FEDERAL UNIVERSITY OF AMAZONAS - UFAM

INSTITUTE OF COMPUTING - ICOMP

POSTGRADUATE PROGRAM IN INFORMATICS - PPGI

Unified Time Series Framework for Explainable Artificial Intelligence

Hendrio L. S. Bragança

Manaus - AM

February, 2025

Hendrio L. S. Bragança

Unified Time Series Framework for Explainable Artificial Intelligence

Thesis presented to the Postgraduate Program in Informatics of the Institute of Computing of Federal University of Amazonas for the degree of Doctor in Informatics

Advisor

DSc Eduardo Souto

Federal University of Amazonas - UFAM

Institute of Computing - IComp

Manaus - AM

February, 2025

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

B813u Bragança, Hendrio Luis de Souza
Unified Time Series Framework for Explainable Artificial Intelligence / Hendrio Luis de Souza Bragança. - 2025.
143 f. : il., color. ; 31 cm.

Orientador(a): Eduardo James Pereira Souto.
Tese (doutorado) - Universidade Federal do Amazonas, Programa de Pós-Graduação em Informática, Manaus, 2025.

1. Explainable Artificial Intelligence (XAI). 2. Time-Series data. 3. Machine Learning. 4. Signal Processing. I. Souto, Eduardo James Pereira. II. Universidade Federal do Amazonas. Programa de Pós-Graduação em Informática. III. Título

---

Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Informática

FOLHA DE APROVAÇÃO

"UNIFIED TIME SERIES FRAMEWORK FOR EXPLAINABLE ARTIFICIAL INTELLIGENCE"

HENDRIO LUIS DE SOUZA BRAGANÇA

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Eduardo James Pereira Souto - **Presidente**

Prof. Dr. Rodrigo de Melo Souza Veras - **Membro Externo**

Profa. Dra. Andrea Gomes Campos - **Membro Externo**

Profa. Dra. Natalia Castro Fernandes - **Membro Externo**

Prof. Dr. Juan Gabriel Colonna - **Membro Interno**

Manaus, 17 de fevereiro de 2025.

Documento assinado eletronicamente por **Eduardo James Pereira Souto**, **Professor do Magistério Superior**, em 17/02/2025, às 12:22, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Juan Gabriel Colonna**, **Professor do Magistério Superior**, em 06/05/2025, às 21:55, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Andrea Gomes Campos**, **Usuário Externo**, em 07/05/2025, às 11:22, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Natalia Castro Fernandes**, **Usuário Externo**, em 08/05/2025, às 11:30, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rodrigo de Melo Souza Veras**, **Usuário Externo**, em 23/05/2025, às 10:41, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2449510** e o código CRC **4DA8B825**.

Avenida General Rodrigo Octávio, 6200 - Bairro Coroado I Campus Universitário Senador Arthur Virgílio Filho, Setor Norte - Telefone: (92) 3305-1181 / Ramal 1193
CEP 69080-900, Manaus/AM, coordenadorppgi@icomp.ufam.edu.br

Dedico esta obra a Deus e a todas as pessoas especiais que tornaram minha jornada possível: à minha querida esposa, por seu amor e parceria; ao meu filho, que me inspira a buscar um futuro melhor; à minha mãe, por seu amor e resiliência; e ao meu orientador, cuja experiência e incentivo foram fundamentais para a concretização deste trabalho. Sem cada um de vocês, nada disso seria possível.

Acknowledgements

First and foremost, I offer my deepest gratitude to God, who has been my unwavering source of strength, wisdom, and guidance throughout this journey. I am profoundly grateful to my wife, Vanessa, whose love, patience, and unwavering support have been my foundation. Her encouragement, sacrifices, and belief in me have been invaluable, providing the strength I needed to stay committed to this journey. To my son, Gustavo, you are my greatest inspiration. Your joy, curiosity, and boundless energy remind me every day of the importance of perseverance and the pursuit of knowledge. To my mother, Eliane, your unconditional love and guidance have shaped the person I am today. Your sacrifices, wisdom, and endless encouragement have been my driving force, and I owe much of my success to your unwavering support.

I extend my sincere gratitude to my advisor, DSc Eduardo Souto, whose mentorship, guidance, and patience have been important in shaping this research.

I would also like to extend my gratitude to my colleagues, friends, and fellow students, who have provided me with a supportive and encouraging academic environment. Finally, I would like to acknowledge the support and assistance of the Program in Informatics of the Institute of Computing of Federal University of Amazonas, which provided me with the resources and infrastructure necessary to complete my research. I would also like to express my appreciation to the CAPES, FAPEAM, BASEGRANT program, and the PROPPGI program for their financial support, which has been crucial in allowing me to pursue my research goals.

I am truly grateful for all of the support and guidance I have received, and I hope that this thesis will stand as a testament to all of their hard work and dedication.

If I have seen further, it is by standing upon the shoulders of giants.

Sir Isaac Newton

Unified Time Series Framework for Explainable Artificial Intelligence

Autor: Hendrio L. S. Bragança

Orientador: DSc Eduardo Souto

Abstract

The increasing complexity of machine learning (ML) models has made their decision-making processes difficult to interpret, posing a critical challenge in high-stakes domains where trust and transparency are essential. Although Explainable Artificial Intelligence (XAI) methods aim to address this issue, most existing techniques face limitations when applied directly to time series data due to its sequential and contextual nature. In this work, we present the Unified Time Series Framework for Explainable Artificial Intelligence (UTS-XAI), which integrates a standard time series classification pipeline with explainability capabilities and domain-specific evaluation tools. The framework is compatible with multiple explainability methods, such as SHAP, LIME, and Saliency Maps, and supports their systematic evaluation through adapted versions of widely used XAI metrics (faithfulness, robustness, sensitivity, and stability) reinterpreted for temporal data. These metrics are combined with time series–specific similarity and distance measures such as MSE, MAE, and DTW to quantify explanation quality. We also introduce Global Interpretable Clustering (GIC), a visualization technique designed to assess the consistency of feature attributions across explainers and models. Experiments conducted on three real-world cardiac arrhythmia datasets (MITBIH, SVDB, INCART), using three ML architectures (XGBoost, DeepConvLSTM, and FCN), show that SHAP provides more faithful and stable explanations, while LIME and Saliency Maps exhibit greater sensitivity to noise and perturbations. These results highlight that accuracy alone

is not sufficient in time series modeling without robust interpretability. By embedding explainability into the model development lifecycle, UTS-XAI sets a new standard for interpretable and trustworthy AI in temporal data analysis.

Key-words: Explainable Artificial Intelligence (XAI), Time-Series data, Machine Learning.

Unified Time Series Framework for Explainable Artificial Intelligence

Autor: Hendrio L. S. Bragança

Orientador: DSc Eduardo Souto

Resumo

A crescente complexidade dos modelos de aprendizado de máquina (ML) tornou seus processos de tomada de decisão difíceis de interpretar, representando um desafio crítico em domínios de alto risco onde confiança e transparência são essenciais. Embora os métodos de Inteligência Artificial Explicável (XAI) visem abordar essa questão, a maioria das técnicas existentes enfrenta limitações quando aplicadas diretamente a dados de séries temporais devido à sua natureza sequencial e contextual. Neste trabalho, apresentamos o framework UTS-XAI (Unified Time Series Framework for Explainable Artificial Intelligence), que integra um pipeline padrão de classificação de séries temporais com recursos de explicabilidade e ferramentas de avaliação específicas de domínio. O framework é compatível com múltiplos métodos de explicabilidade, como SHAP, LIME e Mapas de Saliência, e suporta sua avaliação sistemática por meio de versões adaptadas de métricas de XAI amplamente utilizadas (fidelidade, robustez, sensibilidade e estabilidade) reinterpretadas para dados temporais. Essas métricas são combinadas com medidas de similaridade e distância específicas de séries temporais, como MSE, MAE e DTW, para quantificar a qualidade da explicação. Também apresentamos o Global Interpretable Clustering (GIC), uma técnica de visualização projetada para avaliar a consistência das atribuições de características entre explicadores e modelos. Experimentos conduzidos em três conjuntos de dados de arritmia cardíaca do mundo real

(MITBIH, SVDB, INCART), utilizando três arquiteturas de ML (XGBoost, DeepConvLSTM e FCN), mostram que o SHAP fornece explicações mais fiéis e estáveis, enquanto o LIME e os Mapas de Saliência exibem maior sensibilidade a ruídos e perturbações. Esses resultados destacam que a precisão por si só não é suficiente na modelagem de séries temporais sem uma interpretabilidade robusta. Ao incorporar a explicabilidade ao ciclo de vida de desenvolvimento do modelo, o UTS-XAI estabelece um novo padrão para IA interpretável e confiável na análise de dados temporais.

Keywords: Inteligência Artificial Explicável (XAI), Séries Temporais, Aprendizado de Máquina.

List of Figures

List of Tables

List of abbreviations and acronyms

AI *Artificial Intelligence*

CNN *Convolutional Neural Network*

CS *Cosine Similarity*

DeepConvLSTM *DeepConvLSTM*

DTW *Dynamic Time Warping*

ECG *Electrocardiogram*

ED *Euclidean Distance*

EX *SHAP Explainer*

FCN *Fully Convolutional Network*

GIC *Global Interpretable Clustering*

HAR *Human Activity Recognition*

INCART *St. Petersburg INCART Arrhythmia Database*

LIME *Local Interpretable Model-Agnostic Explanations*

LOSO *Leave-One-Subject-Out*

LTE *LIME Tabular Explainer*

MAE *Mean Absolute Error*

MIT-BIH *MIT-BIH Arrhythmia Database*

ML *Machine Learning*

MSE *Mean Squared Error*

PCA *Principal Component Analysis*

RMSE *Root Mean Squared Error*

SHAP *SHapley Additive exPlanations*

SM *Saliency Maps*

SVDB *MIT-BIH Supraventricular Arrhythmia Database*

t-SNE *t-distributed Stochastic Neighbor Embedding*

TEX *SHAP Tree Explainer*

TSC *Time Series Classification*

UMAP *Uniform Manifold Approximation and Projection*

UTS-XAI *Unified Time Series Framework for Explainable Artificial Intelligence*

XAI *Explainable Artificial Intelligence*

XGBoost *Extreme Gradient Boosting*

CONTENTS

1

INTRODUCTION

Advances in Artificial Intelligence (AI) and Machine Learning (ML) have driven the global adoption of these technologies due to their exceptional performance in various domains. However, the increasing complexity and nonlinear structure of cutting-edge models, such as deep neural networks, make it difficult to understand how they arrive at their decisions (SOKOL; VOGT, 2024). These so-called "black-box" models have generated uncertainty and skepticism, particularly in domains where understanding the rationale behind predictions is critical (GUIDOTTI et al., 2018; ADADI; BERRADA, 2020; RUDIN, 2019; PAWLICKA et al., 2023).

A central challenge lies in reconciling the remarkable accuracy of ML models with the need for transparency and trust. When stakeholders cannot interpret ML model results, implementing these systems in high-risk areas becomes problematic. In medical applications, for instance, clinicians and patients alike must understand why a model suggests a particular diagnosis or treatment to trust its recommendations fully. Without such understanding, even the most accurate model may be deemed unsafe or unfit for real-world decision-making (HOHMAN et al., 2018; RUDIN, 2019; LINARDATOS; PAPASTEFANOPOULOS; KOTSIANTIS, 2021; ROJAT et al., 2021; CHEN et al., 2023; CHADDAD et al., 2023).

Explainable Artificial Intelligence (XAI) has emerged as a promising field that seeks to address these concerns by making complex ML models more interpretable and understandable. XAI seeks to complement powerful ML architectures with mechanisms that explain how and why models make their predictions, thus promoting trust, en-

hancing system debugging, and guiding model refinement. The existent XAI methods have already made significant progress, offering granular explanations for complex models and allowing human users to identify errors, biases, and unexpected behaviors, ultimately fostering greater acceptance of ML systems (LUNDBERG; LEE, 2017; RUDIN, 2019).

Despite these advances, current XAI methods often remain segregated from the core ML development pipeline. Models are frequently trained first, and only afterward are XAI methods applied as a separate, add-on step. This disjointed approach may limit the potential benefits of XAI, as explanations are often produced post hoc and may not be fully integrated into the model development, selection, and deployment lifecycle. To realize the full promise of both accuracy and understandable, a unified framework that seamlessly merges traditional ML pipelines with state-of-the-art XAI methods is needed.

Early Explainable AI approaches have predominantly focused on image-based tasks, employing visual explanation methods such as heatmaps or saliency maps overlaid on images to highlight key regions influencing the model's prediction (HOHMAN et al., 2018; SCHLEGEL; KEIM, 2021). These intuitive strategies are effective in domains where spatial correlations are easily visualized, helping bridge the gap between complex model outputs and meaningful human understanding.

However, applying these approaches to different data types presents major challenges. Time-series data, for example, are fundamentally different from images since they describe temporal sequences rather than visual features. Temporal correlations, contextual patterns over time, and multisensor inputs complicate the generation of clear, intuitive, and reliable explanations. Although techniques such as Local Interpretable Model-Agnostic Explanations (LIME) (RIBEIRO; SINGH; GUESTRIN, 2016b) and SHapley Additive exPlanations (SHAP) (LUNDBERG; LEE, 2017) have shown promise as general-purpose XAI methods, their direct application to time series data is not straightforward. Factors such as evolving dependencies, changing the importance of features over time, and the difficulty of visually integrating attributions into line plots demand novel explanation strategies and domain-specific adaptations (SCHLEGEL;

KEIM, 2021; ROJAT et al., 2021).

Another significant challenge lies in evaluating the quality and reliability of explanations for time series models (GUIDOTTI et al., 2018; SCHLEGEL et al., 2019; SCHLEGEL et al., 2020; ROJAT et al., 2021; PAWLICKA et al., 2023; MIRZAEI et al., 2023; SOKOL; VOGT, 2024; LONGO et al., 2024). Although there are a variety of qualitative and quantitative metrics to assess the quality of explanations in image-based tasks, these metrics were not designed with the complexities of time series data in mind.

Qualitative evaluations often rely on human judgment and comprehensibility (RIBEIRO; SINGH; GUESTRIN, 2016b), but interpreting line plots with relevance scores is inherently more abstract. Quantitative evaluations, including methods such as pixel flipping (SAMEK; WIEGAND; MÜLLER, 2017), sanity checks (ADEBAYO et al., 2018), or sensitivity analyses (REBUFFI et al., 2020), have largely been used for images. Their assumptions and validation strategies do not always translate well to time series, where temporal correlations, multi-sensor inputs, and evolving patterns challenge conventional notions of critical features and local neighborhoods.

Consequently, new visualization strategies and standardized evaluation methodologies that account for temporal context and complexity are needed to improve XAI for time-series data (ADEBAYO et al., 2018; REBUFFI et al., 2020; ROJAT et al., 2021).

This thesis aims to overcome these limitations by proposing the Unified Time Series Framework for Explainable Artificial Intelligence (UTS-XAI) that integrates the traditional time series classification pipeline with state-of-the-art XAI methodologies and domain-specific evaluation metrics. By integrating explainability methods in the modeling process, UTS-XAI aims to make interpretability a central design principle rather than a post hoc addition. We also propose intuitive visualization tools for time-series data so-caled Global Interpretable Clustering (GIC) that provides a structured methodology for evaluating the consistency and reliability of explainability methods. Finally, we design new XAI evaluation metrics that fit temporal patterns and systematically validate their effectiveness with real-world datasets.

