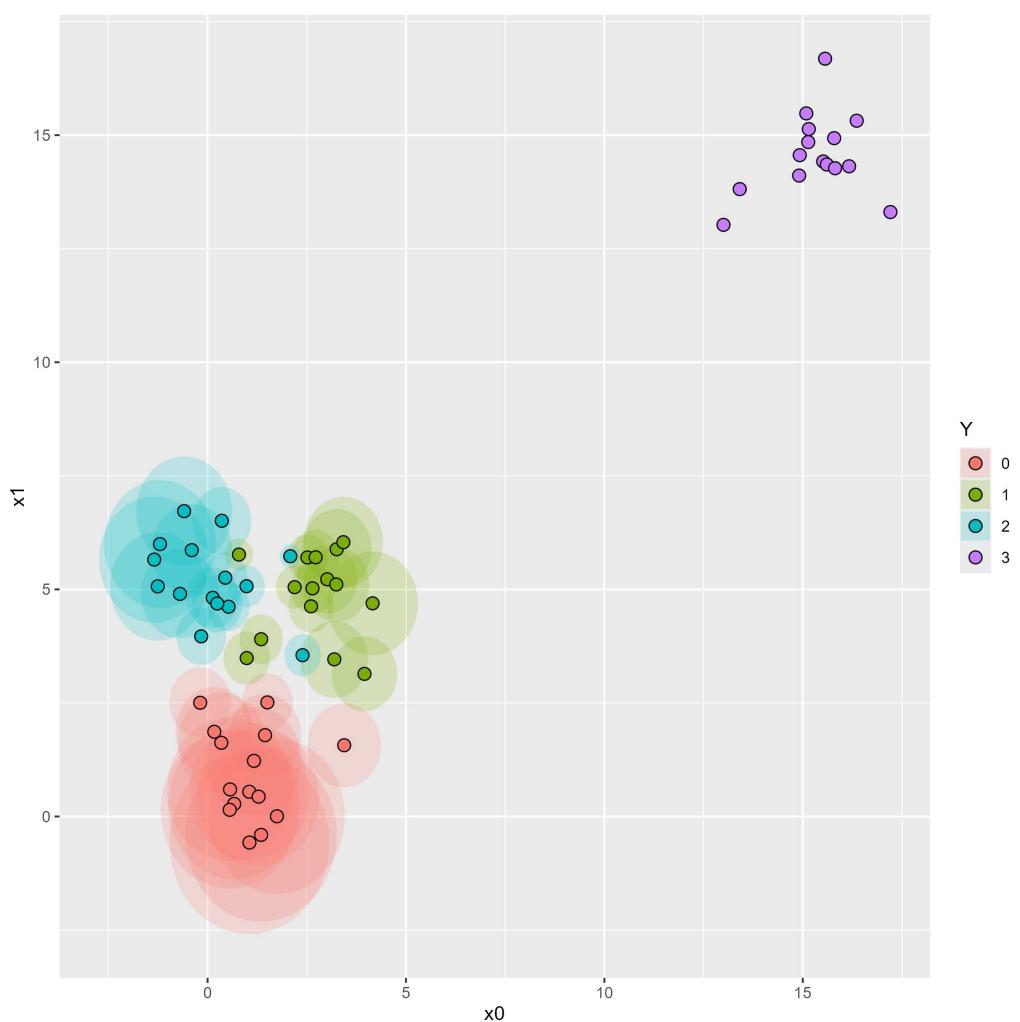
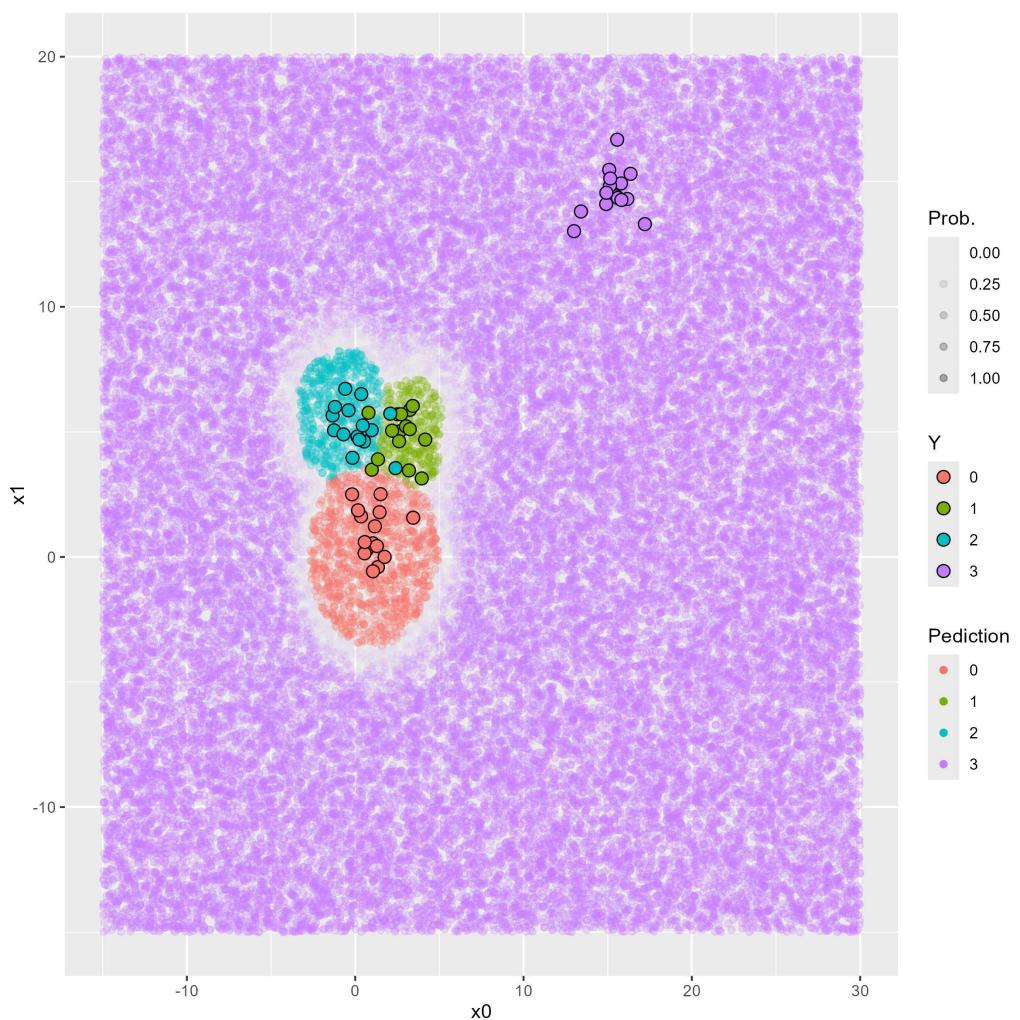
Figure 16 – EVM 0.95 quantiles

Figure 17 – EVM Prediction Grid

Scheirer *et al.* (2013) proposed a solution to OSR as a risk-minimizing constrained functional optimization problem by developing an extension of the 1-class SVM, the 1-vs-Set Machine, which consists of two parallel hyperplanes tailored to improve both generalization and specialization. The main hyperplane is a base SVM which defines half-spaces and aims at maximizing the margin so it just touches the extremes of the positive instances. The second hyperplane is added in such a way as to minimize the positive labeled region bounded between two planes and handle risk of encountering unknowns. This method delineates distinct regions corresponding to known classes for each binary SVM, however, it does not have a systematic approach for measuring distances between instances, nor specify the space where distances should be measured. Consequently, it cannot define decision boundaries that each class belongs to. In essence, the method lacks a technique for setting the decision space occupied by each known class, leaving the model vulnerable to misclassification when encountering novel or previously unseen instances.

In the field of face recognition, Vareto and Schwartz (2020) proposed an approach that integrates two key techniques: The Affinity Propagation Clustering (APC) algorithm and an ensemble of Partial Least Squares (PLS) models. Affinity Propagation Clustering is employed to group the subjects of the face gallery dataset. When a sample q is presented, the APC algorithm identifies the k most similar clusters and creates a training subset. Subjects in this training subset are randomly partitioned into d positive and negative subsets. The d partitions are used to train d PLS models which constitute the ensemble. At the testing phase the sample q is presented to the ensemble along with positive samples in the d training subsets. A histogram is built using the output of the ensemble and thresholding decides whether q belongs to the face gallery. This method is a scalable approach for open-set face identification that can handle galleries having hundreds or thousands of subjects. The algorithm combines a clustering technique with an ensemble of regression models to effectively identify individuals from the gallery dataset that exhibit significant similarity to the testing image. It also categorizes them as unknown when the degree of similarity is insufficient to confidently match an individual in the gallery. Addressing the practical challenges of sheep face recognition, like changes in the flock and variations in facial features due to different views, Li *et al.* (2024) proposed the Li-SheepFaceNet and Li-ArcFace loss function which employs deep learning techniques and OSR to improve effectiveness in sheep management. The method employs a Seesaw block to construct the lightweight SheepFaceNet model, improving performance and reducing computational resources at the same time. The

method achieved high open-set recognition accuracy in both self-built and publicly available datasets, demonstrating its relevance in intelligent livestock farming practices.

3.3 OPEN SET RECOGNITION ON TIME SERIES CLASSIFICATION

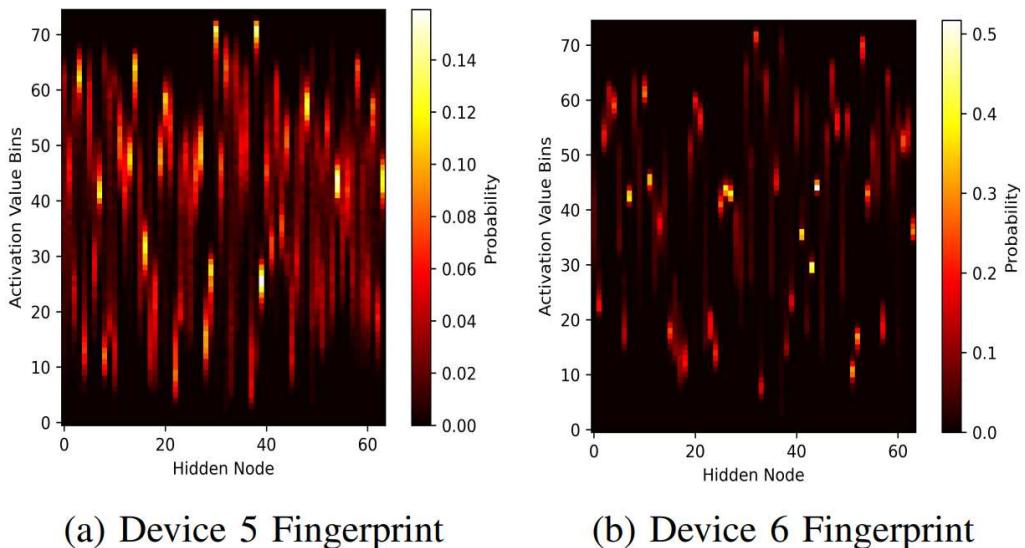
Open-set recognition techniques have a well-established presence, although their application to time series classification has not been as extensive as observed in other domains, such as image classification. However, the relative scarcity of contributions in this area is counterbalanced by the quality of the research, where researchers came up with inventive and innovative methodologies to address this challenge. Open Set InceptionTime (AKAR *et al.*, 2022) integrates the use of barycenters into the model. The model computes their DTW distance and cross-correlation with class-specific barycenters when presented with novel samples. 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 is also used as the baseline for other works in the same domain and will be employed as a benchmark in the current study. The MEROs (Multi-Feature Extraction and Reconstruction Learning for Open-Set Recognition) (OH; KIM, 2022) method is introduced to address the preservation of unique features associated with unknown classes through multi-feature extraction applied to TS data. It also uses OpenMax for unknown detection, but in contrast to previous 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. The experiments conducted with various TS datasets demonstrate that MEROs achieves improved detection of unknown classes and maintains strong predictive performance.