To validate UTS-XAI, we conducted extensive experiments on three real-world time-series datasets (MIT-BIH, SVDB, and INCART) for cardiac arrhythmia classifica-

tion, evaluating the framework across three machine learning architectures (XGBoost, DeepConvLSTM, and Fully Convolutional Networks (FCN)) and three widely used XAI techniques (SHAP, LIME, and Saliency Maps). Our results demonstrate that faithfulness evaluations reveal that SHAP-based methods (e.g. Explainer) consistently produce the most accurate feature attributions, while LIME and Saliency Maps often fail to capture critical temporal dependencies. Robustness evaluations indicate that Saliency Maps are highly sensitive to noise, often producing unstable attributions under perturbations, whereas SHAP explanations remain more consistent. Sanity checks confirm that some XAI techniques (e.g., LIME) struggle to differentiate between meaningful and random feature attributions, raising concerns about their reliability. Localization metrics show that XAI methods perform differently across models, with tree-based methods (XGBoost + SHAP) achieving better alignment with expert-defined relevant features, while deep models require additional refinement to improve interpretability.

Our proposed Global Interpretable Clustering provides an effective qualitative approach to understanding feature attribution stability, revealing that certain XAI methods produce inconsistent explanations across different model architectures. We notice an interesting trend related to thresholding: as we increase the importance filter, features that are globally relevant but not locally prominent become more influential in defining cluster structure. This is evident in the UMAP and PCA plots, where clusters at higher thresholds appear more compact, showing that the thresholding operation highlights globally consistent feature contributions, refining cluster definitions as the threshold increases.

Despite achieving high classification accuracy for arrhythmia detection, our findings highlight that accuracy alone is insufficient without rigorous evaluation of model explanations. Through UTS-XAI, we demonstrate that structured explainability evaluation not only enhances trust in AI-driven decisions, but also enables meaningful model refinement by identifying weaknesses in interpretability.

We fundamentally rethink how explainability should be incorporated into time-series classification. By moving beyond post hoc explanations and embedding XAI into the model development pipeline, UTS-XAI establishes a new standard for interpretable,

transparent, and trustworthy AI in sequential data analysis.

1.1 Research Objectives

This thesis aims to propose and evaluate an unified framework for time series classification that integrates a traditional machine learning classification pipeline with a novel domain-specific Explainable AI methodology, thereby enhancing the interpretability, transparency, and trustworthiness of machine learning models in time-series applications. Overall, it aims to answer the following research question.

> **"How can we integrate advanced explainable artificial intelligence methods and improve explainability evaluation into time-series classification to develop robust, trustworthy, and interpretable machine learning models for real-world applications?"**

To answer this central question, the thesis tackles the following challenges.

1. **Propose a robust time series classification pipeline**: design and implement time series classification pipeline employing state-of-the-art ML models suited for temporal data. This pipeline serves as the foundational component of the proposed framework, providing a reliable approach to processing, training, and evaluating models on diverse time series datasets.

2. **Propose a time series explainable AI evaluation methodology**: propose and validate a novel methodology to quantitatively assess the interpretability of time series models using XAI methods. This involves adapting and extending existing XAI evaluation metrics, such as sanity checks, faithfulness, sensitivity, robustness, stability, and localization, and coupling them with similarity measures such as DTW, MSE, MAE, MAE, RMSE, Euclidean distance. The resulting methodology will provide objective, standardized criteria for evaluating the quality and reliability of explanations in time-series contexts.

3. **Integrate classification and explainability into a unified framework**: synthesize the time series classification pipeline with the newly developed XAI evaluation methodology to form the Unified Time Series Framework for Explainable Artificial Intelligence (UTS-XAI). This integrated solution aims to achieve both high model accuracy and meaningful interpretability, providing a framework for building, explaining, and validating time series models in real-world applications.

4. **Propose the Global Interpretable Clustering**: by proposing a new enhanced visualization method to visualize and compare different XAI methods in the time series domain, we apply dimensionality reduction techniques (e.g., PCA, t-SNE, UMAP) to the generated explanations and group similar interpretability patterns together, enabling exploration, understanding, and assessing the quality of model explanations at scale.

1.2 Contributions

This thesis makes three key contributions toward advancing Explainable Artificial Intelligence in the context of time-series data:

1. The Unified Time Series Framework for Explainable Artificial Intelligence (UTS-XAI): we introduce a unified framework that incorporates explainability into the core phases of the time series classification modeling process. In doing so, it provides a structured environment where the production of explanations and the verification of their quality are natural, integral steps in model development.

2. A Novel Methodology for Quantitative Assessment of XAI Methods in the Time Series Domain: addressing the scarcity of standardized evaluation metrics suitable for time-series data, we develop a novel quantitative evaluation methodology. Identifying AI metrics, such as sanity, faithfulness, sensitivity, robustness, stability, and localization, to the temporal domain, and integrating various measures of similarity that are appropriate for time series data (e.g., Dynamic Time Warping (DTW), Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean

Squared Error (RMSE), euclidean distance (ED), and cosine similarity (CS)). By establishing criteria that account for temporal dependencies and evolving feature importance, our methodology enables rigorous, reproducible assessments of how faithfully explainable AI methods capture model reasoning in time series applications.

3. We propose Global Interpretable Clustering (GIC), a qualitative clustering methodology to visualize and interpret the feature importance maps generated by explainable AI methods. Using dimensionality reduction techniques such as Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP), our approach highlights latent structures and similarities in explanation patterns, providing intuitive visual summaries. The clustering may help to compare different explainable AI methods.

1.2.1 The Unified Time Series Framework for Explainable Artificial Intelligence (UTS-XAI)

While traditional ML pipelines focus on model training and performance evaluation, our UTS-XAI framework integrates explainability at traditional classification pipeline enabling that the interpretability of model predictions is both meaningful and quantifiable.

Figure 1.1 outlines the core components of UTS-XAI. Similarly to a conventional classification pipeline, time series data are sourced, pre-processed, and split into training and testing subsets (e.g., using cross-validation). The model is trained on the resulting training segments and subsequently applied to the test segments, generating predictions that are evaluated using standard classification metrics (e.g., accuracy). However, the UTS-XAI methodology extends this conventional workflow by incorporating XAI-driven evaluation criteria and visualization tools to assess and enhance the interpretability of the resulting predictions.

Once the ML model provides predictions, an XAI method (e.g., SHAP) is em-

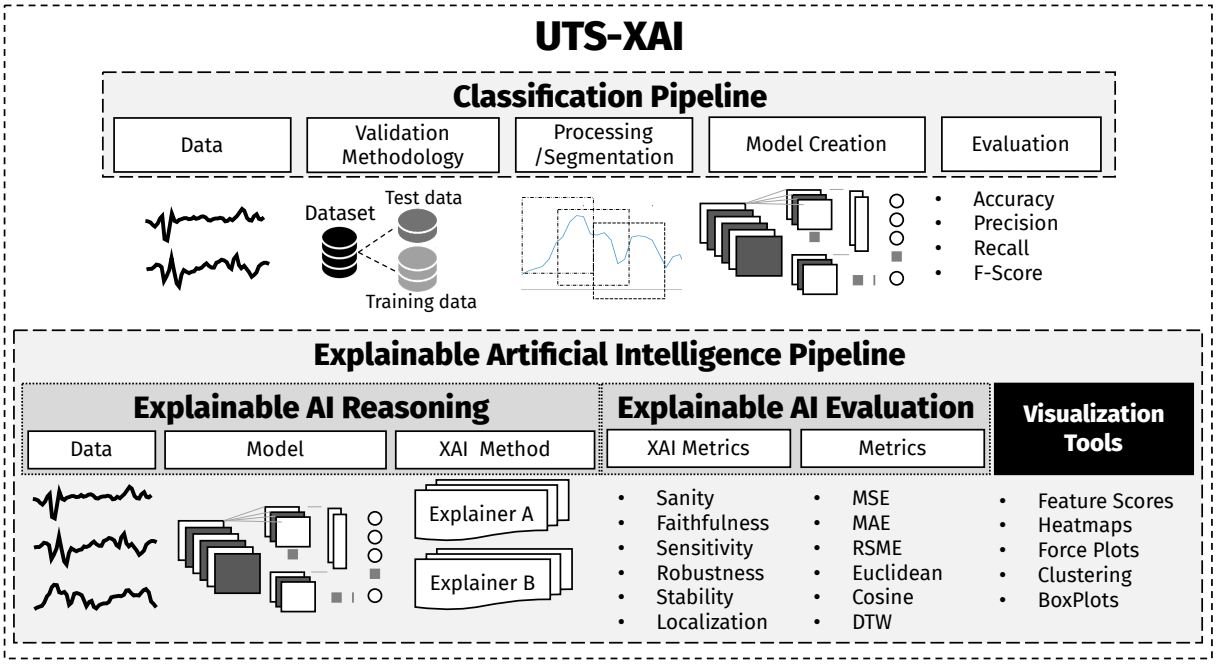


Figure 1.1 – The Unified Time Series Framework for Explainable Artificial Intelligence (UTS-XAI) integrates a traditional time series classification pipeline with an advanced explainability pipeline.

ployed to derive interpretable explanations on a per-sample basis. These explanations are typically expressed as feature importance maps (scores), indicating which segments of the time series most influenced the model's decision. Beyond this, our approach augments the explanation process by applying domain-specific XAI metrics, such as sanity, faithfulness, sensitivity, robustness, stability, and localization, and time series similarity measures (e.g., MSE, MAE, RMSE, Euclidean distance, and cosine similarity) to objectively quantify the quality of explanations. The final stage involves presenting the interpretability results through user-friendly visualizations that are meaningful in a time-series context.

By integrating interpretability into the entire classification pipeline, the UTS-XAI addresses a fundamental need: the ability to rigorously and objectively evaluate explanation methods for time series data. Now, explanation methods are not only visually appealing but also robust, reliable, and mathematically sound, thereby reducing the reliance on purely qualitative assessments and better supporting critical decision-making scenarios where incorrect attributions can have significant consequences.

1.2.2 A Novel Methodology for Quantitative Assessment of XAI Methods in the Time Series Domain

We propose a novel Explainable AI methodology for evaluating interpretable XAI methods specifically for the time series domain. As can be seen in Figure 3.3, we use more appropriate distance metrics for the time series that allow us to evaluate and compare each XAI evaluation method.

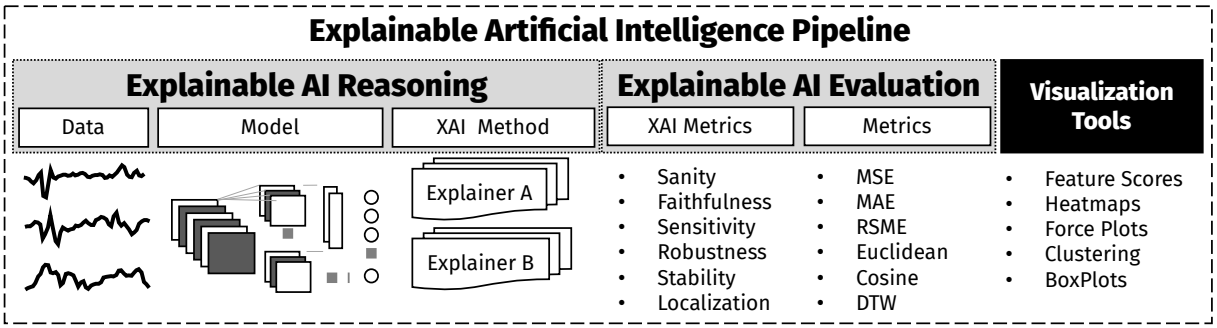


Figure 1.2 – The novel Explainable AI methodology for evaluating interpretable XAI methods specifically for the time series domain, integrating model explanations, quantitative metrics, and intuitive visualizations.

The proposed XAI methodology begins with a trained time-series classification model and its associated data. Next, an XAI technique, such as SHAP or LIME, is applied to derive local and global explanations, translating complex model decisions into interpretable outputs. To rigorously assess the quality of these explanations, we employ XAI specific metrics such as faithfulness, sensitivity, and robustness, each providing a different perspective of how well the explanations align with the model's underlying logic. In parallel, we incorporate existing visualization techniques suitable for time-series data.

1.2.3 Global Interpretable Clustering

We introduce a novel methodology, Global Interpretable Clustering, that provides a qualitative assessment of explainable AI methods by grouping and visualizing patterns of feature importance, as illustrated in Figure 1.3. By applying dimensionality reduction techniques (e.g. PCA, t-SNE, and UMAP) to feature importance maps generated by ex-

plainable AI methods, thisw methodology visually conveys the consistency, coherence, and divergence of their interpretations.

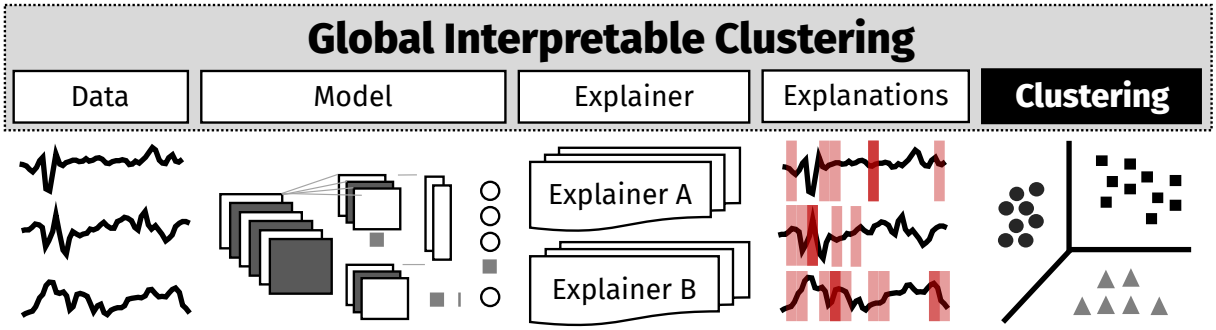


Figure 1.3 – Our proposed Global Interpretable Clustering methodology can reveals patterns of feature importance across different explainable AI methods, enabling a qualitative comparison of their consistency and ability to capture meaningful relationships within the data.

This clustering-based method represents a key contribution to the field of XAI. Unlike traditional quantitative metrics, which often focus on a single aspect of interpretability, GIC provides a more holistic, human-centric perspective. By examining how feature importance patterns cluster and overlap, we can readily discern which XAI techniques produce stable, trustworthy explanations and which may struggle to capture critical relationships in the data.

To our knowledge, this form of qualitative evaluation has not previously been explored. As a result, it offers a novel lens through better understanding, comparison, and refinement of explainability methods, ultimately enhancing the reliability and practical utility of AI-driven decision support.

1.3 Publications Arising from this Work

This section overviews the journal and conference publications that form this thesis, outlining the distribution of work among the authors and their respective contributions.

1. Bragança, H., Colonna, J. G., Oliveira, H. A., & Souto, E. (2022). How Validation Methodology Influences Human Activity Recognition Mobile Systems. Sensors, 22(6), 2360.

2. Bragança, H., & Souto, E. (2025). Unified Time Series Framework for Explainable Artificial Intelligence (2025). (In Production).

3. Bragança, H., & Souto, E. (2025). Explainability for Time Series Data (2025). (In Production).

1.4 Other Publications

Additionally, I contributed to the following publications. These publications are not directly linked to this thesis but were developed in our research group (Emerging Technologies and Systems Security).

1. Bragança H., Rocha, V., Souto, E., Feitosa E., Kreutz D. Explaining the Effectiveness of Machine Learning in Malware Detection: Insights from Explainable AI. In: Simpósio Brasileiro de Segurança da Informação E De Sistemas Computacionais (SBSEG), 2023.

2. Bragança, H., Rocha, V., Barcellos, L., Souto, E., Feitosa E., Kreutz D. Capturing Android Malware with MH-100K: A Novel and Multidimensional Dataset. In: Simpósio Brasileiro de Segurança da Informação E De Sistemas Computacionais (SBSEG), 2023.