In addressing the challenge of open-set recognition for underwater acoustic targets, Yang *et al.* (2022) introduces a recognition approach that integrates deep learning and template matching. The proposed method comprises three key stages: initial feature extraction through a Gated Recurrent Unit and Convolution Auto-Encoder (GRU-CAE) collaborative deep learning network, template creation, and template matching. The CAE network excels in spatial information extraction from the ship noise spectrogram, while the GRU classifier network effectively captures the temporal structure features. The deep collaborative features $V(x)$ are extracted and then used to construct feature templates, which are the means ω_j for the N underwater acoustic

target categories. Then the template matching recognition thresholds ϵ_j for each category are set. If $d(V(x_{test}), \omega_j) > \epsilon_j$, the euclidean distance between the collaborative features vector of a test sample $V(x_{test})$ and feature template ω_j is greater than the category threshold ϵ_j , for all j then the sample is detected as unknown. This methodology demonstrates the effective integration of deep learning and template matching for open-set recognition in ship radiated noise using a distance based algorithm.

HiNoVa (PUPPO *et al.*, 2023) is also a method that leverages activations of multiple hidden layers of a DNN, specifically an architecture that begins with a Convolutional Neural Network followed by a Long Short-Term Memory (CNN+LSTM) model. The HiNoVa methodology utilizes 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. Figure 18 shows an example for two devices and their respective activation histograms. This matrix histogram serves as extracted features. During testing, the algorithm computes 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.

Figure 18 – HiNoVa hidden layer activations



In Liu *et al.* (2024) researchers investigate OSC for known and unknown music genre classes by applying softmax thresholding and Openmax on two open-source datasets. The research evaluates inference quality using metrics such as accuracy, precision and recall. Ad-

ditionally, the study examines the impact of threshold levels on accuracy, provides confusion matrix analysis, and offers graphical insights through dimensionality reduction of the dataset features.

Multi-label audio classification (AC) is a common machine learning task applied to various scenarios such as urban sound data, everyday environments, and music, where instances might have more than one class. However, much of the existing research assumes a small, fixed class vocabulary, focusing on closed-set tasks that do not reflect real-world scenarios. To address this, Sridhar and Cartwright (2023) introduces the employment of OSC in the field of multi-class audio classification. By also implementing techniques such as softmax probability thresholding and Openmax across five distinct datasets, the research demonstrates promising results in identifying unknown sound events within polyphonic audio contexts. This approach effectively addresses the limitations of traditional closed-set models that operate with a fixed class vocabulary, thereby offering a more comprehensive solution for real-world audio classification challenges. Furthermore, the study outlines critical issues for future research, emphasizing the need for enhanced robustness and generalization in OSC models, as well as exploring novel methodologies to improve the detection and classification of unknown sound events.

Also in handling the challenge of open-set recognition for sound events, You *et al.* (2024) improves traditional sound event classification by integrating deep learning techniques. The method uses a compact cluster structure in the features of known classes, facilitating the recognition of unknown classes by providing ample space to locate unknown samples. This is achieved by applying center loss and supervised contrastive loss to optimize the model. The center loss minimizes intra-class distances by pulling embedded features towards the cluster center, while the contrastive loss maximizes inter-class feature dispersion. Additionally, it applies self-supervised learning for detecting unknown sound events. Experimental results on three datasets demonstrate that the proposed approach, combined with self-supervised learning, achieves performance improvements across various audio processing tasks, like genre classification, vocal style differentiation, and acoustic scene classification. The study highlights the potential of these methods to generalize across varied tasks and sets a foundation for future work on adding semantic information of unknown samples to improve detection accuracy.

4 METHODOLOGY

This chapter presents the methodology employed in our experiments to investigate open-set recognition using the blending of neural networks trained on known and unknown time series data. It begins by describing the data sets used, highlighting their diversity, relevance to real-world scenarios, and common use in OSR research. Next, it shows the neural network models used in our experiments and the hyperparameters explored to enhance model performance. The experimental setup section presents the process of training neural networks and blending their predictions, emphasizing the steps and approach to handling known and unknown time series data effectively. Next, it presents the evaluation metrics used to test model performance, focusing on accuracy, precision, recall, and F1-score. Finally, it introduces the concept of openness in open set recognition, a metric that quantifies the gap between known and unknown classes and provides the openness values for our experiments. This structure provides a comprehensive understanding of the methodology, covering data preparation, model training, evaluation, and the challenges of open-set recognition in time series classification.

4.1 DATA

To conduct this study, we used a diverse and comprehensive set of TS data sets provided by the UEA archive (BAGNALL *et al.*, 2018). This collection of data sets is usually the standard for research on time series as it represents several real-world scenarios. Analyzing the TS metadata in table 1, we see that the TS has a 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 limited 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.

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

4.2 MODELS

Our experiment leverages state-of-the-art DNNs by applying transfer learning to nine architectures. We constructed a set of DNNs that has both widely adopted architectures and those less commonly used while also incorporating models that exhibit strong performance in the domain of TSC, as well as those that are not specifically tailored for such tasks: FCN (SHELHAMER *et al.*, 2016), InceptionTime (FAWAZ *et al.*, 2020), LSTM (HOCHREITER; SCHMIDHUBER, 1997), LSTM-FCN (KARIM *et al.*, 2018), mWDN (WANG *et al.*, 2018), OmniScaleCNN (TANG *et al.*, 2022), ResCNN (WEN *et al.*, 2019), ResNet (HE *et al.*, 2015), XceptionTime (RAHIMIAN *et al.*, 2019). The selection of models was made so that we have a diverse set of neural networks and take advantage of their strengths. After varying hyper-

parameters for some neural networks, we had up to 25 models for each time series. In our blending approach, the final model employed was XGBoost (CHEN; GUESTRIN, 2016) due to the robustness, scalability, and ease of use. The motivations behind using blending of DNNs stems from observing that some of these DNNs demonstrate proficiency in classifying known instances, but struggle with detecting unknown instances. Conversely, other DNNs exhibit superior performance in detecting tasks but may lack robust classification capabilities. Additionally, creating a deterministic rule that effectively combines a proficient classifiers with a proficient detector proves to be more complex than initially anticipated.

4.3 EXPERIMENTAL SETUP

We conducted the experiment ensuring that it is representative of real world scenarios applying the following steps for each time series.

4.3.1 Training Neural Networks

The data sets utilized in this study were obtained from the source already pre-segmented into training and testing subsets with varying proportions for each split. We further partitioned the training set into training and validation sets, with proportions 70% and 30% respectively, to facilitate the training. Following this partitioning process, the OpenMax layer was trained on the neural network architecture. Thus, for each model we have the fitted neural network and the fitted OpenMax layer. The neural networks are implemented in Python using the pytorch and trained on a GeForce GTX 1080 Ti.

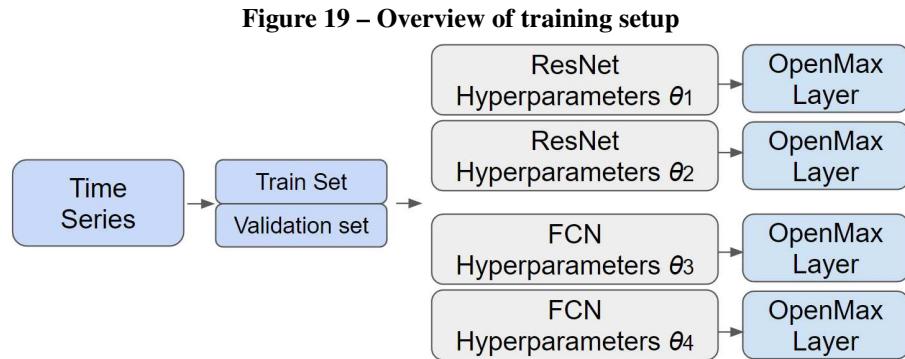
4.3.1.1 Hyperparamenters

In exploring various neural network architectures for our study, we systematically varied the hyperparameters to increase model variety. The chosen architectures have conventional and advanced DNNs. Below is an overview of hyperparamenters search space:

- **FCN, ResNet, ResCNN, InceptionTime and OmniScaleCNN:** default settings;
- **LSTM_FCN:** Shuffle (True or False);
- **LSTM:** Number of layers (1, 2, or 3), Bidirectionality (False or True);

- **mWDN**: levels (2, 4, or 6); **XceptionTime**: Default setting, Number of filters (8 or 24), Adaptive sizes (10, 50, 90, 150, 200 or 300).

Figure 19 presents a visual representation of the training setup.



4.3.2 Training Blendings

To evaluate the performance of our paradigm in the presence of both known and unknown TS, each time series was used on multiple neural networks, and their respective OpenMax layers. This set of diverse models constituted the basis for our blending approach. The machine learning engineering applied is the following

- For each model, the Neural Network activations and the OpenMax layer predictions were extracted as features (figure 20).
- These features were concatenated across all models, making a novel feature matrix for the known time series.
- We also extracted these features for the other Time Series that served as representatives of the unknowns and known-unknowns. Notably, these unknown time series instances inherently deviated in terms of both sequence length and number of dimensions of the known time series. To address this challenge, we implemented a resampling technique to ensure compatibility and enable consistent model evaluation across both known and unknown classes.
- The data is split into train, test and open-set splits. The open-set split of our dataset has exclusively data from the category of unknown Time Series. In contrast, the training and testing splits have a combined set of both known and known-unknowns TS data, as

represented in figure 21. With access to a sample of unknown TS in the training stage, the model can be optimized for classification while also learning to detect unknowns.

Figure 20 – Blending Feature Extractor

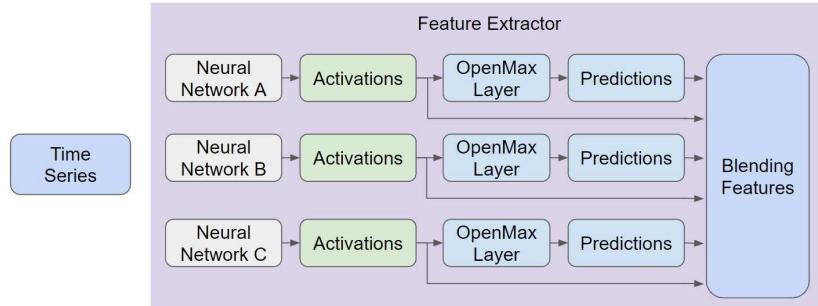
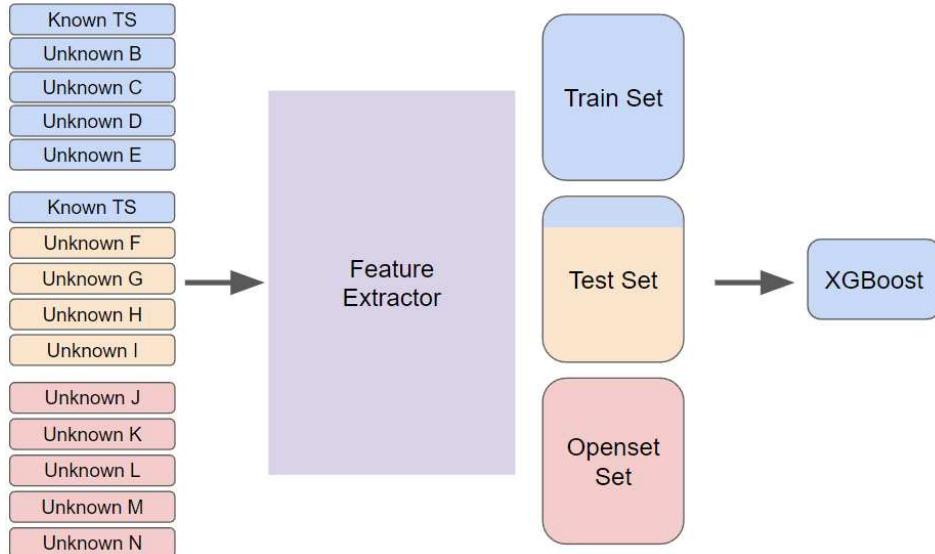


Figure 21 – Data Set Split Overview



It is important to underline that the three distinct data subsets are intentionally built so that they do not share instances of unknown TS. We adopted this separation strategy to mitigate the risk of overfitting to the patterns in the known-unknown class.

4.3.3 Performance Measures

To assess the performance of our open-set recognition methodology, we employ standard evaluation metrics: Accuracy, Precision, Recall, F1-score. Let

- **True Positive (TP):** The number of correctly predicted positive instances.
- **True Negative (TN):** The number of correctly predicted negative instances.

- **False Positive (FP):** The number of incorrectly predicted positive instances (Type I error).
- **False Negative (FN):** The number of incorrectly predicted negative instances (Type II error)

the evaluation metrics are defined as:

- **Accuracy:** measures the proportion of correctly identified instances among the total instances. It gives an overall performance measure of the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

- **Precision:** measures the proportion of correctly identified positive cases out of all cases that are classified as positive. It indicates the correctness of the positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

- **Recall:** measures the proportion of correctly identified positive cases out of all actual positive cases. It indicates the ability of the classifier to find all the positive samples.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

- **F1-score:** is the harmonic mean of precision and recall. It provides a balance between precision and recall. A higher F1-score indicates better performance, considering both precision and recall.

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

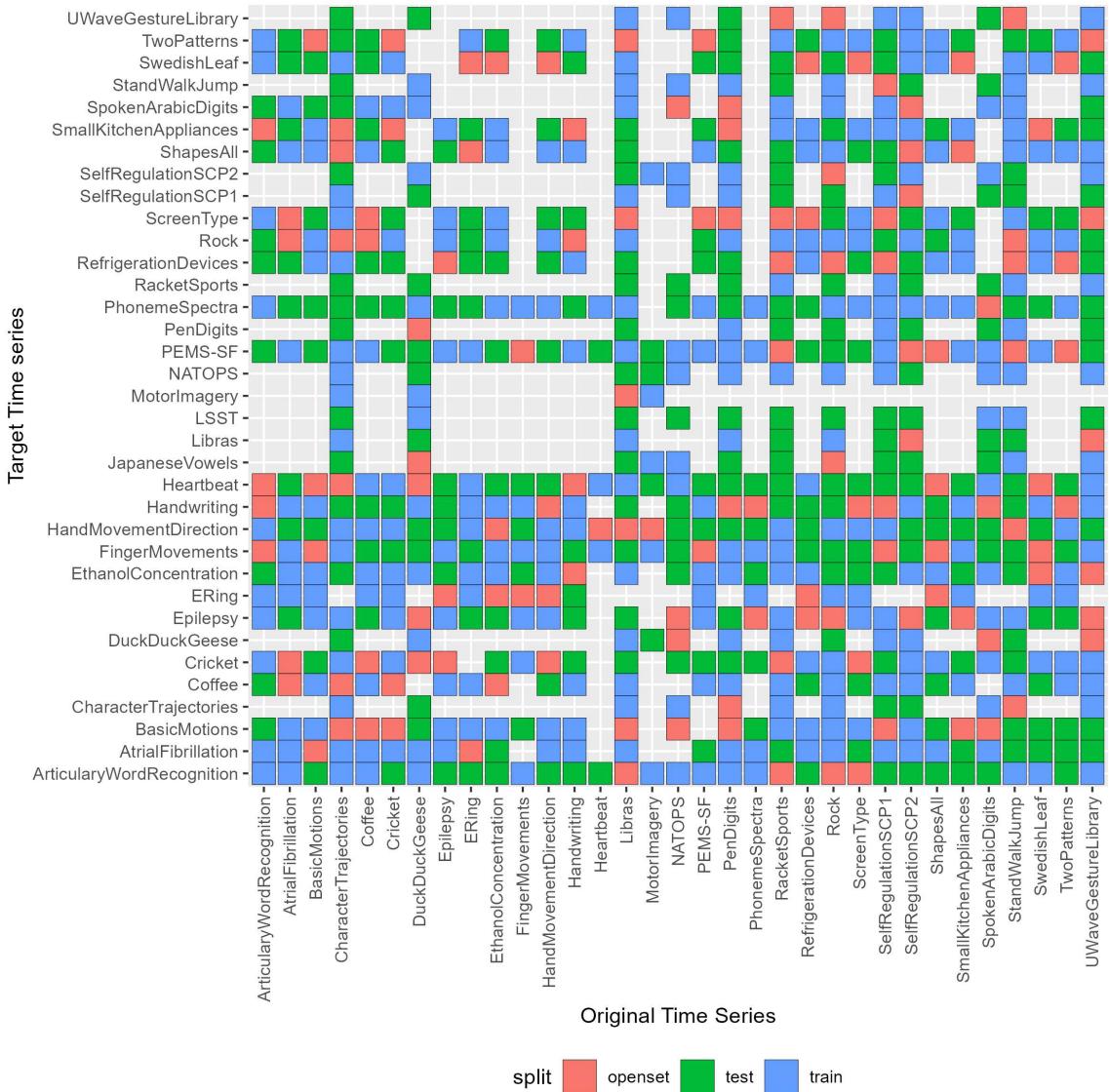
When dealing with multiple classes, it is essential to aggregate performance metrics to gain insights into the overall model performance. One common approach is to use the macro averaging strategy. Unlike other averaging methods like micro or weighted, which consider class imbalance or sample size, the macro average treats each class equally. It calculates the metric independently for each class and then computes the unweighted mean across all classes. This means that each class contributes equally to the final aggregated metric. As a result, the macro average provides a balanced assessment of the model's performance across all classes, making it particularly useful when classes have varying degrees of importance or when the dataset is imbalanced.

We particularly focus on Accuracy and F1-score as they are widely used as standard in OSR studies and are also the metrics shown on the benchmark. The quality metric employed for evaluating the quality on the split is computed as the mean of the individual quality metrics assigned to each Time Series instance contained within the respective split. This approach is deliberately adopted to mitigate any potential biases from imbalances in data set sizes between various splits, thereby ensuring that each split is assessed based on an unbiased representation of its Time Series. Figure 22 presents a comprehensive overview of the ArticularyWordRecognition time series data, showing the accuracy associated with each individual time series and its corresponding allocation across different data splits. The dashed line represents the average Accuracy of the split, which is used as the overall quality of the split withing the Time Series. Figure 23 shows the experiment coverage. On X axis we have the "Original Time Series", being the time series the models were trained to classify. On Y axis we have "Target Time series", which are the time series used as target for the models and blendings. The color represents which split the time series was assigned to. Note that there are some empty spaces due to bugs related to the ML process, but the coverage is still better than satisfactory.

Figure 22 – Accuracy of blending and Split strategy



Figure 23 – Experiment Coverage



4.3.4 Openness

In the field of machine learning and particularly in open set recognition, one important metric is "openness" (GENG *et al.*, 2021), a concept introduced to quantify the gap between known classes, on which the model is trained, and the total classes, including unknown ones, faced during testing. Let C_{TA} , C_{TR} , and C_{TE} respectively represent the set of classes to be recognized, the set of classes used in training, and the set of classes used during testing. Then, the openness of the corresponding recognition task O is defined as:

$$O = 1 - \sqrt{\frac{2 \times |C_{TR}|}{|C_{TA}| + |C_{TE}|}} \quad (10)$$

where $0 \leq O \leq 1$ and $|\cdot|$ denotes the number of classes in the corresponding set. A larger openness corresponds to more open problems, while the problem is completely closed when the openness is equal to 0. Table 2 presents the openness of the splits for the TS of both experiments, benchmark (AKAR *et al.*, 2022) and ours.