3. Rocha, V. Assolin J., Braganca H., Kreutz D., Feitosa E. AMGenerator e AMExplorer: Geração de Metadados e Construção de Datasets Android. In: Simpósio Brasileiro de Segurança da Informação E De Sistemas Computacionais (SBSEG), 2023.

4. Andrade, C., Bragança, H., Feitosa, E., & Souto, E. (2023). Android malware detection with MH-100K: An innovative dataset for advanced research. Data in Brief, (Under Review).

5. Bragança, H., Rocha, V., Barcellos, L., Souto, E., Kreutz, D., & Feitosa, E. (2023). Android malware detection with MH-100K: An innovative dataset for advanced research. Data in Brief, 109750.

6. Nellessen, P., Bragança, H., E., & Souto, E. (2023). Leveraging Knowledge Distillation for Efficient Human Activity Recognition on Wearable Devices. Pervasive and Mobile Computing. (Under Review).

7. Andrade, C., Bragança, H., Fernandes, H., Feitosa, E., & Souto, E. (2022). Continuous Authentication using Operational System Performance Counters. Computers & Security. (Under Review).

8. Paz, I.; Bragança, H.; Souto, E. Autenticação Contínua Usando Sensores Inerciais dos Smartphones e Aprendizagem Profunda. In: Simpósio Brasileiro de Segurança da Informação E De Sistemas Computacionais (SBSEG), 2022, Santa Maria. Porto Alegre: Sociedade Brasileira de Computação, 2022. p. 209-222.

9. Lima, W. S., Bragança, H. L., & Souto, E. J. (2021). NOHAR-NOvelty discrete data stream for Human Activity Recognition based on smartphones with inertial sensors. Expert Systems with Applications, 166, 114093.

10. Bragança, H., Colonna, J. G., Lima, W. S., & Souto, E. (2020). A smartphone lightweight method for human activity recognition based on information theory. Sensors, 20(7), 1856.

1.5 Document Structure

The remainder of this thesis is organized into the following chapters:

- **Chapter 2** — This chapter introduces essential concepts and theoretical foundations underlying time-series classification and explainable AI. It reviews the standard pipelines for time series classification, covering data pre-processing, feature engineering, and model training, and present the traditional XAI workflow, emphasizing the techniques most relevant to machine learning interpretability. The chapter also highlights the current limitations and challenges in applying XAI to time series data and concludes by identifying research opportunities that motivate the methodologies proposed in this work. This chapter also provides a review

of the literature related to Explainable AI methods applied in the context of time series data. It categorizes existing XAI techniques, focusing on their application to time-series tasks and discussing their strengths, limitations, and relevance to the problem at hand.

- **Chapter 3** — This chapter presents our proposal - The Unified Time Series Framework for Explainable Artificial Intelligence (UTS-XAI). It explains the conceptual framework developed to integrate traditional time series classification tasks with explainable AI methods. The chapter details the design and rationale behind the framework, the XAI evaluation techniques incorporated, and how they interact with the time series classification process.

- **Chapter 4** — This chapter starts with the Experimental Protocol (Section 4.1 which outlines the experimental setup used to test the proposed framework. We explain how results will be conducted by presenting the evaluation scenarios, grouped in three parts: Generating classification models, evaluation the GIC method and and Explainable AI method evaluation. Further, it presents details of the datasets, the performance metrics adopted to assess model performance, and the XAI-specific metrics employed to measure interpretability quality. Finally, we present the UTS-XAI evaluation results for each evaluation scenario (Section 4.2).

- **Chapter 5** — The conclusion presented in this chapter highlights the primary contributions of this thesis and offers a perspective on prospective research possibilities. It examines how the proposed methodologies address the research question, enhance the state-of-the-art, and delineate the principal conclusions. It also examines the wider implications of the work, along with existing limitations and prospective research opportunities.

2

EXPLAINABLE ARTIFICIAL INTELLIGENCE FOR TIME SERIES DATA

This chapter provides a foundation for understanding the proposed framework by explaining the essential concepts and methods that guide our research. We begin by describing the standard machine learning pipeline for time series classification, focusing on its key components and constraints. Following that, we look at recent developments in Explainable AI pipelines, stressing their importance in enriching and improving the classification process by offering transparency and interpretability. Finally, we investigate previous studies in the field, examining their contributions and limits, before concluding with a discussion to contextualize our framework within the existing research community.

2.1 It is All About Time

Think about how important it is to closely watch the heart rate of a patient using electrocardiogram signals. Each heartbeat is recorded in real time, resulting in a time series in which doctors detect irregularities and diagnose conditions quickly (e.g., arrhythmia). Time-series data are essential for several scientific and practical applications, allowing researchers to analyze events over time and derive important conclusions.

To find patterns in time-series data, existing machine learning algorithms can learn from data without predefined assumptions about how that data is generated or structured (SARKER, 2021). Categorizing time-dependent data using patterns found within the series is an important part of machine learning's time-series classification process. In machine learning, time-series classification involves assigning labels to time-dependent data based on patterns within the series (FAWAZ et al., 2019). These models must account for the inherent temporal dependencies in which observations at one point in time are influenced by past events.

Healthcare, banking, and industrial monitoring are just a few areas that might benefit from the prediction capabilities made possible by time-series ML models. In the next section, we will introduce the methodology used to find patterns and classify time series.

2.2 The Time Series Classification Pipeline

Time series classification (TSC), as illustrated in Figure 2.1, is a systematic approach to labeling ordered data by capturing patterns and temporal dependencies. Each observation in a time series is chronologically ordered, necessitating methods that consider both point-wise information and the relationships among data across time. TSC has been used in healthcare, finance, and security, where large volumes of temporal data must be analyzed to allow decisions grounded in quantifiable evidence.

Developing TSC models commonly involves several steps (SHOAIB et al., 2015; LIMA et al., 2018; WANG et al., 2019; DEHGHANI; GLATARD; SHIHAB, 2019; BRAGANçA et al., 2020; DANG et al., 2020; FERRARI et al., 2021). First, data acquisition and preprocessing address issues such as noise, missing values, and outliers through procedures such as interpolation, filtering, or normalization. Second, feature engineering can focus on extracting domain-specific characteristics (for instance, statistical measures or spectral features), although deep learning methods often learn representations directly from raw time series. Third, model selection and training might involve a range of algorithms, including distance-based methods (for example, Dynamic Time Warping),

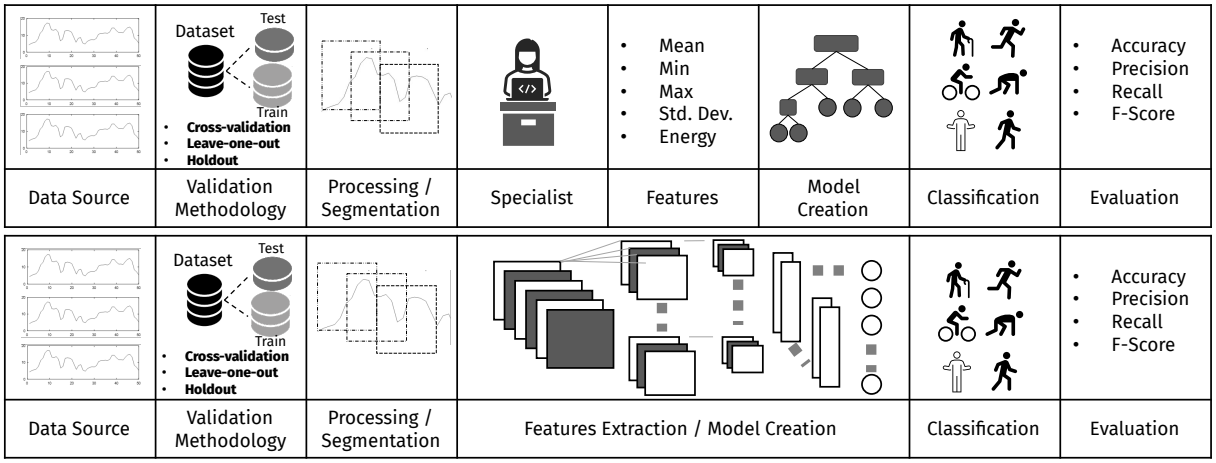


Figure 2.1 – The common methodology used in the time series classification task: data source (acquisition), validation methodology, segmentation, feature extraction, model creation, classification and evaluation.

traditional machine learning approaches (e.g. random forests), or deep architectures (e.g. convolutional or recurrent networks). The choice of model depends on dataset size, computational constraints, and domain knowledge. The subsequent sections discuss each stage of this pipeline, focusing on methodological considerations and common challenges.

2.2.1 Data Acquisition

The data acquisition phase is responsible for collecting data from a source to capture temporal phenomena. A time series is formally defined as a sequence of values ordered in time, often originating from sensors or observations in diverse application domains. The unified notation for time series, adapted from works such as (SCHÄFER, 2016; BAGNALL et al., 2017; RUIZ et al., 2021), describes a multivariate time series as $X = [X^1, ..., X^H]$, where H is the number of input channels, $X^i = (x_1^i, ..., x_N^i) \in \mathbb{R}^T$ is an ordered set of real values, and N denotes the number of time steps. For $H = 1$, the series is considered *univariate*; otherwise, it is *multivariate*. Time series data inherently involve complex temporal dependencies, where individual time points are interdependent, linked through their progression over time.

The type of data collected is determined by the intended application and domain. Choosing appropriate sensors is guided by the nature of the phenomena under study

and the types of information necessary for analysis (SHOAIB et al., 2015; BRAGANçA et al., 2020; WANG et al., 2019). A frequent example involves gyroscopes and accelerometers in Human Activity Recognition (HAR) systems (BRAGANçA et al., 2020), where they capture movements such as vibrations, oscillations, rotations, or angular changes. Smartphones and wearables equipped with these sensors generate large volumes of time series data related to user behavior and interactions with the environment.

In other contexts, healthcare and environmental monitoring rely extensively on time series for continuous measurement. Electrocardiogram (ECG) sensors, for instance, record the electrical activity of the heart, providing streams of data that may reveal cardiac anomalies, such as arrhythmias.

Following data acquisition, the collected measurements often remain in a raw state. Although these raw time series can be rich in information, they typically require preprocessing to address issues related to noise, missing values, or data quality. The completeness and reliability of the acquired data have a direct impact on subsequent tasks such as feature extraction and model training, making a systematic acquisition strategy an important foundation for any time series classification pipeline.

2.2.2 Splitting the Validation Data

Choosing a validation approach is an important component in machine learning model development, especially for models intended to generalize to new data. Before preprocessing or segmenting raw time series, it is conventional to define a validation procedure that reduces biases and avoids data leakage. A validation phase provides performance estimates aligned with the model's generalization capabilities rather than the memorization of training patterns. Models without an appropriate validation strategy may overfit, causing performance metrics that do not hold when applied to other data (BRAGANÇA et al., 2022). In addition, validation protocols address data leakage, a situation in which test set information unwittingly contaminates the training process and leads to inflated performance measures.

Several methodologies exist for splitting a dataset, including the holdout method,

k-fold cross-validation, and leave-one-subject-out (LOSO) cross-validation. The holdout method divides the dataset into training and test subsets, yielding a single estimate of performance. This approach is computationally efficient but can produce a pessimistic assessment because the model is trained on fewer samples; moreover, estimates may depend strongly on how the data are split (ARLOT; CELISSE et al., 2010; KOHAVI et al., 1995; GHOLAMIANGONABADI; KISELOV; GROLINGER, 2020). By contrast, k-fold cross-validation systematically partitions the dataset into k folds of approximately equal size. Each fold serves once as the test set, while the remaining $k - 1$ folds act as the training set. This procedure, although more computationally expensive, often yields more reliable estimates of model performance (ARLOT; CELISSE et al., 2010; DUDA; HART; STORK, 2000; WONG, 2015).

In settings with multiple subjects, leave-one-subject-out (LOSO) cross-validation trains on data from all but one subject and tests on the held-out subject (GHOLAMIANGONABADI; KISELOV; GROLINGER, 2020). This approach is used to assess whether a model can handle variability introduced by new, previously unseen individuals. However, LOSO may generate higher variance in performance estimates when the number of subjects is small (KOHAVI et al., 1995; WONG, 2015). The selection of a validation method typically depends on dataset size, computational constraints, and the context in which the model is applied.

2.2.3 Preprocessing

Pre-processing is a preparatory phase in time-series data analysis that addresses quality-related issues to facilitate reliable model training. Time-series datasets frequently contain noise, missing values, anomalies, and inconsistent sampling rates, all of which can influence the performance of machine learning models. Through filtering and formatting, pre-processing refines the raw data, which can contribute to enhanced model interpretability and more stable predictions.

Noise is a common concern in time-series data. Sensor faults, environmental disruptions, and network interruptions are typical sources of high-frequency fluctua-

tions that may obscure underlying signals. Filtering methods, such as low-pass filters, reduce these short-term variations while preserving slower trends. Moving average filters, which smooth data over a specified time window, help highlight periodic or cyclical patterns that might otherwise be concealed.

Normalization is another principal step, especially in settings that combine data from multiple sensors or measurement devices. Variations in scales and units can lead to inconsistent feature representations and hinder model performance. Z-score normalization, one widely used method, subtracts the mean and divides by the standard deviation of each feature, thus centering and scaling the data. This transformation aligns all variables more closely and aids in constructing models that weigh features comparably.

Inconsistent sampling intervals also affect time series analyses. Sensors may operate at different frequencies or experience intermittent recording, resulting in misaligned time steps. Resampling techniques, including interpolation or downsampling, convert the data to a uniform sampling rate, allowing comparisons or joint analyses among multiple time series. Maintaining a consistent time axis can improve downstream tasks and simplify the modeling pipeline.

Pre-processing thus tackles noise, scale discrepancies, and sampling irregularities that can undermine classification or regression models. Data that undergo these transformations frequently exhibit lower variance due to external factors and provide a clearer representation of the phenomena under investigation. After completing preprocessing, the segmentation phase typically follows. This subsequent step partitions continuous time series into discrete segments that represent distinct events or patterns of interest, as discussed in the next section.

2.2.4 Segmentation

Segmentation divides continuous time series data into smaller intervals, facilitating the examination of local behaviors and simplifying subsequent analysis. The division of a continuous signal into segments can be achieved through various techniques,

depending on the characteristics of the data and the objectives of the study. Common segmentation methods include fixed-size segmentation, sliding-window segmentation, event-based segmentation, and adaptive segmentation.

Fixed-size segmentation partitions the time series into segments of equal duration. This straightforward approach is often applied when the phenomena under investigation occur at regular intervals, such as daily, weekly, or seasonal cycles. Although computationally efficient, fixed-size segmentation may not adequately capture events or changes that occur at irregular intervals.

Sliding-window segmentation employs a fixed-length window that moves along the time series, often with a predetermined amount of overlap between successive windows. Overlapping windows can provide a more continuous representation of the temporal dynamics and may capture transient phenomena that non-overlapping segments could miss. This technique is commonly applied in fields such as human activity recognition, where the boundaries of the activities are not clearly defined.

Event-based segmentation, also known as change-point detection, partitions the time series based on detected shifts in its statistical properties. In this method, various techniques—such as statistical tests, Bayesian methods, or machine learning approaches—are used to identify points in time where the characteristics of the data (e.g., mean, variance, or autocorrelation) change abruptly. Algorithms including cumulative sum, hidden Markov models, and kernel-based methods are among the approaches employed.