Table 2 – Openness

Original Ts	Benchmark Test	Our Train	Our Test	Our Openset
ArticularyWordRecognition	0.14	0.46	0.46	0.22
AtrialFibrillation	0.29	0.81	0.73	0.53
BasicMotions	0.33	0.75	0.75	0.35
CharacterTrajectories	0.38	0.50	0.53	0.41
Cricket	0.07	0.49	0.65	0.19
DuckDuckGeese	0.00	0.72	0.66	0.54
Epilepsy	0.65	0.55	0.78	0.52
ERing	0.11	0.48	0.55	0.51
EthanolConcentration	0.40	0.77	0.67	0.56
FingerMovements	0.00	0.81	0.55	0.58
HandMovementDirection	0.27	0.75	0.64	0.67
Handwriting	0.17	0.44	0.48	0.11
Heartbeat	0.50	0.71	0.68	0.29
Libras	0.09	0.58	0.60	0.35
MotorImagery	0.50	0.70	0.61	0.29
NATOPS	0.07	0.65	0.67	0.41
PEMS-SF	0.14	0.73	0.54	0.22
PenDigits	0.00	0.57	0.68	0.52
PhonemeSpectra	0.06	0.31	0.26	0.15
RacketSports	0.31	0.69	0.80	0.65
SelfRegulationSCP1	0.53	0.81	0.85	0.70
SelfRegulationSCP2	0.29	0.83	0.80	0.80
SpokenArabicDigits	0.00	0.53	0.58	0.54
StandWalkJump	0.45	0.81	0.78	0.66
UWaveGestureLibrary	0.48	0.72	0.65	0.44

5 RESULTS

This chapter begins with an overview of the quality metrics obtained from the experiments, including F1 score, accuracy, precision, and recall, which are crucial for evaluating the performance of the classification model. It presents detailed tables and figures illustrating the metrics across different data splits (training, test, and open-set). It highlights the model's capability to identify known classes and detect unknown ones. The chapter then shows a comparative analysis with a benchmark study, attesting the superior effectiveness and robustness of the novel approach in handling open-set scenarios. This comparison includes an analysis of the impact of the dataset openness on the quality metrics, further reinforcing the applicability and reliability of the proposed method in real-world scenarios. The experimental results and benchmark analysis demonstrate the model's capabilities and contributions to the field of OSR for TSC.

5.1 OVERVIEW

Tables 3 and 4 present a detailed overview of the quality metrics obtained from the experiment, accuracy, and F1 scores, which serve as well-established benchmarks in classification and detection tasks. These results reveal that for a significant portion (76.2%) of the TS, the metrics exceed the 0.9 mark, with the F1 Score averaging 0.892 and accuracy at 0.969. Precision, table 5, and Recall, table 6 averaged at 0.896 and 0.890 respectively. This indicates that the model demonstrates outstanding performance and robustness in data fitting, as evidenced by the high levels of accuracy and F1 scores attained. These findings also reinforce the effectiveness of the proposed method in delivering reliable classification results in open-set scenarios. The 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 in accurately identifying known classes and detecting unknown instances. Datasets such as ERing, Heartbeat, and MotorImagery display a notable variance in performance. Here, the test or open-set metrics are observed to be lower than those achieved during training, suggesting a propensity for overfitting within 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, a focused approach towards Machine Learning En-

gineering, 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. Figure 24 presents the the quality metrics overview of the experiment graphically.

Table 3 – F1 Score

Time Series	train	test	openset
ArticularyWordRecognition	1.00	0.89	0.78
AtrialFibrillation	0.97	0.79	0.83
BasicMotions	0.98	0.86	0.87
CharacterTrajectories	1.00	0.93	1.00
Coffee	0.97	0.96	1.00
Cricket	0.99	0.94	1.00
DuckDuckGeese	0.97	0.91	0.81
Epilepsy	0.98	0.85	0.66
ERing	0.99	0.45	1.00
EthanolConcentration	0.97	0.97	1.00
FingerMovements	1.00	0.59	1.00
HandMovementDirection	0.99	0.99	1.00
Handwriting	0.99	0.43	0.80
Heartbeat	0.88	0.38	1.00
Libras	0.99	0.99	1.00
MotorImagery	0.93	0.72	0.33
NATOPS	1.00	0.94	1.00
PEMS-SF	0.98	0.88	1.00
PenDigits	0.99	0.90	1.00
PhonemeSpectra	0.96	0.85	1.00
RacketSports	0.98	0.77	1.00
RefrigerationDevices	0.97	0.84	0.60
Rock	0.98	0.73	0.88
ScreenType	0.97	0.83	0.62
SelfRegulationSCP1	0.98	0.80	0.88
SelfRegulationSCP2	0.97	0.84	0.76
ShapesAll	0.99	0.79	1.00
SmallKitchenAppliances	0.97	0.78	0.74
SpokenArabicDigits	0.99	0.82	0.87
StandWalkJump	0.97	0.92	1.00
SwedishLeaf	0.99	0.94	1.00
TwoPatterns	0.98	0.74	0.69
UWaveGestureLibrary	0.99	0.90	1.00

The analysis also 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 in accurately identifying known classes and detecting unknown instances. Datasets such as ERing, Heartbeat, and MotorImagery display a notable variance in performance. Here, the test or open-set metrics are observed to be lower than those achieved during training, suggesting a propensity for overfitting within the model. This performance disparity, particularly

Table 4 – Accuracy Score

Time Series	train	test	openset
ArticularyWordRecognition	1.00	0.92	0.93
AtrialFibrillation	0.99	0.91	1.00
BasicMotions	1.00	0.99	0.98
CharacterTrajectories	1.00	1.00	1.00
Coffee	1.00	1.00	1.00
Cricket	1.00	1.00	1.00
DuckDuckGeese	0.99	0.98	0.83
Epilepsy	1.00	0.99	0.99
ERing	1.00	0.91	1.00
EthanolConcentration	0.98	0.98	1.00
FingerMovements	1.00	0.59	1.00
HandMovementDirection	0.99	0.99	1.00
Handwriting	0.99	0.82	0.93
Heartbeat	0.99	0.66	1.00
Libras	1.00	1.00	1.00
MotorImagery	0.99	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	0.96	0.95	1.00
RacketSports	1.00	0.92	1.00
RefrigerationDevices	1.00	0.99	0.84
Rock	0.99	0.87	0.97
ScreenType	0.99	0.97	0.98
SelfRegulationSCP1	1.00	0.89	0.96
SelfRegulationSCP2	0.99	0.97	0.90
ShapesAll	0.99	0.97	1.00
SmallKitchenAppliances	1.00	0.88	0.95
SpokenArabicDigits	1.00	0.99	1.00
StandWalkJump	0.98	0.94	1.00
SwedishLeaf	1.00	1.00	1.00
TwoPatterns	1.00	1.00	0.98
UWaveGestureLibrary	1.00	1.00	1.00

in the context of distinguishing between known and unknown classes, brings concerns regarding the model's generalization capabilities. However, a focused approach 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 5 – Precision Score

Time Series	train	test	openset
ArticularyWordRecognition	1.00	0.89	0.79
AtrialFibrillation	0.97	0.81	0.83
BasicMotions	0.98	0.87	0.88
CharacterTrajectories	1.00	0.93	1.00
Coffee	0.97	0.96	1.00
Cricket	0.99	0.94	1.00
DuckDuckGeese	0.98	0.91	0.83
Epilepsy	0.98	0.85	0.67
ERing	0.99	0.46	1.00
EthanolConcentration	0.97	0.97	1.00
FingerMovements	1.00	0.59	1.00
HandMovementDirection	0.99	0.99	1.00
Handwriting	0.99	0.44	0.81
Heartbeat	0.88	0.38	1.00
Libras	0.99	0.99	1.00
MotorImagery	0.93	0.73	0.33
NATOPS	1.00	0.94	1.00
PEMS-SF	0.98	0.89	1.00
PenDigits	0.99	0.90	1.00
PhonemeSpectra	0.96	0.85	1.00
RacketSports	0.99	0.79	1.00
RefrigerationDevices	0.97	0.84	0.62
Rock	0.98	0.76	0.89
ScreenType	0.97	0.83	0.62
SelfRegulationSCP1	0.98	0.83	0.89
SelfRegulationSCP2	0.97	0.84	0.78
ShapesAll	0.99	0.79	1.00
SmallKitchenAppliances	0.97	0.81	0.75
SpokenArabicDigits	0.99	0.82	0.88
StandWalkJump	0.98	0.93	1.00
SwedishLeaf	0.99	0.94	1.00
TwoPatterns	0.98	0.74	0.69
UWaveGestureLibrary	0.99	0.90	1.00

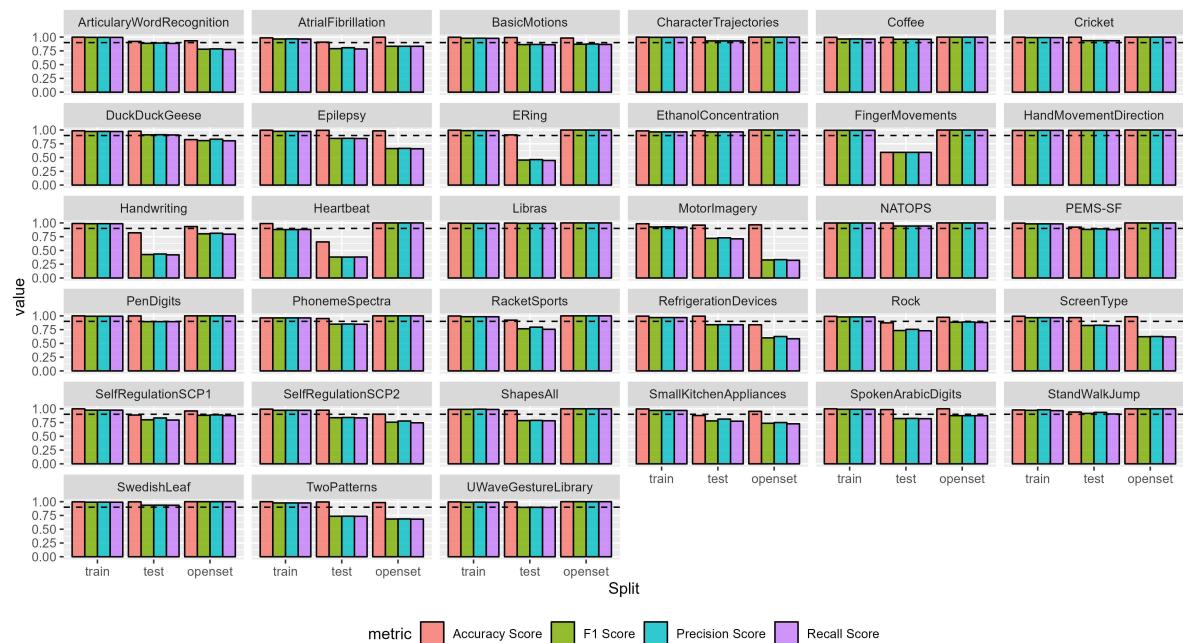
Figure 24 – Overview of Quality Metrics

Table 6 – Recall Score

Time Series	train	test	openset
ArticularyWordRecognition	1.00	0.89	0.78
AtrialFibrillation	0.96	0.78	0.83
BasicMotions	0.98	0.86	0.87
CharacterTrajectories	1.00	0.93	1.00
Coffee	0.96	0.96	1.00
Cricket	0.99	0.93	1.00
DuckDuckGeese	0.97	0.91	0.80
Epilepsy	0.98	0.85	0.66
ERing	0.99	0.45	1.00
EthanolConcentration	0.97	0.97	1.00
FingerMovements	1.00	0.59	1.00
HandMovementDirection	0.99	0.99	1.00
Handwriting	0.99	0.42	0.80
Heartbeat	0.88	0.38	1.00
Libras	0.99	0.99	1.00
MotorImagery	0.92	0.71	0.32
NATOPS	1.00	0.94	1.00
PEMS-SF	0.98	0.88	1.00
PenDigits	0.99	0.89	1.00
PhonemeSpectra	0.96	0.85	1.00
RacketSports	0.98	0.76	1.00
RefrigerationDevices	0.97	0.84	0.58
Rock	0.98	0.73	0.88
ScreenType	0.97	0.82	0.62
SelfRegulationSCP1	0.98	0.80	0.88
SelfRegulationSCP2	0.97	0.83	0.75
ShapesAll	0.99	0.78	1.00
SmallKitchenAppliances	0.97	0.77	0.73
SpokenArabicDigits	0.99	0.82	0.87
StandWalkJump	0.97	0.91	1.00
SwedishLeaf	0.99	0.94	1.00
TwoPatterns	0.98	0.73	0.68
UWaveGestureLibrary	0.99	0.90	1.00