Adaptive segmentation methods modify the segment length according to the local characteristics of the time series. By employing multi-resolution analysis or dynamic windowing, these methods adjust the segmentation criteria to reflect variations in the complexity or volatility of the data. Adaptive approaches are applied in domains such as biomedical signal processing, where the duration of relevant events can vary over time.

The selection of a segmentation method depends on the specific application, the nature of the data, and computational considerations. Each approach offers distinct advantages and may be more appropriate for certain types of temporal patterns or events.

Segmentation produces a structured representation of the time series that facilitates subsequent processing steps, such as feature extraction and model development. In the following section, the feature extraction process is discussed, which transforms segmented data into suitable representations for machine learning.

2.2.5 Feature Extraction Process

Feature extraction is a fundamental phase in the development of machine learning models for time-series data. In this phase, raw signals are transformed into a compact representation by selecting or constructing features that capture the underlying temporal patterns and relationships. This transformation facilitates the subsequent modeling task by reducing data dimensionality and computational overhead while retaining essential information.

Two principal approaches to feature extraction have been developed: hand-crafted feature extraction and automatic feature extraction using deep learning models, such as Convolutional Neural Networks (CNN).

Hand-crafted feature extraction involves the manual design of features based on expert knowledge of the application domain (SHOAIB et al., 2015; SHOAIB et al., 2014; LI et al., 2018; ANGUITA et al., 2013; LIMA et al., 2018; BRAGANçA et al., 2020). In time series analysis, commonly used hand-crafted features include statistical metrics (e.g., mean, variance, skewness), frequency components derived from Fourier or wavelet transforms, and autocorrelation measures. For example, in electrocardiogram (ECG) signal analysis, features such as heart rate, QRS duration, and amplitude characteristics are used to characterize cardiac function. Hand-crafted features may be grouped into several domains. Time-domain features, computed directly from the data, are computationally efficient and straightforward to interpret. Frequency-domain features, obtained through methods such as Fourier or wavelet analysis, help capture periodicities and oscillatory behavior. Symbolic-domain features convert continuous time series data into sequences of discrete symbols, thereby compressing the data and facilitating pattern recognition (LIMA et al., 2018; BRAGANçA et al., 2020). While hand-crafted

features benefit from transparency and low computational cost, their design can be labor-intensive and may miss subtle or complex data structures that are not immediately evident to experts.

Automatic feature extraction is performed by deep learning architectures, notably CNNs, which learn hierarchical representations directly from raw time series data (NWEKE et al., 2018). Through the application of convolutional filters, these networks capture low-level patterns in the early layers and progressively build more abstract representations in deeper layers. In effect, CNNs combine the processes of feature extraction and classification, which may allow for the modeling of intricate, nonlinear relationships within the data. Despite their potential, automatic feature extraction techniques generally require larger datasets and greater computational resources. Moreover, the abstract nature of the features generated by deep models can complicate the interpretation of model decisions in settings where transparency is desired.

Both hand-crafted and automatic feature extraction methods offer distinct advantages and limitations. Hand-crafted features provide a clear and computationally efficient representation, but depend on domain expertise and may overlook nuanced patterns. In contrast, automatic feature extraction via deep learning can capture complex relationships without explicit feature design, albeit at the cost of increased data and computational demands, as well as reduced interpretability. The choice between these approaches depends on the specific requirements of the application, the available resources, and the desired balance between model performance and transparency.

2.2.6 Classification

Classification is a supervised learning task that maps input data to predefined categorical labels. In this context, the objective is to assign a discrete label (e.g., "spam" versus "not spam") based on patterns derived from previously labeled data (dataset). The process begins with a labeled dataset and involves training a model to recognize relationships between the input features and their corresponding categories, so that the model can generalize its predictions to new, unseen inputs.

The classification process typically follows several stages. Initially, the data undergoes preliminary operations such as splitting into training and test subsets, preprocessing, segmentation, and feature extraction. Following these steps, a classification algorithm is chosen according to the nature of the data and the specific requirements of the task. Common algorithms include decision trees, support vector machines, and various forms of deep neural networks. After training, the model is evaluated using standard metrics such as accuracy, precision, recall, and F1-score, which provide quantitative measures of its performance.

Traditional classification methods are designed for static data in which the features are independent and unordered. In contrast, time series classification addresses sequential data where temporal order and dependency are significant. For example, time series classification may be applied to distinguish between normal and abnormal electrocardiogram signals or to identify human activities based on motion sensor data. The sequential structure of time series data requires specialized preprocessing, feature extraction, and model architectures, such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs), that can accommodate temporal dependencies.

Time series classification tasks typically commence with a labeled dataset. Formally, the time series classification defines a mapping $X \rightarrow y$ that minimizes error in a dataset $D = \{(X_1, y_1), ..., (X_N, y_N)\}$, where N is the number of data samples, $X \in D$ is a time series, $y_i \in \mathbb{R}^C$ denotes the one-hot vector of a class label to which the input belongs, and C is the number of classes (BAGNALL et al., 2017; FAWAZ et al., 2019; RUIZ et al., 2021). In time series segmentation, we search $X \rightarrow Y$ that maps an input sample to a dense classification $Y = [y_1, ..., y_N] \in \mathbb{R}^{C \times N}$, i.e., a class label is predicted for each time step.

In the training phase, the ML model learns a mapping from the extracted features to the target labels. Conventional machine learning approaches rely on hand-crafted features derived from domain knowledge, such as statistical measures, frequency components, or other descriptive attributes—to represent the data. These features are then used with classifiers such as logistic regression, decision trees, or support vector machines. Alternatively, deep learning models, such as CNNs and RNNs, perform auto-

matic feature extraction directly from raw time series data. The training of deep models involves propagating input through multiple layers of nonlinear transformations and updating model parameters using optimization algorithms (e.g., stochastic gradient descent or Adam) based on a loss function such as categorical cross-entropy.

In the classification phase, the trained model is applied to a new, unlabeled time series X_u. The model processes X_u to generate a feature representation and subsequently assigns a predicted label y_u This assignment can be performed using one of two approaches:

- Similarity measures: the feature representation of X_u is compared with those of the training examples using metrics such as cosine similarity, and the label of the most similar time series is assigned.
- Direct prediction: the model outputs a probability distribution over the classes, and the label corresponding to the highest probability is selected.

The evaluation of the model is carried out on a separate test set, using metrics such as accuracy, precision, recall, and F1 score. In binary classification tasks, the area under the ROC curve (AUC-ROC) may also be employed. In cases of class imbalance, techniques such as resampling or class weighting are applied during training to adjust for unequal representation of the classes.

Overall, the classification process, especially when applied to time series data, comprises a sequence of methodical steps, from data preparation and feature extraction to model training and evaluation, that collectively aim to produce models capable of robust generalization to new observations. The next section will introduce the most common evaluation metrics used to estimate a classifier's future performance.

2.2.7 Evaluation Metrics for Classification

The performance of a classification model is typically measured using a range of quantitative metrics that reflect its predictive accuracy and error characteristics (ARLOT; CELISSE et al., 2010; WONG, 2015). Commonly used metrics include accuracy, recall

(sensitivity), specificity, precision, and the F1-score (BULLING; BLANKE; SCHIELE, 2014; LIMA et al., 2019).

Accuracy is defined as the proportion of correct predictions among the total number of predictions. In datasets with balanced classes, accuracy provides a direct measure of performance. However, in situations characterized by class imbalance—where one class predominates—a model that exclusively predicts the majority class may yield a high accuracy rate while inadequately identifying instances from minority classes. In these instances, precision and recall offer a more nuanced evaluation. Precision quantifies the ratio of true positive predictions to all positive predictions, and is pertinent in applications where the cost of false positives is high. Recall, in contrast, measures the ratio of true positive predictions to all actual positive cases, which is important when failing to detect a positive instance has substantial consequences. The F1-score, computed as the harmonic mean of precision and recall, provides a single metric that balances these two aspects.

Table 4.4 summarizes these evaluation metrics, presenting their respective equations and brief descriptions to serve as a reference for their computation and interpretation.

Table 2.1 – Summarization of accuracy, recall, precision and F-measure. TP means true positives, TN true negatives, FP false positives and FN means false negatives.

Metric	Equation	Description
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Accuracy is the ratio of correct predictions divided by the total predictions.
Precision	$\frac{TP}{TP+FP}$	Precision is the ratio of true positives and total positives predicted.
Recall	$\frac{TP}{TP+FN}$	Recall is the ratio of true positives to all the positives in ground truth.
F Measure	$2 \times \frac{Precision \times Recall}{Precision+Recall}$	The F-measure is the harmonic mean of precision and recall.

The evaluation phase concludes the traditional classification pipeline, but additional considerations often follow, especially in high-stakes domains such as healthcare,

finance, or legal systems. In these cases, the interpretability and transparency of the model become significant. While simpler models, such as decision trees and logistic regression, offer a direct representation of decision rules, more complex models often operate as opaque systems.

The subsequent sections examine recent developments aimed at enhancing model transparency and reliability, thereby complementing the evaluation of predictive performance with assessments of model comprehensibility.

2.2.8 Explainable Algorithms for Interpretable and Transparent Systems

In many applications, the decision-making process of a machine learning model must be transparent, interpretable, and justifiable. Explainable machine learning methods provide mechanisms for users to understand and assess model predictions. This requirement is prominent in domains such as healthcare, finance, and legal systems, where opaque decisions can lead to ethical, social, or operational complications (LINARDATOS; PAPASTEFANOPOULOS; KOTSIANTIS, 2021).

The need for explainability originates in the early stages of model development. Decisions made during data collection, preprocessing, and feature engineering reflect the expertise and perspectives of system designers (e.g., machine learning engineers, data scientists). Such decisions may introduce biases or overlook variations in user contexts (VEALE; BINNS, 2017; BRAGANÇA et al., 2022). Explainability frameworks allow for the examination of these factors by identifying sensitive attributes or decision rules and by elucidating the contribution of specific features to classification outcomes. Addressing these aspects is relevant not only from an ethical standpoint but also to maintain user trust, as perceived inaccuracies or biases may diminish confidence in the system (YIN; VAUGHAN; WALLACH, 2019; TOREINI et al., 2020; ALIKHADEMI et al., 2021).

Traditional machine learning models, more specifically complex architectures, such as deep neural networks, are often opaque and operate as "black boxes" that pro-

duce predictions without accompanying explanations. This lack of transparency poses challenges in regulatory contexts, such as compliance with the "right to explanation" under the General Data Protection Regulation (GDPR) and limits the effectiveness of user-centered, iterative development processes.

Explainable Artificial Intelligence (XAI) addresses these challenges by developing methods and tools that make machine learning models more interpretable and open to examination. Although XAI techniques are increasingly adopted, they are frequently applied as standalone components within conventional machine learning pipelines, resulting in workflows that are not fully integrated.

Incorporating XAI methods into traditional classification frameworks responds to the need to strike a balance between technical performance and social responsibility. By embedding XAI throughout the model development lifecycle, machine learning systems can be adapted to meet both regulatory and ethical requirements, thus improving overall trust and reducing barriers to adoption. Thus, a dedicated XAI pipeline represents an advance toward more interpretable, transparent, and accessible artificial intelligence systems.

2.3 The Explainable Artificial Intelligence

Explainable Artificial Intelligence is a specialized field of artificial intelligence dedicated to elucidating the inner workings of machine learning models, particularly those often referred to as "black boxes." As AI systems become increasingly complex and pervasive, their lack of transparency poses significant challenges, especially in high-stakes domains such as healthcare.

Two related yet distinct concepts in XAI are *interpretability* and *explainability* (DOSHI-VELEZ; KIM, 2017; LIPTON, 2018; GUIDOTTI et al., 2018; LINARDATOS; PAPASTEFANOPOULOS; KOTSIANTIS, 2021). In this context, interpretability refers to the extent to which a human can discern the factors that influence a model's output. This notion is concerned with the observable relationships between inputs and outputs, thereby facilitating an understanding of the cause-and-effect dynamics that underlie

the model's predictions. In contrast, explainability addresses the degree to which the internal logic and structure of a model are accessible for analysis. It involves providing detailed accounts of the computational processes and representations that drive decision-making. A model is often described as a black box if its internal operations are either undisclosed or too complex to be readily comprehended (GUIDOTTI et al., 2018); in such cases, measures of interpretability and explainability serve as proxies for understanding its behavior.

XAI methods aim to furnish representations of model behavior that are comprehensible to non-expert stakeholders, thereby supporting validation, error analysis, and compliance with regulatory standards. These methods are commonly divided into two categories based on when and how interpretability is achieved:

- Intrinsic Explainability: This approach is applicable to models that are inherently interpretable by design, such as linear regression, decision trees, or rule-based systems. The architecture of these models facilitates a direct mapping between inputs and outputs that can be readily examined.

- Post-Hoc Explainability: In this case, interpretability is attained after the model has been trained. External techniques are applied to provide explanations of the model's behavior, which is particularly useful for complex models (e.g., deep neural networks) that are not interpretable by design.

Furthermore, XAI methods are often characterized by the scope of the explanation they provide (SHEU; PARDESHI, 2022; ROJAT et al., 2021; THEISSLER et al., 2022; DAS; RAD, 2020; CHADDAD et al., 2023):

- Global Explanations: these methods attempt to describe the overall decision-making process of a model across an entire dataset. Global explanations offer a broad perspective on the model's operational logic, although obtaining such comprehensive interpretations is challenging for highly complex models.

- Local Explanations: these techniques focus on explaining individual predictions by isolating the factors that contribute to a specific output. Local explanations are

particularly useful for case-by-case analysis and for providing instance-specific justifications.

- Model-Specific Methods: these techniques are tailored to particular model architectures and exploit specific properties of the model to generate explanations.
- Model-Agnostic Methods: these approaches are applicable to any model regardless of its internal structure, enabling a uniform explanation framework across different model types.

2.4 The Traditional Explainable AI Pipeline

The increasing adoption of AI models for time series classification underscores the need for transparency and interpretability. In applications with significant consequences, understanding the basis for model predictions is essential. Figure 2.2 presents the Explainable AI Reasoning Pipeline for time series classification tasks. This framework provides systematic methods for analyzing, interpreting, and visualizing model decisions, thereby bridging the gap between complex AI systems and human understanding.

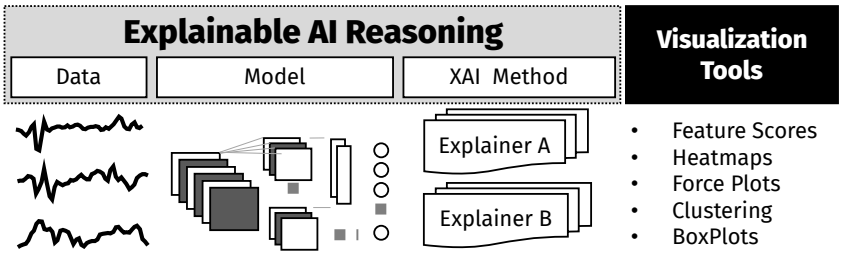


Figure 2.2 – The Explainable AI Reasoning Pipeline for time series classification tasks, consisting of three core phases—Data, Model, and XAI Method, followed by visualization tools for interpretability.

The Explainable AI Reasoning Pipeline comprises four core phases:

- Data phase: this phase addresses the raw input data, typically represented as time series signals (e.g., waveforms). It involves capturing time-dependent features from one or multiple sources or channels, followed by preprocessing and transformation techniques designed to prepare the data for subsequent modeling.

- Model phase: preprocessed time series data are fed into the AI model during this phase. Models often utilize deep learning architectures such as convolutional neural networks or recurrent neural networks to extract temporal features, identify patterns, and perform classification tasks based on the input sequences.
- XAI methods phase: once the model generates predictions, post-hoc explainability techniques are applied to interpret the outcomes. Denoted here as Explainer A and Explainer B, these methods analyze the model's internal mechanisms and highlight specific features or time segments that have a significant influence on the predictions.
- Visualization tools phase: To effectively communicate the explanations, various visualization techniques are employed, including:
  - *Feature Scores:* graphical representations that indicate the importance of specific time features or intervals.
  - *Heatmaps:* visual depictions of regions within the time series that contribute substantially to the model's decisions.
  - *Force Plots:* illustrations that convey the impact of individual features on the model's predictions.
  - *Clustering:* grouping of similar patterns to facilitate interpretability.
  - *Boxplots:* Statistical summaries that detail the distribution of feature contributions.

2.4.1 Explainable AI Reasoning

The Explainable AI Reasoning component focuses on generating interpretations for the predictions made by machine learning models. The pipeline begins with the preprocessing of raw data, which involves cleaning, normalization, and feature engineering to prepare the input for the model. In time series classification, this step often includes segmenting data into meaningful windows, handling missing values, and applying

transformations such as Fourier or wavelet decompositions to highlight relevant temporal patterns. Once preprocessed, the data is fed into the model phase, where machine learning or deep learning architectures are employed. These models may be designed to capture temporal dependencies, extract complex features, and perform robust time series classification.

Once the predictive model is trained and validated, Explainable AI (XAI) methods are applied to interpret its outcomes. The model processes the input data and generates predictions. The goal of the Explainable AI Reasoning stage is to provide information into how the model reaches its predictions by employing various explainability techniques.

Two of the most prominent and widely adopted XAI methods are LIME (RIBEIRO; SINGH; GUESTRIN, 2016b) and SHAP (LUNDBERG; LEE, 2017), widely used to understand the rationale behind classifier predictions, which address different aspects of explainability. (UDDIN; SOYLU, 2021; BETTINI; CIVITARESE; FIORI, 2021; DAS et al., 2021; ROY et al., 2021; BRAGANÇA et al., 2022). These methods help uncover the underlying mechanisms behind the model's predictions, identifying which features, time intervals, or patterns are most influential.

Local Interpretable Model-Agnostic Explanations (LIME) (RIBEIRO; SINGH; GUESTRIN, 2016b) provides local interpretability by creating simpler surrogate models (e.g., linear models or decision trees) around the specific instance being explained. These surrogate models approximate the behavior of the original model near a particular data point, offering information into why the model made a specific prediction for that instance.

SHapley Additive exPlanations (SHAP) (LUNDBERG; LEE, 2017) provides both global and local explanations by calculating Shapley values based on cooperative game theory. Each feature is assigned a numerical score representing its contribution to the final prediction, offering consistent and fair attributions, making SHAP a widely used tool for understanding feature importance across the entire dataset.

In the context of time series classification, SHAP helps to identify the most critical time windows or segments that influence the model's prediction; quantify the impact

of each input feature or time point on the overall decision, aiding domain experts in validating the model's behavior; and Enable comparisons across different time series instances to understand common or unique patterns influencing outcomes.

In addition to SHAP and LIME, attention mechanisms and saliency maps provide further interpretability classification tasks such as attention mechanisms and Saliency Maps. Attention mechanisms is commonly used in transformer-based architectures, attention weights provide an intuitive explanation of which time steps are most important during the model's decision-making process. Saliency maps highlight the most influential time steps by computing gradients with respect to the input features, revealing which parts of the time series had the greatest impact on the predictions.

2.4.2 Visualization Tools for Explainable AI methods

To enhance the interpretability of XAI explanations, the workflow integrates visualization tools that present the outputs of explainability methods in intuitive formats:

Heatmaps are widely used to visualize the importance of time steps or intervals. They represent the contributions of input features across time in a color-coded format, enabling practitioners to identify which temporal regions most influenced the model's predictions. Heatmaps are particularly effective in highlighting patterns in high-dimensional time series data.

Boxplots provide a statistical summary of data, showcasing the distribution, variability, and outliers within the importance scores. This method is useful for comparing how the influence of specific features or time windows varies across multiple predictions, facilitating an understanding of model consistency.

Dimensionality reduction strategies such as PCA (Principal Component Analysis), t-SNE (t-Distributed Stochastic Neighbor Embedding), and UMAP (Uniform Manifold Approximation and Projection) are employed to reduce the dimensionality of time series data and visualize relationships between instances. PCA projects the data into lower-dimensional linear spaces, while t-SNE and UMAP are non-linear techniques that preserve local and global structures, respectively. In addition, clustering methods

are used to group similar time series patterns together, making it possible to identify common trends, outliers, or anomalies in the data. This clustering is beneficial for exploring how the model behaves across different types of time series patterns.

Combining these methods allows domain specialists to acquire granular and global insights about model behavior. Heatmaps and boxplots make temporal analysis easier, while clustering approaches highlight overall structures and correlations in the data. Together, these visualizations improve the interpretability of XAI approaches, making them critical for verifying, debugging, and deploying time series classification models in real-world settings.

2.5 Related Works

Although significant progress has been made in the evaluation of explanation methods, much of the existing research has predominantly focused on the image domain. Efforts to adapt and extend these evaluation frameworks to time series data are still emerging. Studies such as SCHLEGEL et al. (2019), LOEFFLER et al. (2022), SERRAMAZZA et al. (2023), BAER et al. (2025), and KNOF; BOERGER; TCHOLTCHEV (2024) represent promising advances that address challenges specific to temporal data. Nonetheless, significant gaps remain in the systematic adoption of comprehensive evaluation frameworks that incorporate appropriate metrics tailored for the unique complexities of time series Explainable Artificial Intelligence (XAI).

An overview of these studies, including the authors, explanation techniques, evaluation metrics, datasets, models, and similarity measures, is summarized in Table 2.2. State-of-the-art research in XAI evaluation, despite its groundbreaking nature, exhibits several limitations that our proposed UTS-XAI Framework seeks to overcome.

Current evaluation approaches often rely on a narrow set of metrics that may not capture the full scope of interpretability or are not adequately adapted for time series data. For instance, CEREKCI et al. quantitatively evaluates saliency methods for mammogram analysis using primarily the Pointing Game Score, providing a valuable but limited perspective. Similarly, ADEBAYO et al. and YEH et al. propose sanity

Table 2.2 – Summarization of related works, which present the authors, the XAI methods used in each study, the appropriate metrics for evaluating XAI, the datasets and models, and finally the similarity metrics.

Author	XAI Methods	XAI Metrics	Datasets	Models	Similarity Metrics
LOEFFLER et al. (2022)	GradCam Guided GradCam Gradient Guided Backprop. SmoothGrads LRP Kernel-SHAP LIME Int. Gradients Random Baseline	Sanity Faithfulness Sensitivity Robustness Stability Localization	GunPointAgeSpan FordA FordB MelbournePedestrian NATOPS ElectricDevices	U-Time bi-LSTM FCN TCN	SSIM DTW
ADEBAYO et al. (2018)	Gradient Gradient-SG Gradient-Input GradCAM Guided BackProp Guided GradCAM Integrated Gradients Integrated Gradients-SG	Sanity	ImageNet Fashion MNIST MNIST	Inception v3 CNN MLP	Spearman rank corr. (abs/no-abs) SSIM Pearson corr. HOGs corr.
YEH et al. (2019)	Smooth-Grad Integrated Gradients Guided Back-Propagation KernelSHAP	Fidelity Sensitivity	MNIST Cifar-10 ImageNet	Linear SVM Neural Network Random Forest Logistic Regression CNN ResNet	L_2 norm
MELIS; JAAKKOLA (2018)	Saliency Occlusion LRP Kernel-SHAP LIME Int. Gradients Random Baseline	Robustness	UCI datasets (Glass, Wine, Ionosphere, Leukemia) COMPAS MNIST ImageNet	Linear SVM Neural Network Random Forest Logistic Regression CNN ResNet	Euclidean norm (L_2)
SCHLEGEL et al. (2019)	Saliency LRP DeepLift LIME SHAP	Perturbation Analysis Sequence Evaluation	FordA FordB ElectricDevices MelbournePedestrian ChlorineConcentration Earthquakes NonInvasiveFetalECGThorax1 NonInvasiveFetalECGThorax2 Strawberry MIT-BIH Arrhythmia	CNN RNN ResNet-based (Paper Models)	Mean
CEREKCI et al. (2024)	Grad-CAM Grad-CAM++ Eigen-CAM	Pointing Game Score	Mammogram Dataset	ResNet50	N/A
PAWLICKI et al. (2024)	Anchors LIME SHAP Integrated Gradients	Faithfulness Robustness Localization Complexity Randomization Axiomatic	CIC IoT 2023 CSE-CIC-IDS2018	Neural Network (PyTorch) Random ForestClassifier CartModel Sequential (TensorFlow)	Pearson Correlation Coefficient Spearman's Rank Correlation Coefficient
HEDSTRÖM et al. (2023)	Saliency Integrated Gradients SmoothGrad Guided Backpropagation GradCAM Guided-GradCAM LRP LIME Kernel SHAP	Faithfulness Robustness Localization Complexity Randomization Axiomatic	ImageNet GunPoint AgeSpan, FordA, FordB, ElectricDevices, Melbourne Pedestrian, NATOPS)	U-Time bi-LSTM Fully Convolutional Network (FCN) Temporal Convolutional Network (TCN)	SSIM Cosine Similarity Dynamic Time Warping (DTW)
SCHLEGEL; KEIM (2023)	Saliency Integrated Gradients DeepLift Occlusion GradientShap DeepLift Shap KernelShap	Perturbation Analysis Skewness of Attributions Class Distributions Number of Perturbed Values Needed	UCR benchmark datasets (FordA, FordB, ElectricDevices)	CNN	Euclidean Distance Cosine Distance
SOKOL; VOGT (2024)	LIME LIMEtree Counterfactuals Post-hoc methods Ante-hoc methods Attribution-based methods	Fidelity, Accuracy Consistency, Comprehensibility Correctness, Completeness Continuity, Contrastivity Covariate Complexity, Compactness Compositionality, Confidence Context, Coherence Controllability, Informativeness Acceptability, Usability Task Performance, Trust Correctability	N/A (conceptual paper)	N/A (conceptual paper)	N/A
SERRAMAZZA et al. (2023)	SHAP dCAM Ridge Classifier Random	Precision Recall F1-score PR-AUC ROC-AUC Explanation Power	Synthetic (Pseudo Periodic, Gaussian, Auto Regressive) Real-world (Counter Movement Jump, Military Press)	ROCKET dResNet Ridge Classifier	N/A (Min-Max Normalization used)
BAER et al. (2025)	Gradients (GR) Integrated Gradients (IG) Feature Occlusion (FO)	Degradation Score (DS) Normalized AUC-PR (AUC-PR')	Synthetic Time Series (Level shifts, Gaussian pulses, Sine waves, Local trends, Amplitude contrast, Length contrast)	ResNet InceptionTime	Spearman Correlation
DEMBINSKY et al. (2025)	Feature Attributions (FAs) Concept Explanations (CEs) Example Explanations (ExEs) White-Box Surrogates (WBSs) Natural Language Explanations (NLEs)	Parsimony Plausibility Coverage Fidelity Continuity Consistency Efficiency	N/A (conceptual paper)	N/A (conceptual paper)	N/A
KNOF; BOERGER; TCHOLTCHEV (2024)	LRP SHAP	Truthfulness Analysis (Accuracy Drop) Stability Analysis (Frobenius-Norm) Consistency Analysis (Top-k Agreement)	PTB-XL Benchmarking Dataset (ECG)	CNN	Frobenius-Norm
Our Work (UTS-XAI)	LIME TabularExplainer SHAP Explainer SHAP TreeExplainer Gradient	Sanity Faithfulness Sensitivity Robustness Stability Localization	MIT-BIH Arrhythmia MIT-BIH Supraventricular INCART 12-lead	DeepConvLSTM FCN XGBoost	MSE MAE RMSE Variance SSIM DTW Euclidean Cosine

checks and measures of (in)fidelity and sensitivity, but their experimental focus is largely on image-centric datasets, limiting direct applicability to time series. MELIS; JAAKKOLA introduces metrics to quantify the robustness of explanations based on Lipschitz estimates, demonstrating instability in current methods, particularly for non-temporal data.

Emerging efforts in time series XAI evaluation, such as SCHLEGEL et al. and LOEFFLER et al., begin to incorporate temporal dimensions. SCHLEGEL et al. introduces new perturbation-based verification techniques for time series explanations, primarily measuring accuracy drop. LOEFFLER et al. proposes a framework of six orthogonal metrics (sanity, faithfulness, sensitivity, robustness, stability, and a novel localization metric) specifically for visual interpretations on time series, using metrics like SSIM and DTW to capture time-series specific qualities. However, these studies, while foundational, do not always provide a unified and adaptable pipeline for comprehensive evaluation across diverse models and explainers. SERRAMAZZA et al. evaluates SHAP and dCAM for Multivariate Time Series Classification (MTSC), revealing that simple adaptations of SHAP can outperform bespoke MTSC methods and highlighting the inadequacy of some synthetic datasets. KNOF; BOERGER; TCHOLTCHEV proposes a framework for MTSC with truthfulness, stability, and consistency analyses, demonstrating trade-offs between XAI quality criteria for ECG data. BAER et al. investigates class-dependent evaluation effects in time series attribution, finding contradictions between perturbation-based and ground truth metrics.

Furthermore, while comprehensive overviews such as PAWLICKI et al. and HEDSTRÖM et al. (Quantus) catalog numerous XAI metrics, and conceptual frameworks from SOKOL; VOGT and DEMBINSKY et al. provide structured approaches, they often do not delve into the specific adaptations and empirical validations required for robust evaluation within the time series domain.

Our UTS-XAI Framework addresses these limitations by extending standard XAI evaluation metrics (faithfulness, robustness, sensitivity, stability, localization, sanity) to the temporal domain, and combining them with time series–specific similarity and distance measures (e.g. MSE, DTW). Metrics such as DTW are advantageous for time

series data, as they effectively capture sequence alignment, a fundamental aspect of temporal dependency analysis, as highlighted by LOEFFLER et al. and SCHLEGEL; KEIM. This multidimensional assessment provides a detailed and robust evaluation of interpretability techniques in temporal contexts. Our evaluations span diverse model architectures representing both deep learning methods and traditional boosting approaches. This provides comprehensive coverage across algorithmic paradigms and demonstrates the adaptability of our framework to a wide range of applications and architectures.

Current approaches often rely on image-centric datasets (e.g., ImageNet, Fashion MNIST, MNIST, CIFAR-10) or generic UCI datasets, which limits the applicability of their findings to the complex structure of real-world time series data. Some studies, such as SCHLEGEL et al. and LOEFFLER et al., utilize UCR time series datasets, and SERRAMAZZA et al. and BAER et al. use synthetic time series, the latter also notes the limitations of such synthetic benchmarks for time series analysis. Our work distinguishes itself by employing real-world medical datasets directly relevant to healthcare applications and underscore the practical significance of our contributions.

Our framework not only highlights the importance of XAI in healthcare but also offers a blueprint for advancing explainability in other domains with temporal data, paving the way for future research and practical applications. Our framework stands out from existing works because of its integrated approach, which covers a broad range of methods, metrics, models, and datasets. We address the limitations of current methodologies and adapt our approach to the unique challenges of time series data.

3

UTS-XAI — Unified Time Series Framework for Explainable Artificial Intelligence

In this thesis, we propose the Unified Time Series Framework for Explainable Artificial Intelligence (UTS-XAI), designed to enhance the evaluation and interpretability of time series classification models that enable the user to (1) build, train and evaluate machine learning models for time series domain, (2) understand the reasons behind model decisions using different explainable IA method, and (3) evaluate explanations using metrics specifics to explainable AI domain.