In our pursuit of further understanding the results of the blending, we examine the correlations and dependencies among quality metrics across various splits for all time series. The scatter plot depicted in Figure 25 illustrates a predominantly positive relationship among the metrics, however with notable outliers that skew the magnitude of this linear association. To mitigate the influence of these outliers on our analysis, we compute Pearson, Spearman, and Kendall correlations, as presented in Figure 26 and tables table 7, 9, 8. While most correlations are positive across all types, none exceeds 0.5. This indicates, from a statistical standpoint, that the quality metrics of the test split are not significantly dependent on those of the train split, and similarly, the open-set metrics are independent of both train and test splits. Furthermore, the violin plot showcased in figure 27 assists in examining the distribution and variability of the metrics across splits. Notably, the train split consistently exhibits the highest values and lowest variance among all metrics. Conversely, the test and open-set splits demonstrate greater variance, with the values in the open-set split tending to cluster closer to 1 compared to the more dispersed values observed in the test split. These findings indicate that while the blending technique may exhibit lower performance levels on the test set, it consistently demonstrates superior performance on the open-set data, thus affirming its robustness across diverse datasets. This underscores its efficacy as a viable approach for time series classification in open-set scenarios, suggesting that the blending method effectively generalizes to previously unseen classes or instances, thereby attesting its applicability and reliability in real-world applications.

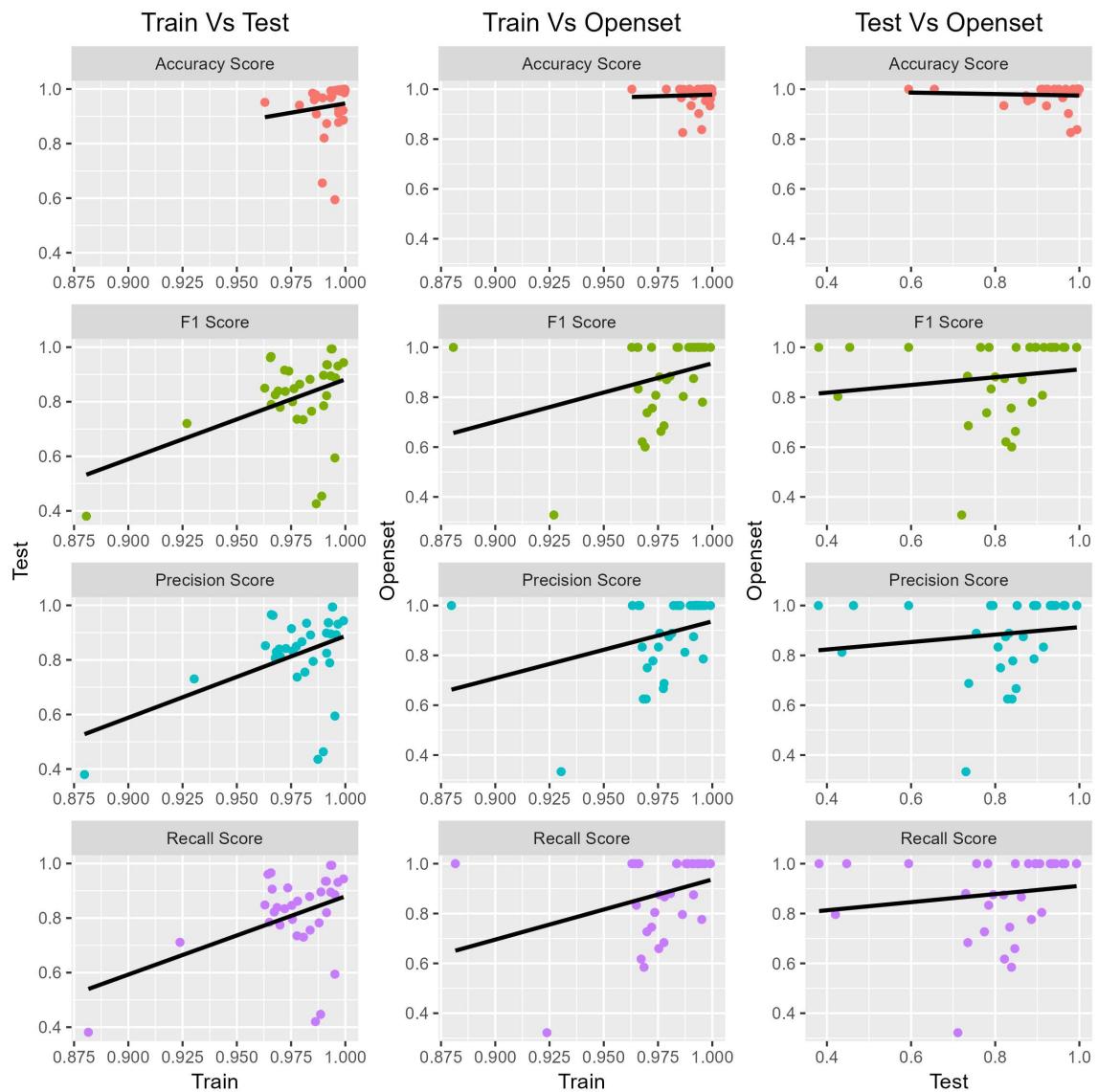
Figure 25 – Scatter plot of metrics

Figure 26 – Correlation Metrics

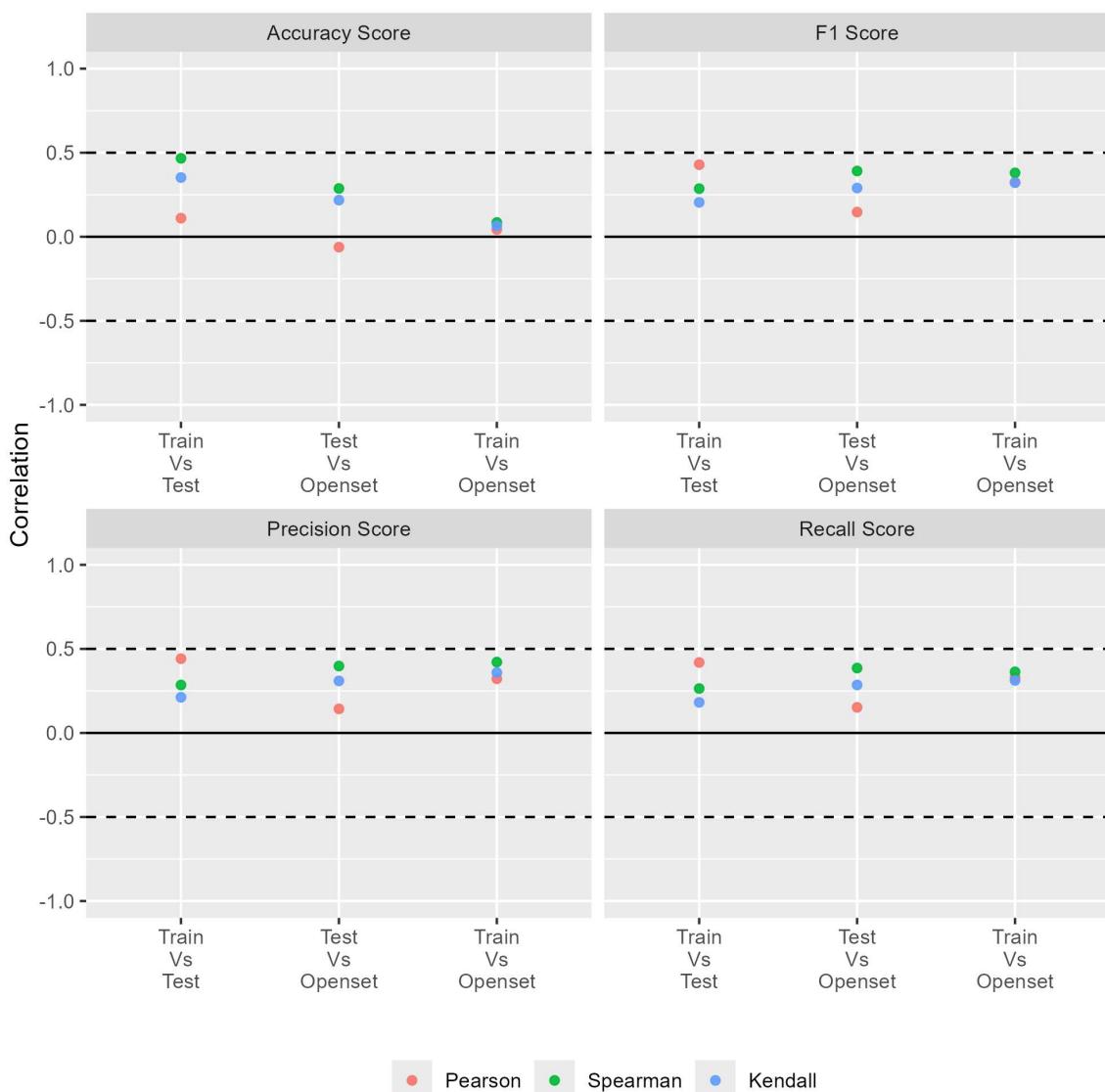
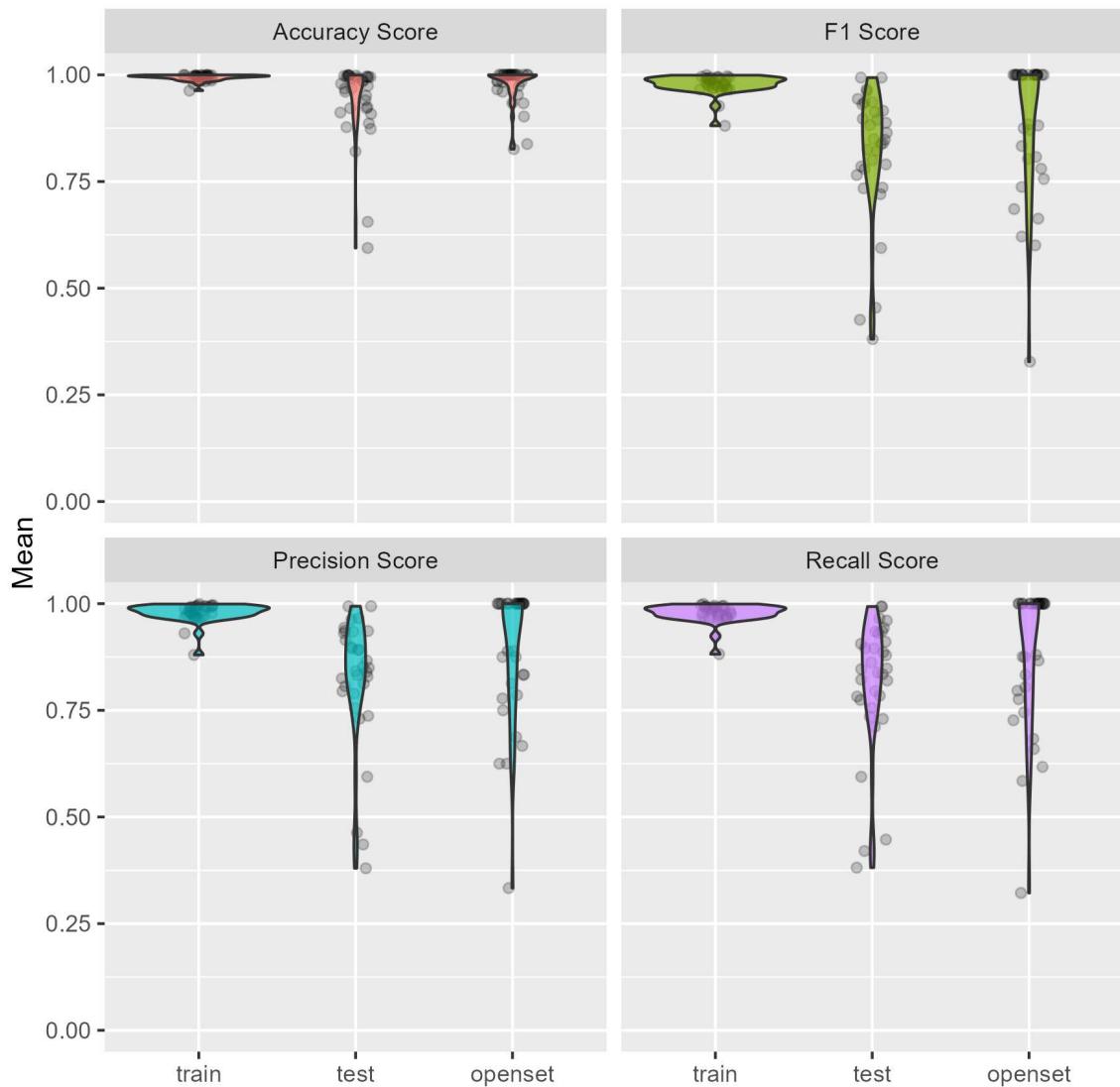