While the classification components largely follow conventional practices (BRAGANÇA et al., 2022), the novelty lies in embedding explainability directly into the modeling cycle. UTS-XAI employs XAI techniques (e.g., SHAP or LIME) to interpret model decisions and integrates a dedicated evaluation layer that goes beyond conventional metrics such as accuracy and recall. This approach aims to meet the growing demand for transparency and trust in time series applications.

3.1 UTS-XAI Overview

The proposed UTS-XAI framework integrates a traditional time series classification pipeline with an advanced explainability layer, as illustrated in Figure 3.1. The classifi-

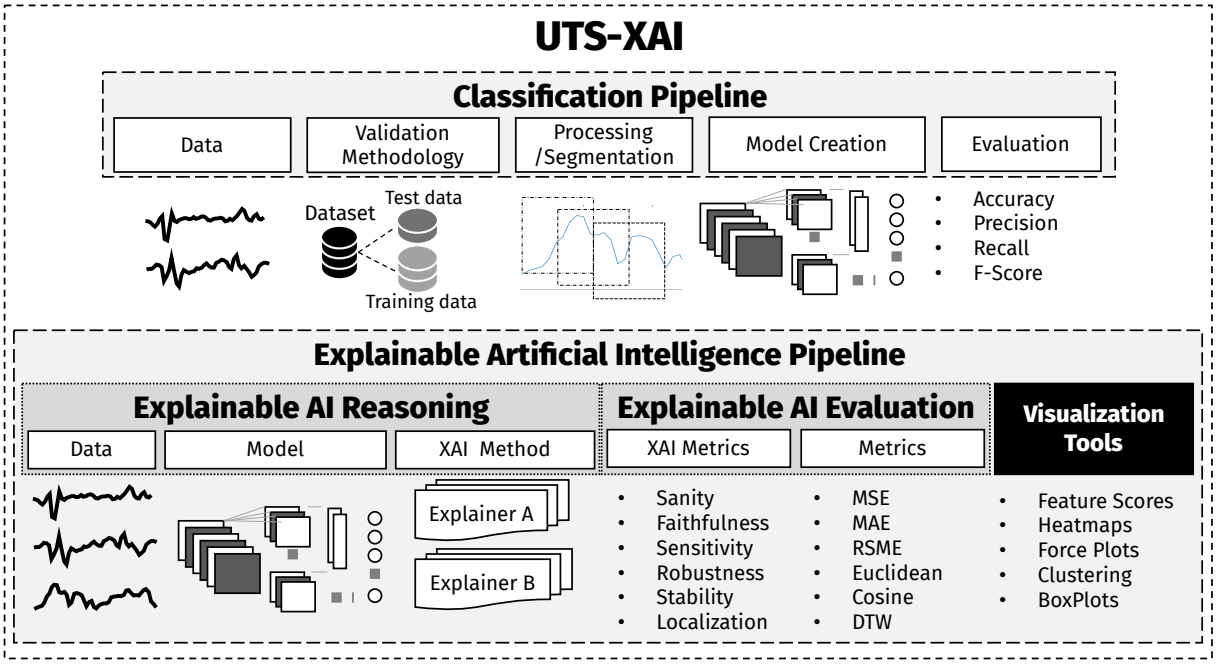


Figure 3.1 – Unified Time Series Framework for Explainable Artificial Intelligence (UTS-XAI). The UTS-XAI framework integrates a traditional time series classification pipeline with an recent and advanced explainability pipeline. UTS-XAI aims to enhance model interpretability and reliability in time series classification.

cation pipeline encompasses standard steps such as data acquisition, preprocessing and segmentation, model creation, validation methodology, and performance evaluation. The explainability pipeline is composed of reasoning modules based on state-of-the-art XAI methods, assessed by adapted evaluation metrics such as *faithfulness* and *robustness*, and supported by visualization tools including heatmaps, boxplots, and our proposed Global Interpretable Clustering (GIC).

3.2 The Classification Pipeline for Time Series Data

Figure 3.2 illustrates the UTS-XAI classification pipeline. The initial steps of this pipeline were first introduced in (BRAGANÇA et al., 2022). The classification pipeline begins with a data source, followed by a validation methodology to split the dataset into training and test sets. The segmentation stage processes time series data to enhance pattern recognition. During model creation, machine learning models are trained on the segmented data. Finally, the evaluation stage evaluates the performance of the

models using metrics such as accuracy, precision, recall, and F1-score. The details of this pipeline are discussed in Chapter 2.

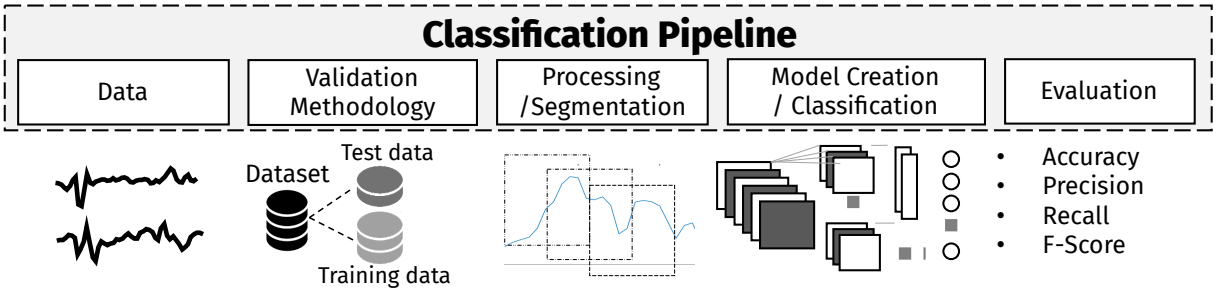


Figure 3.2 – Overview of UTS-XAI classification pipeline.

In machine learning-based systems, achieving high classification accuracy is important, but it is not sufficient on its own. Integrating an explainable AI pipeline after the classification process is no longer optional, it is a necessity to provide transparency and trust into model predictions. In the next section, we discuss the primary motivations for integrating an Explainable AI pipeline into the classification workflow.

3.3 The Explainable AI Pipeline

Modern machine learning models, particularly deep learning, are frequently regarded as "black boxes" because of their complex architectures. Although these models may yield remarkable results, their weakness in interpretability may limit user trust and acceptance. An XAI pipeline mitigates this issue by explaining the reasoning behind a model's decision, emphasizing the important input features that influenced the result.

At the core of our framework is an XAI pipeline. The pipeline employs an explainable IA reasoning module, which interact with the machine learning models to generate explanations via feature importance maps. The explanations lead to a better understanding of the model, which in turn enables the diagnosis of model flaws and suggest potential refinement strategies.

Figure 3.3 illustrates the UTS-XAI Explainable AI pipeline. This pipeline is built upon the same dataset and trained model used in the classification process. The XAI pipeline focuses on generating explanations for model predictions, assessing their

quality, and presenting interpretable results to users. In the UTS-XAI framework, the pipeline is divided into two components: Explainable AI Reasoning and Explainable AI Evaluation.

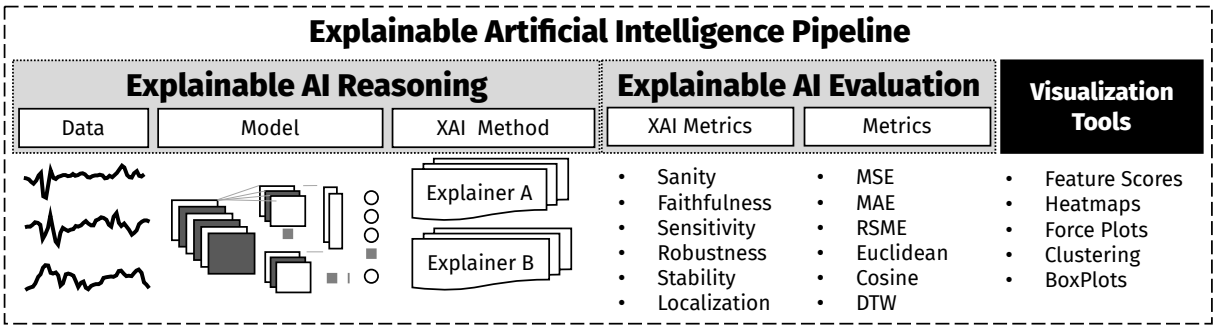


Figure 3.3 – Overview of our novel XAI evaluation methodology for time series classification.

The **Explainable AI Reasoning** component focuses on generating explanations for model predictions. This involves applying XAI methods, such as SHAP and LIME, directly to trained models to uncover their decision-making processes. These methods identify the time series segments most influential in determining the model's output, offering both global and local interpretability. By revealing which features or data segments are most important to predictions, explainable AI reasoning bridges the gap between complex machine learning models and human comprehension.

The **Explainable AI Evaluation** component introduces a formal mechanism to assess the quality of the generated explanations. It employs qualitative and quantitative metrics, such as faithfulness, robustness, sensitivity, and localization, to provide accurate explanations and reflect the underlying model behavior.

Unlike traditional approaches that are based only on visual inspections, this step uses rigorous criteria to evaluate the reliability and consistency of the explanations. Additional performance metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Euclidean Distance, and Dynamic Time Warping (DTW), are incorporated to align interpretability with model performance. In the next section, we will present in more detail the XAI pipeline proposed in this research.

3.4 Explainable AI Reasoning

The Explainable AI Reasoning component is dedicated to understanding how machine learning models make decisions. Explainable AI methods are at the center of this component as they improve model interpretability, assist in debugging by identifying areas where the model may be misdirected, and reveal important features for specific predictions. Feature importance maps visualizations act as a bridge, reducing the gap between complex machine learning processes and human understanding.

The Figure 3.4 illustrates the workflow of our XAI evaluation pipeline for time series classification. It begins with the raw time-series data retrieved from a database, which is then segmented and fed into a trained model (e.g., neural network or XGBoost). The model outputs a class probability vector, such as [0.9, 0.1], indicating its prediction. A XAI explainer, represented here by SHAP, is then applied to compute importance scores for each time step. These scores are visually overlaid as a heatmap on the original signal, with darker hues highlighting intervals deemed more relevant to the model's decision. Finally, this importance map is quantitatively evaluated using several metrics (see Section 4.2.3). We illustrate a real-world application of our pipeline in Figure 3.5. Figure 3.5(a) overlays the feature-importance heatmap on the ECG waveform used for arrhythmia detection, where darker shaded regions highlight intervals deemed highly influential by the XAI method. Figure 3.5(b) displays the corresponding importance scores as a standalone time series, making it easier to track temporal attribution dynamics across the heart cycle.

The feature importance values can be derived through various explainable AI techniques, each offering distinct perspectives on how input features influence model predictions. In our framework, we employ three XAI methods that employ a different underlying mechanism and may produce different results, as shown in Figure 3.6. We present more details about saliency maps, Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) in the following sections.

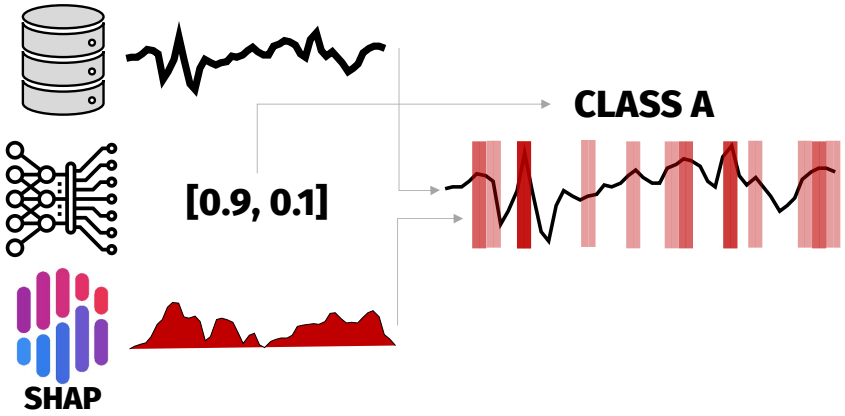
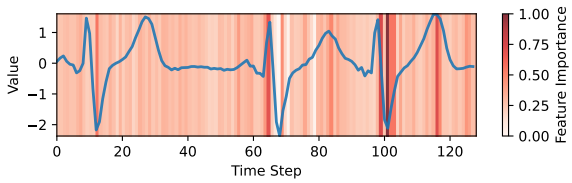
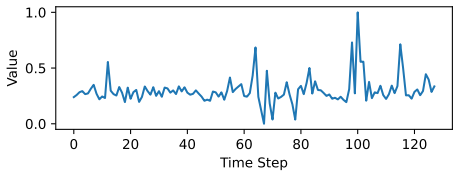


Figure 3.4 – XAI workflow: the time-series data is fed into a trained model. A XAI explainer is then applied to compute importance scores for each time step. These scores are visually overlaid as a heatmap on the original signal, with darker hues highlighting intervals deemed more relevant to the model's decision.



(a) Overlay of the feature-importance heatmap on the ECG signal.

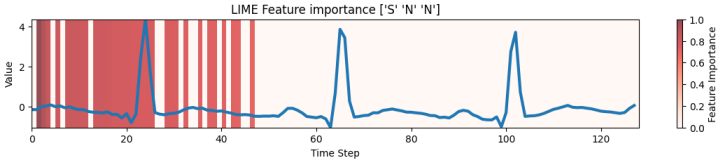


(b) Feature importance values as a standalone time series.

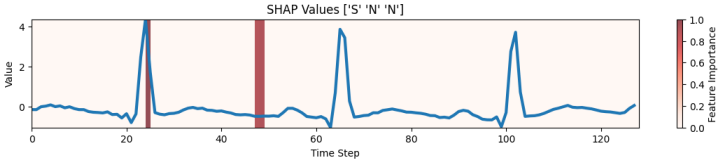
Figure 3.5 – (a) ECG signal (blue curve) with an overlaid feature-importance heatmap (red shades), indicating saliency levels at each time step. (b) Time series plot of importance scores derived from the XAI explainer for the same ECG segment, allowing precise inspection of attribution fluctuations.

3.4.1 Saliency Maps

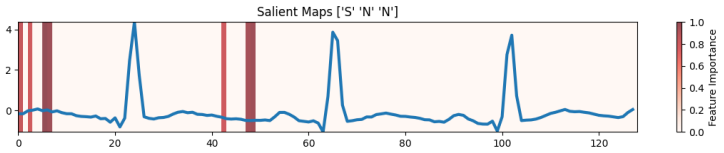
Saliency methods are a popular class of methods designed to visualize and interpret the internal workings of convolutional neural networks, including the generation of saliency maps (SIMONYAN; VEDALDI; ZISSERMAN, 2013). It has been widely adopted and extended, and variations of saliency maps have become a standard interpretability tool in numerous domains beyond image data, including time series and natural language processing. The saliency maps algorithm's core lies in the calculation of gradients. These gradients, which relate to the loss in relation to the input tensor, reveal how modifications to each input value could impact the loss. This is an important observation for determining the input regions that are of utmost importance for model decision-making process.



(a) Local Interpretable Model-Agnostic Explanations (LIME)



(b) SHapley Additive exPlanations (SHAP)



(c) Saliency Map

Figure 3.6 – Comparison of XAI methods LIME, SHAP, and Saliency for the same instance and model.

The Algorithm 3.1 is specifically designed for generating saliency maps. The central component of this process is the $GradientTape$ mechanism (TensorFlow framework), which precisely logs operations to enable automatic differentiation. Observing $input$, the algorithm prepares to compute gradients with respect to the input of the model. When the input tensor is given to the model, a prediction is generated in inference mode. This means that layers such as dropout or batch normalization are implemented consistently during model evaluation rather than training. After making a prediction, the algorithm calculates the loss by comparing the actual labels with the model's predictions using categorical cross-entropy.