Figure 27 – Violin plot of metrics**Table 7 – Pearson Correlation**

Pair	Accuracy	F1	Precision	Recall
Test Vs Openset	-0.06	0.15	0.14	0.15
Train Vs Openset	0.04	0.32	0.32	0.33
Train Vs Test	0.11	0.43	0.44	0.42

Table 8 – Kendall Correlation

Pair	Accuracy	F1	Precision	Recall
Test Vs Openset	0.22	0.29	0.31	0.28
Train Vs Openset	0.06	0.32	0.36	0.31
Train Vs Test	0.35	0.20	0.21	0.18

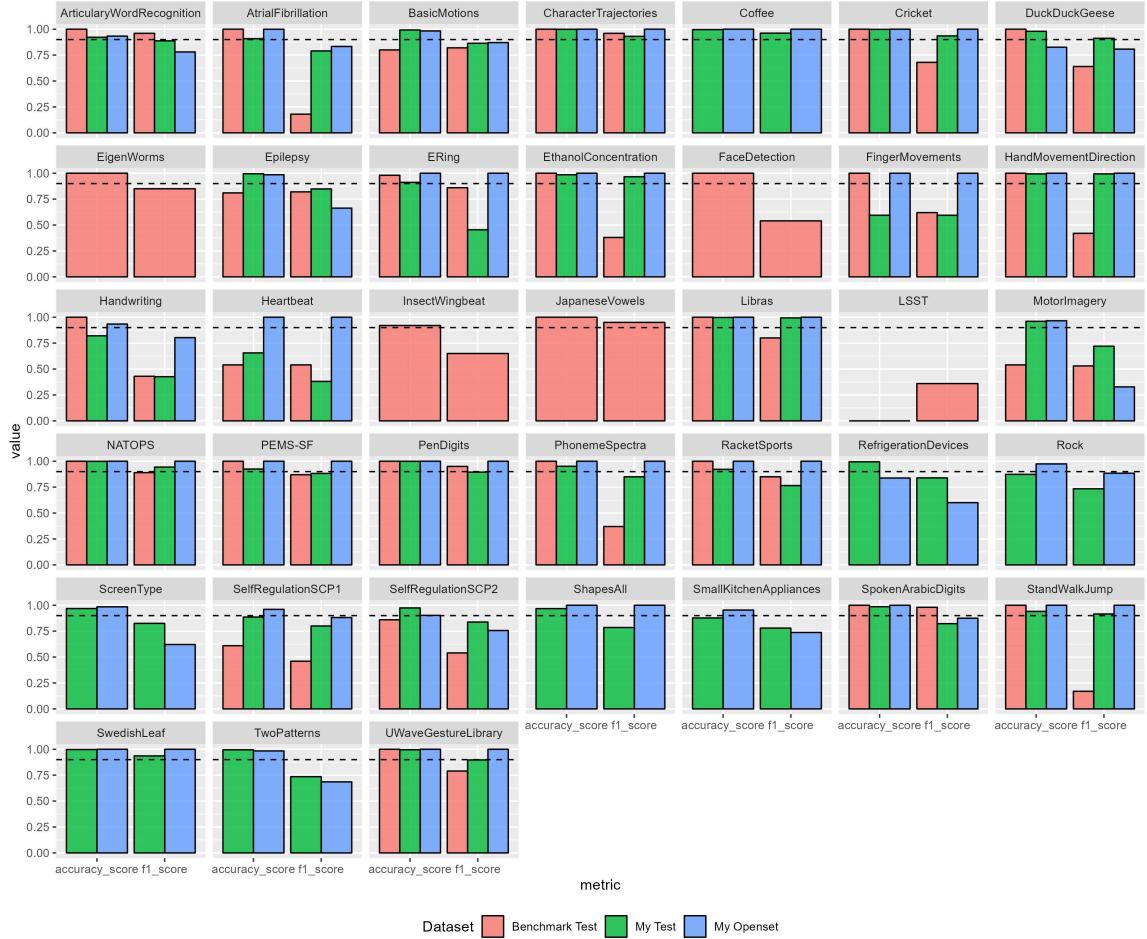
Table 9 – Spearman Correlation

Pair	Accuracy	F1	Precision	Recall
Test Vs Openset	0.29	0.39	0.40	0.39
Train Vs Openset	0.08	0.38	0.42	0.36
Train Vs Test	0.47	0.29	0.28	0.26

5.2 BENCHMARK

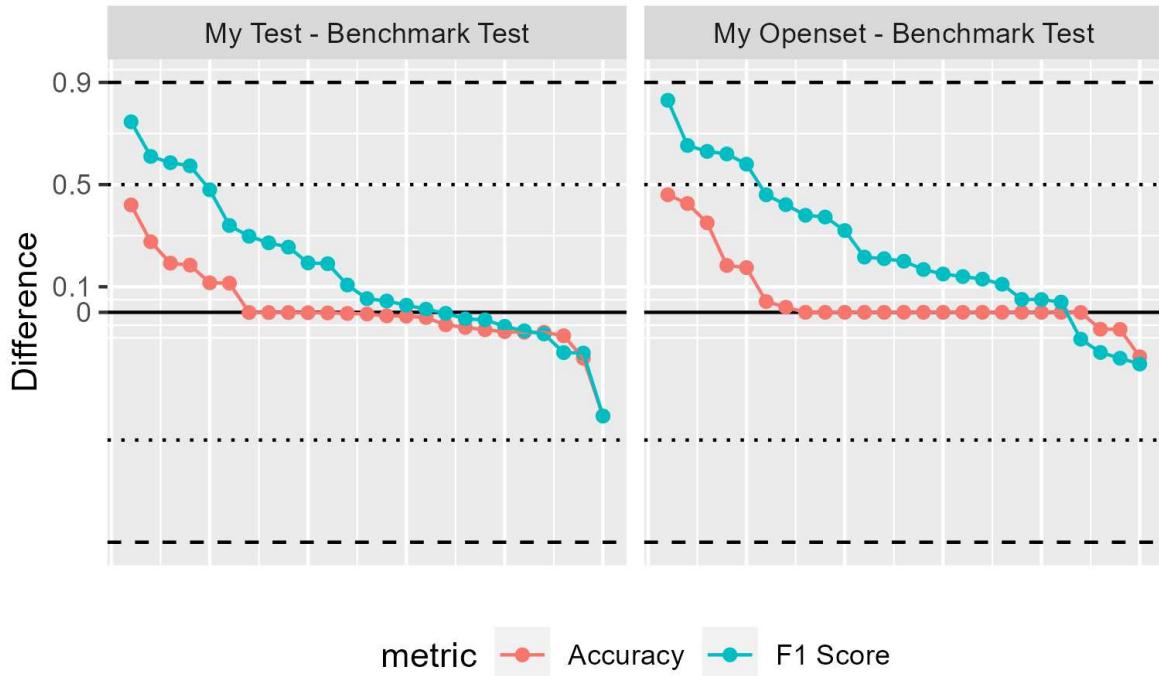
This subsection is dedicated to a comparative analysis to evaluate the effectiveness of our novel open-set recognition method. For this purpose, we reference (AKAR *et al.*, 2022), which establishes the benchmark against which our research is measured. Through a detailed comparison with this established benchmark, our objective is to highlight the unique strengths and significant contributions of our approach in the field of OSR. We must note that a direct comparison between our approach and the benchmark presented in (AKAR *et al.*, 2022) is only partially possible due to certain inherent dissimilarities in the experimental design. One notable distinction lies in the composition of the test split. Whereas the benchmark study employs only two unknown TS, our novel approach employs multiple unknown instances and introduces the additional open-set split. As a consequence the datasets in the benchmark have lower openness compared to ours (see Figure 31). However, 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 10 and 11 compare our novel method and the Benchmark. We notice the second dissimilarity between the methods: they do not share the same time series. This is a minor issue, as we have enough intersections to compare. The Benchmark’s F1 Score averages 0.66, whereas our model achieved 0.82. The average accuracy for Benchmark and our model are 0.90 and 0.94, respectively.