Algorithm 3.1 Saliency Map basic algorithm

procedure GENERATESALIENCYMAP($model, input, true\_label$)
    Initialize *GradientTape* as *tape*
    tape.watch($input\_tensor$)
    $prediction \leftarrow model(input, training = False)$
    $loss \leftarrow categorical\_crossentropy(true\_label, prediction)$
    $gradients \leftarrow tape.gradient(loss, input)$
    $saliency\_map \leftarrow reduce\_max(abs(gradients))$
    return $saliency\_map$
end procedure

When creating the saliency map, the algorithm goes beyond and calculates the

absolute values of these gradients. This guarantees that both positive and negative influences are taken into account equally. The map is enhanced by isolating the highest gradient value across the channels for each input value.

3.4.2 Local Interpretable Model-agnostic Explanations (LIME)

The Local Interpretable Model-Agnostic Explanations (LIME) (RIBEIRO; SINGH; GUESTRIN, 2016a) is a Explainabe AI technique designed to interpret complex models by explaining predictions for individual instances. It achieves this by approximating the behavior of a model locally around a specific data point using a simpler surrogate model.

The Algorithm 3.2 presents a procedure for generating explanations using LIME algorithm. The approach consists of two primary functions: one to create a LIME explainer and another to generate feature importance for a given instance.

Algorithm 3.2 LIME Explanation for Time Series Data

```
1: procedure LIME_EXPLANATION(data, model, input_data)
2:     explainer ← GETLIMEEXPLAINER(data)
3:     feature_importance ← GETLIMEFEATUREIMPORTANCE(explainer, model, input_data)
4:     return explainer, feature_importance
5: end procedure
6: procedure GETLIMEEXPLAINER(data)
7:     explainer ← LimeTabularExplainer(training_data)
8:     return explainer
9: end procedure
10: procedure GETLIMEFEATUREIMPORTANCE(explainer, model, input_data)
11:     feature_importance ← explainer.explain_instance(
    input_data, model, num_features)
12:     return feature_importance
13: end procedure
```

The first function, $GetLimeExplainer$, initializes a LIME Tabular Explainer. During initialization, the system generates feature names for each time step, capturing the sequential nature of time series data. The second function, $GetLimeFeatureImportance$, acts as a bridge between the theoretical model predictions and practical explanations. The LIME explainer generates explanations for predictions made on a given time series

input. This process involves performing a analysis in which the explainer utilizes a surrogate model specific to the local area to estimate the behavior of the complex underlying model near the input instance. The outcome is a set of feature importances that highlight points that exerted the most significant influence on the model's prediction.

3.4.3 SHapley Additive exPlanations

The Algorithm 3.3 shows a systematic approach for utilizing SHAP (SHapley Additive exPlanations) (LUNDBERG; LEE, 2017) to interpret the predictions generated by a machine learning model. The essence of this algorithm lies in its two main components: the creation of a SHAP explainer and the computation of SHAP values to explain the feature importances.

Algorithm 3.3 SHAP Explainer

procedure SHAP_EXPLANATION(model, data, input_data)
    $explainer \leftarrow$ GETSHAPEXPLAINER(model, data)
    $shap\_values \leftarrow$ GETFEATUREIMPORTANCE(explainer, input_data)
    return $explainer, shap\_values$
end procedure
procedure GETSHAPEXPLAINER(model, data)
    $explainer \leftarrow shap.Explainer(model,\ shap.sample(data, 100))$
    return $explainer$
end procedure
procedure GETFEATUREIMPORTANCE(explainer, input_data)
    $shap\_values \leftarrow explainer.shap\_values(input\_data)$
    return $shap\_values$
end procedure

The process of Algorithm 3.3 begins with the $GetShapExplainer$ function, which aims to create a SHAP explainer object tailored to the specific model and dataset being analyzed. This is achieved by utilizing the $Explainer$ class from the SHAP library. The explainer requires two inputs: the model's prediction function and a subset of the input data obtained through sampling. The sampling process, indicated by the code $shap.sample(data, 100)$, selects a representative subset of the entire dataset to facilitate the computation of SHAP values. Due to the significant computational complexity associated with calculating accurate SHAP values for the entire dataset, this step is

extremely important.

After initializing the SHAP explainer, the $GetFeatureImportance$ function computes the SHAP values for a particular input instance. These values quantify the influence that each feature in the input data has on the model's prediction, providing a evaluation of the significance of each feature. The calculation of SHAP values is not merely an estimation based on statistics or heuristics, but rather relies on the rigorous principle of Shapley values in cooperative game theory.

3.4.4 Feature Importance Normalization

Normalization is another important step applied to any XAI method to standardize the importance values across different explanations. It allows results from various models or methods to be comparable by rescaling the importance scores to a consistent range (e.g., ranging from 0 to 1). This process eliminates differences in scale that could otherwise affect the interpretability and analysis of the results.

The Algorithm 3.4 illustrates its application specifically for saliency maps. It begins by identifying the minimum and maximum values within the importance map, denoted as min_val and max_val, respectively. These values represent the lowest and highest intensities present in the importance map. The normalization process then adjusts each value in the importance map by subtracting the minimum value and dividing by the range, which is the difference between the maximum and minimum values.

Algorithm 3.4 Feature Importance Normalization

procedure NORMSALIENCYMAP($importance\_map$)
    $min\_val \leftarrow$ np.min($importance\_map$)
    $max\_val \leftarrow$ np.max($importance\_map$)
    $normalized\_map \leftarrow \frac{importance\_map - min\_val}{max\_val - min\_val}$
    return $normalized\_map$
end procedure

This operation effectively rescales the entire importance map so that the smallest value becomes 0 and the largest value becomes 1, with all other values proportionally adjusted within this range.

3.5 Explainable AI Evaluation

The Explainable AI Evaluation component is a fundamental extension of the Explainable AI Reasoning pipeline, designed to quantitatively assess the effectiveness and reliability of XAI methods. Its primary objective is to validate whether the explanations provided by XAI techniques accurately reflect the model's decision-making process, remain stable under varying conditions.

A key motivation for this evaluation process is the growing need for reliable interpretability in high-stakes AI applications, such as medical diagnostics, financial forecasting, and industrial monitoring, where incorrect or misleading explanations can have severe consequences. Without proper evaluation, XAI methods may produce seemingly valid explanations that fail under scrutiny, leading to potential misinterpretations and reduced trust in AI systems.

The Explainable AI Evaluation component is structured into two key categories of metrics, each serving a distinct role in assessing model explainability:

1. Explainable AI-specific metrics: these metrics are specifically designed to assess the quality and effectiveness of XAI-generated explanations. They evaluate whether the explanations.

    - Provide meaningful feature attributions that differ from random noise (Sanity Checks).
    - Faithfully reflect the model's internal reasoning (Faithfulness).
    - Remain stable and consistent across similar inputs (Stability).
    - React appropriately to changes in class distributions (Sensitivity).
    - Are resistant to adversarial perturbations or noisy inputs (Robustness).
    - Aligning importance maps with relevant regions (Localization).

2. General metrics for explanation comparison: in addition to XAI-specific metrics, the evaluation framework incorporates standard quantitative metrics to compare model predictions and feature importance maps. We present a short detail as follows.

- Classification metrics (Accuracy, Precision, Recall, F1-Score, AUC-ROC): used to verify whether explanations align with the model's predictive performance.
- Similarity metrics (Cosine Similarity, Structural Similarity Index (SSIM)): measure how similar the feature importance maps are across different XAI methods or input perturbations.
- Distance metrics (Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Euclidean Distance, Dynamic Time Warping (DTW)): quantify differences between explanations by measuring the distance between feature importance scores, ensuring that perturbation-based evaluations produce meaningful deviations in model predictions.

We present in Table 3.1 an overview of several evaluation metrics used to measure similarity and distance across importance maps, that are particularly relevant for time-series data analysis. It includes traditional error metrics such as MSE, MAE, and RMSE, each accompanied by their mathematical formulas. For these error metrics, higher values indicate greater prediction errors and, consequently, poorer model performance, while lower values suggest that the model's predictions are closer to the actual values, reflecting higher accuracy.

Table 3.1 – Summary of similarity and distance metrics used to compare importance maps.

Metric	**Formula**	**High Values Indicate**	**Low Values Indicate**
MSE	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	Large errors in predictions, low accuracy.	Better model performance, small prediction errors.
MAE	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$	Poor model performance with large deviations.	Accurate predictions, minimal deviation from actual values.
RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	Higher overall prediction errors.	Better predictions, smaller errors.
Euclidean Distance	$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$	Dissimilarity between two data points.	High similarity or closeness between data points.
Cosine Similarity	$\frac{\vec{A}\cdot\vec{B}}{\lvert\lvert\vec{A}\rvert\rvert\times\lvert\lvert\vec{B}\rvert\rvert}$	High similarity (closer to 1).	Low similarity (closer to 0 or -1).
DTW (Dynamic Time Warping)	$\min_{\pi\in\Pi}\sqrt{\sum_{(i,j)\in\pi}(x_i - y_j)^2}$	Greater dissimilarity between time series sequences.	Higher similarity in shape and timing between sequences.

In addition, Table 3.1 presents similarity measures such as Euclidean distance and cosine similarity. The Euclidean distance quantifies the direct dissimilarity between

two data points, with lower values indicating a closer resemblance, whereas cosine similarity assesses the angular similarity between vectors, where values closer to one denote high similarity. Furthermore, the table includes Dynamic Time Warping (DTW), a metric specifically designed for time series data. DTW is useful for aligning sequences that may vary in time or speed; here, higher DTW values indicate greater dissimilarity between sequences, while lower values suggest that the sequences are well aligned and similar in shape and timing. These metrics can be used to assess different aspects of the explanations. For example, error-based metrics can be used to measure the overall deviation between importance maps produced by different methods or under varying conditions.

3.5.1 Explainable AI-Specific Metrics for Interpretability

In this section, we present the metrics used by the UTS-XAI framework to evaluate the results of the interpretation methods. The evaluation metrics in XAI, such as sanity, faithfulness, sensitivity, robustness, stability, and localization, help bridge the gap between the opaque decision-making processes of complex models and human interpretability. The combination of these metrics improves transparency, aids in debugging and refining models, and ultimately contributes to the ethical deployment of AI systems in critical applications. In the following sections, we explore each metric in detail.

3.5.1.1 Sanity

When interpreting ML models, it is essential to verify that the generated explanations (e.g., feature importance maps) truly depend on the parameters learned from the model and are not simply products of the model's architecture or other artifacts. To this end, we improve and incorporate the sanity metric in our framework based on random labeling to confirm whether an explanation accurately reflects the internal reasoning of the model and are not mere artifacts of the model architecture. Specifically, we train two versions of the same model architecture: one on the correctly labeled dataset and another

on a dataset in which the labels have been randomly shuffled. If the explanation method genuinely relies on the learned data-label relationships, the resulting importance maps should differ substantially between these two models.

As shown in Figure 3.7, the sanity workflow begins by (1) training a model on a data set that retains its original correctly assigned labels. Once this model is fully trained, an explanation method is applied to the test set, often in the form of importance maps, and is applied to the test set, generating a baseline reference for comparison. The original training labels are then randomly shuffled (2), and an identical model architecture is retrained on this permuted dataset. Since the shuffled labels bear no meaningful relationship to the underlying features, this second model should not learn any substantive patterns. The same explanation technique is then applied to this new model, resulting in an additional set of saliency maps or feature-importance measures.

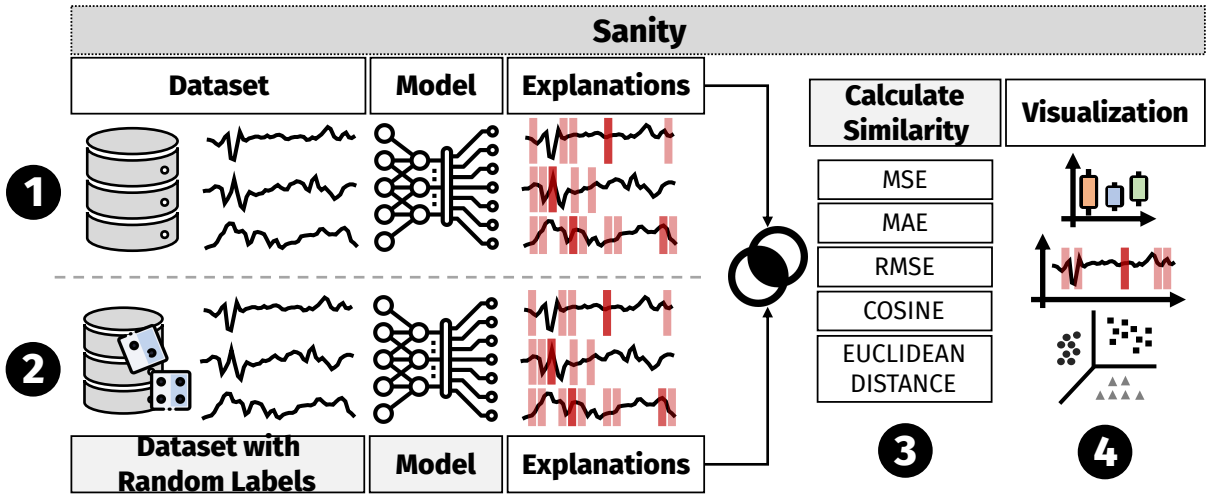


Figure 3.7 – Overview of the sanity metric workflow. A model is first trained on the correctly labeled dataset and used to generate a baseline saliency (or feature-importance) map. Next, the same model architecture is retrained on a randomly labeled dataset, and a second saliency map is produced. Finally, the two saliency maps are compared using similarity or distance metrics to determine whether the explanation method is genuinely sensitive to learned model parameters.

The final step involves a (3) quantitative comparison of the two sets of explanations using similarity or distance metrics and visualizing using a suitable tool (4). If the explanation method truly captures how the model learned parameters relate to the features, the importance maps for the correctly labeled model and the randomly labeled model should diverge. Conversely, if the explanations remain similar, it implies that

the method may be insensitive to the actual parameters learned by the model, failing the'sanity' criterion.

The data randomization test is the cornerstone of the sanity metric. Assesses the dependency of importance maps on the relationship between the data and labels by comparing saliency maps generated from a) a model trained with the original, correctly labeled dataset; b) a model trained with a dataset where the labels have been randomly permuted. A reliable saliency map method should produce markedly different maps for the two scenarios. If the maps remain similar despite randomization, this indicates that the explanation method is insensitive to the learned model parameters, undermining its utility for tasks such as debugging and interpretation.

3.5.1.2 Faithfulness

The faithfulness metric is used to evaluates whether the features identified as 'highly important' truly drive the predictions of the model. In other words, if an explanation highlights certain features, modifying or removing them should cause the model output to shift in a predictable way. Conversely, altering features marked as irrelevant should minimally affect the final prediction. By quantifying how the model responds when important features are ablated, we can assess whether the explanation genuinely captures the underlying mechanisms that drive the model's decisions.

We present one way to assess the faithfulness, as shown in Figure 3.8, which compares the original model predictions with those obtained under highly important feature perturbations.