Figure 28 – Comparison overview between our novel method and the benchmark



A better method to compare the metrics involves computing the discrepancy between the metrics derived from the novel method and the benchmark. We computed the increment on the metric value by applying novel method calculating the differences Metric_{MyTest} - Metric_{BenchmarkTest} and Metric_{MyOpenset} - Metric_{BenchmarkTest}, for the two primary evaluation metrics, Accuracy and F1 Score. The resulting differential values are arranged in descending order, enabling the ranking of TS based on the magnitude of their metric difference. Figure 29 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 on the decrements suggests that a deeper investigation has the potential to bring improvements. These findings imply that our novel approach performs on par with or surpasses the benchmark, indicating its effectiveness and potential superiority in OSR tasks.

Figure 29 – Increment comparison



Analysing Figure 30, which illustrates the distribution of differences, it is observed that they are predominantly positive (indicative of increments). We also see the impact of outliers suggesting a deeper investigation to find potential improvements. Consequently, this reinforces that the novel approach presented is a competitive alternative for the benchmark. Tables 10 and 11 have the individual values.

Figure 30 – Increment Distribution



Table 10 – Compare Accuracy Score

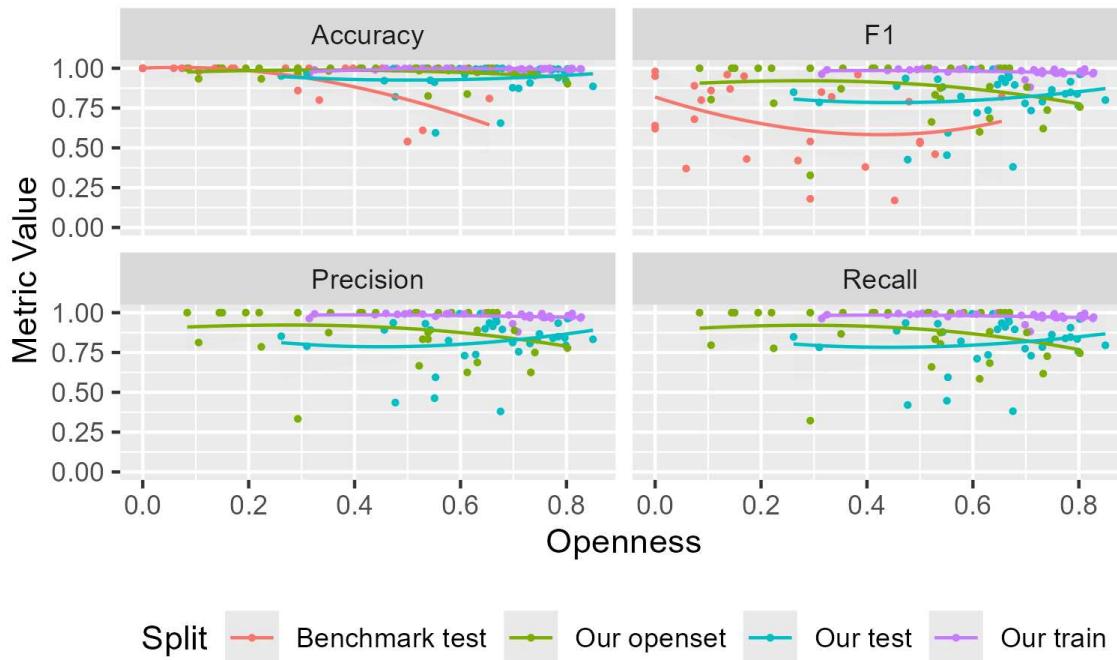
Time Series	(AKAR <i>et al.</i> , 2022)	Our Test	Our Open-set
ArticularyWordRecognition	1.00	0.92	0.93
AtrialFibrillation	1.00	0.91	1.00
BasicMotions	0.80	0.99	0.98
CharacterTrajectories	1.00	1.00	1.00
Coffee		1.00	1.00
Cricket	1.00	1.00	1.00
DuckDuckGeese	1.00	0.98	0.83
EigenWorms	1.00		
Epilepsy	0.81	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

Table 11 – Compare F1 Score

Time Series	(AKAR <i>et al.</i> , 2022)	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 31 illustrates the influence of dataset openness on quality metrics, employing loess (CATTANEO *et al.*, 2021) curves to elucidate the relationship between these variables. One can see that the quality metrics of the alternative method are less impacted by the degree of openness of the time series compared to the benchmark. Furthermore, it is evident that our alternative method consistently produces quality metrics that are either superior or at par with those achieved by the Benchmark for all splits. These findings attest the efficacy and resilience of our approach in time series open-set recognition, particularly in the face of varying levels of dataset openness, which will happen in real world scenarios. Notice there are Precision and recall for the benchmark as they are not present on the benchmark's paper.

Figure 31 – Quality Metrics VS Openness



6 CONCLUSION

Open-set recognition in time series classification is important for handling novel, unseen data in real-world applications. In healthcare it can assist by finding unseen medical conditions. It can be used in financial monitoring to detect fraud and money laundering. It can even be applied as an anomaly detection tool to identify earthquakes. Open set recognition improves the reliability and accuracy of time series classification systems making it more robust in dynamic environments.

In this study, we introduced a new methodology for Open Set Recognition in Time Series Classification, employing a blending of various Artificial Neural Networks equipped with OpenMax layer. This approach has demonstrated performance that is comparable to or exceeds that of the benchmark (AKAR *et al.*, 2022), achieving an average F1 Score of 0.82, which is 0.16 higher, and an average accuracy of 0.94, an increase of 0.04. Furthermore, the proposed model is more robust against varying degrees of openness and offers a more simplified and straightforward implementation. The model's consistent quality attests it as a potential superior option in openset recognition tasks. Future works could leverage the activation of intermediate layers to explore potential improvements, test several models to be used as the blending and more extensively explore the impact of number of known-unknown TS in the overall quality. A more challenging research is to train the neural network, or just the last layer, having the OpenMax acting as an activation function, aiming to optimize it for open set recognition.

All the objectives of this research have been accomplished and are detailed throughout the text. In Section 5.1, models tailored for Open Set Classification were built and their strengths and weaknesses evaluated. Transfer-learning was extensively applied across various neural networks, and the classification and detection capabilities of the models were integrated through blending ensembles. Known-unknown data was incorporated into the machine learning engineering pipeline, and exhaustive experiments using a wide and diverse set of Time Series data were conducted to rigorously evaluate the effectiveness of the proposed model. In Section 5.2, the performance of our models was compared against a benchmark (AKAR *et al.*, 2022), and exploratory data analysis was conducted to assess the relation between metrics, data, and model results.

Future research could bring improvements by leveraging activations from intermediate layers as features. Additionally, training and incorporating different models in the blending

process could provide insights into both prediction quality and the impact of the number of models on overall quality. Another area for exploration is the effect of the number of known-unknown time series on overall performance. A more challenging task for future work involves training neural networks from scratch, or fine-tuning the last layer, using OpenMax as the activation function to optimize the model for open set recognition.

REFERENCES

- AKAR, Tolga; WERNER, Thorben; YALAVARTHI, Vijaya Krishna; SCHMIDT-THIEME, Lars. Open set recognition for time series classification. In: GAMA, João; LI, Tianrui; YU, Yang; CHEN, Enhong; ZHENG, Yu; TENG, Fei (Ed.). **Advances in Knowledge Discovery and Data Mining**. Cham: Springer International Publishing, 2022. p. 354–366. ISBN 978-3-031-05936-0. Available at: <https://doi.org/10.1109/TPAMI.2005.224>.
- BAGNALL, Anthony; DAU, Hoang Anh; LINES, Jason; FLYNN, Michael; LARGE, James; BOSTROM, Aaron; SOUTHAM, Paul; KEOGH, Eamonn. **The UEA multivariate time series classification archive, 2018**. 2018. Available at: <https://www.doi.org/10.5281/zenodo.11206330>.
- BAGNALL, Anthony; LINES, Jason; BOSTROM, Aaron; AL. et. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. **Data Mining and Knowledge Discovery**, v. 31, p. 606–660, 2017. Available at: <https://doi.org/10.1007/s10618-016-0483-9>.
- BENDALE, Abhijit; BOULT, Terrance E. Towards open set deep networks. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [s.n.], 2016. p. 1563–1572. Available at: <https://doi.org/10.1109/CVPR.2016.173>.
- BODESHEIM, Paul; FREYTAG, Alexander; RODNER, Erik; KEMMLER, Michael; DENZLER, Joachim. Kernel null space methods for novelty detection. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [s.n.], 2013. Available at: <https://doi.org/10.1109/CVPR.2013.433>.
- CATTANEO, Matias D.; JANSSON, Michael; MA, Xinwei. **Local Regression Distribution Estimators**. 2021. Available at: <https://doi.org/10.48550/arXiv.2009.14367>.
- CHEN, Tianqi; GUESTRIN, Carlos. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. ACM, 2016. (KDD '16). Available at: <http://dx.doi.org/10.1145/2939672.2939785>.
- DAS, Shemonto. **An Active Learning Framework with a Class Balancing Strategy for Time Series Classification**. 2024. Available at: <https://doi.org/10.48550/arXiv.2405.12122>.
- DEMPSTER, Angus; SCHMIDT, Daniel F.; WEBB, Geoffrey I. **QUANT: A Minimalist Interval Method for Time Series Classification**. 2023. Available at: <https://arxiv.org/abs/2308.00928>.

FAOUZI, Johann. Time series classification: A review of algorithms and implementations. In: **Machine Learning (Emerging Trends and Applications)**. Proud Pen, 2022. Available at: <https://inria.hal.science/hal-03558165>.

FAWAZ, H. Ismail; FORESTIER, G.; WEBER, J.; AL. et. Deep learning for time series classification: a review. **Data Mining and Knowledge Discovery**, v. 33, p. 917–963, 2019. Available at: <https://doi.org/10.1007/s10618-019-00619-1>.

FAWAZ, Hassan Ismail; LUCAS, Benjamin; FORESTIER, Germain; PELLETIER, Charlotte; SCHMIDT, Daniel F.; WEBER, Jonathan; WEBB, Geoffrey I.; IDOUMGHAR, Lhassane; MULLER, Pierre-Alain; PETITJEAN, François. Inceptiontime: Finding alexnet for time series classification. **Data Mining and Knowledge Discovery**, Springer Science and Business Media LLC, v. 34, n. 6, p. 1936–1962, Sep. 2020. ISSN 1573-756X. Available at: <http://dx.doi.org/10.1007/s10618-020-00710-y>.

FISHER, R. A.; TIPPETT, L. H. C. Limiting forms of the frequency distribution of the largest or smallest member of a sample. **Mathematical Proceedings of the Cambridge Philosophical Society**, v. 24, n. 2, p. 180–190, 1928. Available at: [10.1017/S0305004100015681](https://doi.org/10.1017/S0305004100015681).

GE, ZongYuan; DEMYANOV, Sergey; CHEN, Zetao; GARNAVI, Rahil. **Generative OpenMax for Multi-Class Open Set Classification**. 2017. Available at: <https://doi.org/10.48550/arXiv.1707.07418>.

GENG, Chuanxing; HUANG, Sheng-Jun; CHEN, Songcan. Recent advances in open set recognition: A survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Institute of Electrical and Electronics Engineers (IEEE), v. 43, n. 10, p. 3614–3631, Oct. 2021. ISSN 1939-3539. Available at: <http://dx.doi.org/10.1109/TPAMI.2020.2981604>.

GHOJOGH, Benyamin; GHODSI, Ali. **Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and Survey**. 2023. Available at: <https://arxiv.org/abs/2304.11461>.

GRABOCKA, Josif; SCHILLING, Nathanael; WISTUBA, Martin; SCHMIDT-THIEME, Lars. Learning time-series shapelets. In: **Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)**. [s.n.], 2014. Available at: <https://doi.org/10.1145/2623330.2623613>.

HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. **Deep Residual Learning for Image Recognition**. 2015. Available at: <https://www.doi.org/10.1109/cvpr.2016.90>.

HILLS, Jon; LINES, Jason; BARANAUSKAS, Eimantas; MAPP, Jonathan; BAGNALL, Anthony. Classification of time series by shapelet transformation. **Data Mining and Knowledge Discovery**, v. 28, n. 4, p. 851–881, 2014. Available at: <https://doi.org/10.1007/s10618-013-0322-1>.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997. Available at: <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.

JAIN, Lalit P.; SCHEIRER, Walter J.; BOULT, Terrance E. Multi-class open set recognition using probability of inclusion. In: FLEET, David; PAJDLA, Tomas; SCHIELE, Bernt; TUYTELAARS, Tinne (Ed.). **Computer Vision – ECCV 2014**. Cham: Springer International Publishing, 2014. p. 393–409. ISBN 978-3-319-10578-9. Available at: https://doi.org/10.1007/978-3-319-10578-9_26.

KARIM, Fazle; MAJUMDAR, Somshubra; DARABI, Houshang; CHEN, Shun. Lstm fully convolutional networks for time series classification. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 1662–1669, 2018. ISSN 2169-3536. Available at: <http://dx.doi.org/10.1109/ACCESS.2017.2779939>.

LE, Xuan-May; LUO, Ling; AICKELIN, Uwe; TRAN, Minh-Tuan. **ShapeFormer: Shapelet Transformer for Multivariate Time Series Classification**. 2024. Available at: <https://arxiv.org/abs/2405.14608>.

LI, Jianquan; YANG, Ying; LIU, Gang; NING, Yuanlin; SONG, Ping. Open-set sheep face recognition in multi-view based on li-sheepfacenet. **Agriculture**, v. 14, p. 1112, 07 2024. Available at: <https://www.mdpi.com/2077-0472/14/7/1112>.

LI, Zewen; YANG, Wenjie; PENG, Shouheng; LIU, Fan. **A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects**. 2020. Available at: <https://arxiv.org/abs/2004.02806>.

LIN, Jessica; KEOGH, Eamonn; LI, Wei; LONARDI, Stefano. Experiencing sax: a novel symbolic representation of time series. **Data Mining and Knowledge Discovery**, v. 15, n. 2, p. 107–144, 2007. Available at: <https://doi.org/10.1007/s10618-007-0064-z>.

LIU, Kevin; DEMORI, Julien; ABAYOMI, Kobi. **Open Set Recognition For Music Genre Classification**. 2024. Available at: <https://doi.org/10.48550/arXiv.2209.07548>.

MAHDAVI, Atefeh; CARVALHO, Marco. A survey on open set recognition. In: **2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)**. IEEE, 2021. Available at: <https://doi.org/10.1109%2Faike52691.2021.00013>.

MENSINK, Thomas; VERBEEK, Jakob; PERRONNIN, Florent; CSURKA, Gabriela. Distance-based image classification: Generalizing to new classes at near-zero cost. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 35, n. 11, p. 2624–2637, 2013. Available at: <https://doi.org/10.1109/TPAMI.2013.83>.

OH, Hyeryeong; KIM, Seoung Bum. Multivariate time series open-set recognition using multi-feature extraction and reconstruction. **IEEE Access**, v. 10, 2022. Available at: <https://doi.org/10.1109/ACCESS.2022.3222310>.

PUPPO, Luke; WONG, Weng-Keen; HAMDAOUI, Bechir; ELMAGHBUB, Abdurrahman. Hinova: A novel open-set detection method for automating rf device authentication. In: **2023 IEEE Symposium on Computers and Communications (ISCC)**. [s.n.], 2023. p. 1122–1128. Available at: <https://doi.org/10.1109/ISCC58397.2023.10217841>.

RAHIMIAN, Elahe; ZABIHI, Soheil; ATASHZAR, Seyed Farokh; ASIF, Amir; MOHAMMADI, Arash. **XceptionTime: A Novel Deep Architecture based on Depthwise Separable Convolutions for Hand Gesture Classification**. 2019.

ROMANOVA, Alex. Enhancing time series analysis with gnn graph classification models. In: _____. [S.l.]: Complex Networks Their Applications XII, 2024. p. 25–36. ISBN 978-3-031-53467-6.

ROZSA, Andras; GÜNTHER, Manuel; BOULT, Terrance E. Adversarial robustness: Softmax versus openmax. **arXiv preprint arXiv:1708.01697**, 2017. Available at: <https://arxiv.org/abs/1708.01697>.

RUDD, Ethan M.; JAIN, Lalit P.; SCHEIRER, Walter J.; BOULT, Terrance E. The extreme value machine. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Institute of Electrical and Electronics Engineers (IEEE), v. 40, n. 3, p. 762–768, Mar. 2018. ISSN 2160-9292. Available at: <http://dx.doi.org/10.1109/TPAMI.2017.2707495>.

RUDD, Ethan M.; JAIN, Lalit P.; SCHEIRER, Walter J.; BOULT, Terrance E. The extreme value machine. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Institute of Electrical and Electronics Engineers (IEEE), v. 40, n. 3, p. 762–768, March 2018. Available at: <https://doi.org/10.1109/tpami.2017.2707495>.

SAKOE, H; CHIBA, S. Dynamic programming algorithm optimization for spoken word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 26, n. 1, p. 43–49, 1978. Available at: <https://doi.org/10.1109/TASSP.1978.1163055>.

SCHEIRER, Walter J.; ROCHA, Anderson de Rezende; SAPKOTA, Archana; BOULT, Terrance E. Toward open set recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 35, n. 7, p. 1757–1772, 2013. Available at: <https://pubmed.ncbi.nlm.nih.gov/23682001/>.

SHELHAMER, Evan; LONG, Jonathan; DARRELL, Trevor. **Fully Convolutional Networks for Semantic Segmentation**. 2016. Available at: <https://www.doi.org/10.1109/CVPR.2015.7298965>.

SRIDHAR, Sripathi; CARTWRIGHT, Mark. **Multi-label Open-set Audio Classification**. 2023. Available at: <https://arxiv.org/abs/2310.13759>.

TANG, Wensi; LONG, Guodong; LIU, Lu; ZHOU, Tianyi; BLUMENSTEIN, Michael; JIANG, Jing. **Omni-Scale CNNs: a simple and effective kernel size configuration for time series classification**. 2022. Available at: <https://www.doi.org/10.1016/j.neunet.2021.01.001>.

VARETO, Rafael Henrique; SCHWARTZ, William Robson. Unconstrained face identification using ensembles trained on clustered data. In: **2020 IEEE International Joint Conference on Biometrics (IJCB)**. IEEE, 2020. Available at: <http://dx.doi.org/10.1109/IJCB48548.2020.9304882>.

WANG, Jingyuan; WANG, Ze; LI, Jianfeng; WU, Junjie. **Multilevel Wavelet Decomposition Network for Interpretable Time Series Analysis**. 2018. Available at: <https://www.doi.org/10.1145/3219819.3220060>.

WANG, Zhiguang; YAN, Weizhong; OATES, Tim. **Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline**. 2016. Available at: <https://arxiv.org/abs/1611.06455>.

WEN, L.; DONG, Y.; GAO, L. A new ensemble residual convolutional neural network for remaining useful life estimation. **Mathematical Biosciences and Engineering**, v. 16, n. 2, p. 862–880, Jan. 2019. Available at: <https://doi.org/10.3934/mbe.2019040>.

W.K., et. al Wang. A Systematic Review of Time Series Classification Techniques Used in Biomedical Applications. **Sensors** **2022**, 2022. Available at: <https://doi.org/10.3390/s22208016>.

WOLPERT, David. Stacked generalization. **Neural Networks**, v. 5, p. 241–259, 12 1992. Available at: [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1).

YANG, Honghui; ZHENG, Kaifeng; LI, Junhao. Open set recognition of underwater acoustic targets based on gru-cae collaborative deep learning network. **Applied Acoustics**, v. 193, p. 108774, 2022. ISSN 0003-682X. Available at: <https://www.sciencedirect.com/science/article/pii/S0003682X22001487>.

YE, Lexiang; KEOGH, Eamonn. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. **Data Mining and Knowledge Discovery**, v. 22, n. 1-2, p. 149–182, 2011. Available at: <https://doi.org/10.1007/s10618-011-0244-0>.

YOU, Jie; WU, Wenqin; LEE, Joonwhoan. Open set classification of sound event. **Scientific Reports**, v. 14, 01 2024. Available at: <https://doi.org/10.1038/s41598-023-50639-7>.