First, baseline predictions are obtained by running the trained model on the unchanged dataset, and the corresponding explanations are generated (e.g., SHAP). These explanations serve as a starting reference point for evaluating whether the identified 'important' features truly influence the model's output. Next, we test the faithfulness by systematically ablate important features in the input data and then evaluate the effect on the model predictions. Feature ablation zeros out or removes only the features identified as most important. If ablating these supposedly key features triggers a substantial drop

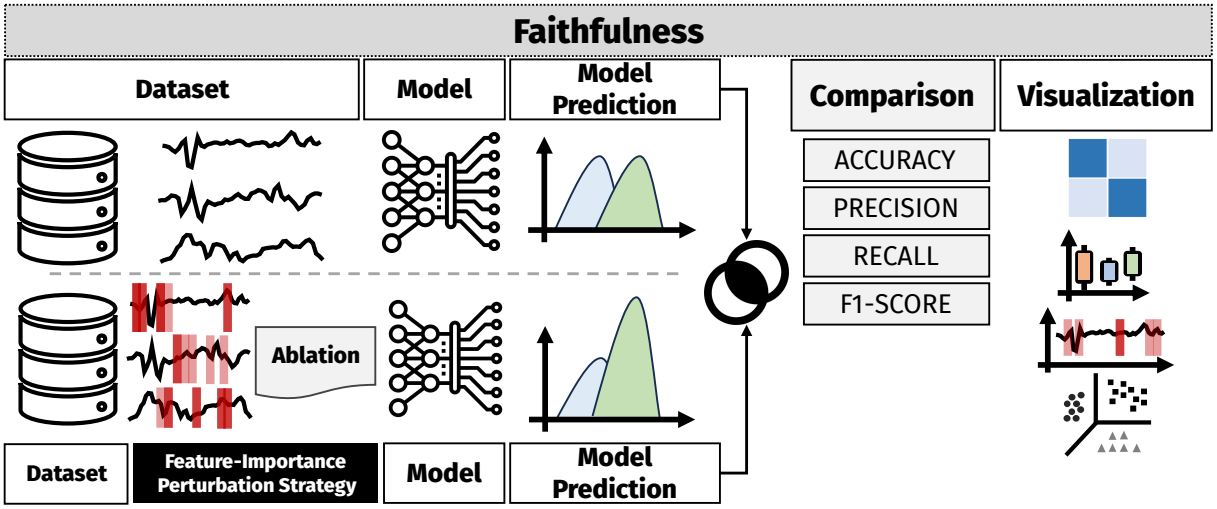


Figure 3.8 – Faithfulness evaluation workflow. Baseline explanations and predictions are first generated for the unaltered input data. Feature ablation in high influential features are then applied to create modified inputs, which the model processes to yield updated predictions. Finally, classification, similarity and distance metrics can be used to compare the altered outputs to the original baseline, revealing whether the explanation accurately reflects the model's true decision-making process.

in the model performance or leads to significant prediction changes, it supports the conclusion that the explanation has successfully captured the features that truly drive the model's decisions.

Once the perturbed inputs (high-influence features) are prepared, they are fed through the same trained model to produce new predictions. These altered predictions are then compared against the original baseline using a variety of similarity or distance metrics. Classification metrics (e.g. accuracy and recall) quantify changes in predictions, similarity metrics (e.g. MSE, MAE, RMSE) help quantify changes in the magnitude of predictions, whereas metrics such as cosine similarity or euclidean distance capture how the direction or spatial relationship of the output vector is altered.

Large deviations in predictions under ablation of high-importance features would indicate that the explanation correctly identified critical components of the input, i.e. removing them has a strong impact. Conversely, smaller or negligible changes when perturbing supposedly unimportant features would reinforce the reliability of the explanation. Repeating this procedure with different perturbation intensities can further demonstrate the robustness of the model, revealing whether it remains consistent or breaks down under certain conditions.

The faithfulness metric is of great value when assessing the interpretability of machine learning models. A model that consistently demonstrates minimal variation in predictions despite substantial alterations to important input features may be regarded as less interpretable, as it implies that the model's predictions are not strongly linked to the features identified as significant. However, a model that is greatly affected by these changes can be seen as more understandable, as it suggests a more distinct connection between the identified features and the model's predictions.

3.5.1.3 Sensitivity

We also include the sensitivity metric in our UTS-XAI framework because it provides a unique perspective on model interpretability by assessing how a model adjusts in response to variations between classes and individual samples. A model needs to identify relevant features for each of the classes to make a correct prediction. Essentially, a sensitive model not only distinguishes between different classes but also can tailors its explanations at a per-sample level, capturing the subtle distinctions that give each input its unique characteristics.

We present the sensitivity workflow in Figure 3.9. The class-level sensitivity workflow begins with a trained model and a dataset subdivided by class labels. For each class, feature-importance maps are produced, which capture how the model 'sees' and weighs the relevant features of the input samples. These maps are then aggregated on a per-class basis, providing a view of how the model attends to features for that specific class. Next, a variance is calculated for the importance maps within each class. These variances can be aggregated (e.g., via averaging) across all classes to yield an overall sensitivity score.

A higher variance means that the model's explanations diverge more when class labels change, implying that the model adapts its focus and highlights different important features for different classes. Consequently, class-level sensitivity offers a straightforward yet powerful means of determining whether a model truly distinguishes among various classes, rather than applying a uniform explanatory pattern across

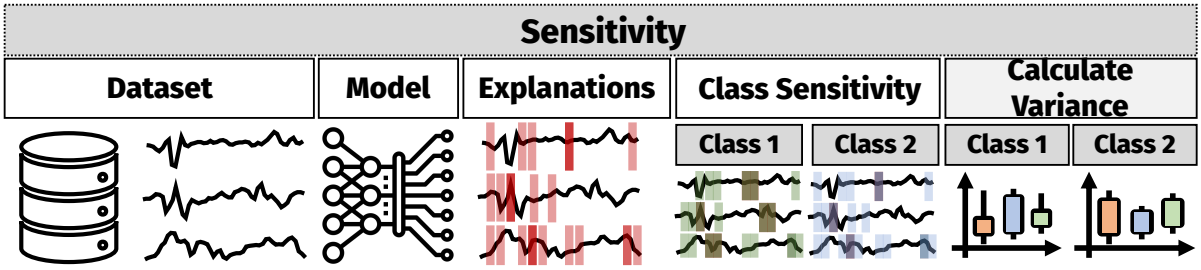


Figure 3.9 – A step-by-step representation of the sensitivity workflow. Importance maps are generated for each class, grouped accordingly, and their variance is computed to determine how the model's focus shifts in response to different class labels. A larger variance indicates that the model is more sensitive to the distinctive features associated with each class.

different segments of the dataset.

3.5.1.4 Robustness

The robustness metric evaluates the reliability of interpretability methods against adversarial perturbations. This approach focuses on understanding how small deliberate changes in input data impact the generated explanations. A robust interpretability method will produce consistent importance maps even when the input is slightly modified, indicating that the model's explanation is not easily disrupted.

The Figure 3.10 illustrates a step-by-step process for assessing the robustness of model explanations when inputs are perturbed. First, the original dataset is fed into a trained model, which outputs both predictions and corresponding explanation maps (highlighted bars or regions that indicate feature importance). Next, a perturbed version of the dataset is created by adding targeted noise or masking, this represents the adversarial component designed to minimally alter the input while potentially misleading the model. The same model is then applied to the perturbed inputs, producing a new set of explanations. Finally, the original and perturbed explanations are compared using various metrics, such as MSE, MAE, RMSE, cosine similarity, or Euclidean distance, to quantify how similar or different they are. By visualizing these metrics and any changes in the explanation maps we can measure whether minor input changes can substantially alter the explanations.

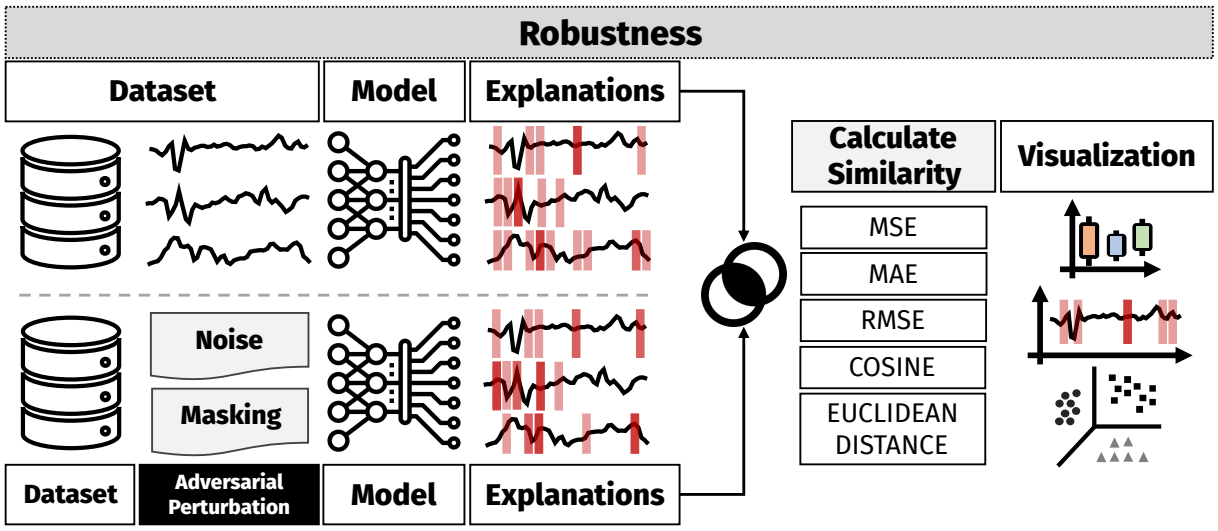


Figure 3.10 – Illustration of a robustness evaluation. The top row shows the baseline process: data fed into a trained model to generate explanations (highlighted regions). In the bottom row, adversarial or noise-based perturbations are applied to the data before generating new explanations. The two sets of explanations are then compared using similarity or distance metrics (MSE, MAE, RMSE, cosine, etc.) and visualized to assess how stable (i.e., robust) the explanations remain under perturbation.

3.5.1.5 Stability

Stability refers to the consistency of importance maps in assigning similar relevance scores to analogous features across different instances of the same class. This property ensures that the interpretability method remains dependable, offering consistent explanations regardless of variations in models or repeated computations.

The stability metric workflow, shown in Figure 3.11, starts by preparing a dataset and selecting multiple model architectures (e.g. neural networks), to cover a wide range of learning approaches. Each architecture is trained multiple times, often using different hyperparameter settings or random seeds, resulting in multiple runs Once training is complete, an XAI method (e.g., SHAP) is applied to a consistent subset of test samples, producing saliency or feature importance maps for every model run. These maps are then compared pairwise using similarity or distance metrics. The goal is to measure how consistently the explanations align across different instances of training or model configurations.

Finally, these pairwise comparisons are aggregated into a single stability metric that captures the degree of variation in the explanations over runs. If explanations

remain largely similar, reflected by small distances or high similarity scores, a high stability rating is assigned, indicating that the interpretability method provides stable results. If, however, explanations differ substantially among models with minor training variations, one concludes that the method (or the underlying model) may be less reliable for interpretability purposes.

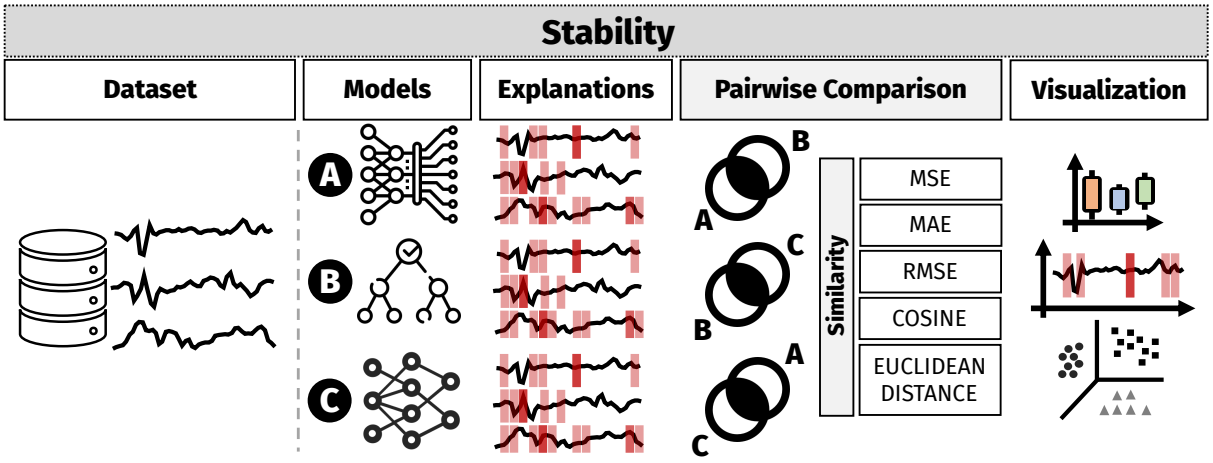


Figure 3.11 – Overview of the Stability Workflow. Multiple machine learning models (or different runs of the same model architecture) are trained with varying seeds, hyperparameters, or data splits. For each model instance, importance maps are generated, and distance or similarity metrics are computed among these maps to yield an overall stability score. A higher stability score suggests more consistent and reliable explanations.

3.5.1.6 Localization

Localization evaluates whether model explanations, typically feature-importance maps, accurately highlight the segments of input data most relevant to the task. In the context of time-series tasks, this means that the importance maps should accurately highlight features within or near the relevant segments of the time-series data. By directly comparing the model's saliency maps to expert-annotated regions, the localization metric sheds light on possible shortcomings. For example, if a model frequently flags irrelevant input segments, it may rely on spurious correlations instead of meaningful patterns. Likewise, a model that highlights accurate time periods in a scattered or unstable manner raises concerns about reliability.

The temporal location of class-specific features that are highly relevant in a time

series should naturally be located within or near its designated segment, which is a temporal subsequence, as shown in Figure 3.12. The relevant segments are often defined by domain-specific knowledge, represent the portions of the input most relevant to the task.

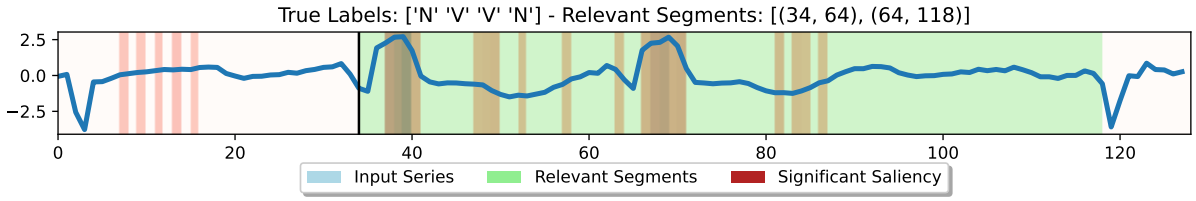


Figure 3.12 – Localization of importance maps align with relevant segments.

We present our localization workflow in Figure 3.13. Domain experts first annotate the data, often time series samples, with specific intervals where relevant features are known to reside. In our context, arrhythmia and normal heartbeats. These serve as ground-truth references for later comparisons. Once the model is trained on this annotated dataset, feature importance maps are produced for each test instance. By applying a threshold to binarize these maps, one can easily compare the 'highlighted' regions to the annotated intervals, thus measuring the correctness of the saliency overlap. The second measure, temporal coherence, captures whether these highlighted time steps form continuous, sensible sequences rather than scattered points. Combining the segment analysis score and the temporal coherence score into a single localization metric provides a holistic sense of how effectively the interpretability method pinpoints pertinent features in the data. Higher scores signal explanations that align both correctly with relevant segments and maintain consistency in contiguous time spans, two hallmarks of effective localization.

3.6 Tools for Explainable AI Visualization

Explainable AI tools offer a variety of visualization techniques to make the decision-making process of machine learning models more transparent. These tools allow us to interpret model predictions by highlighting feature importance, analyzing data distributions, and identifying patterns in decision logic. In this work, we use most

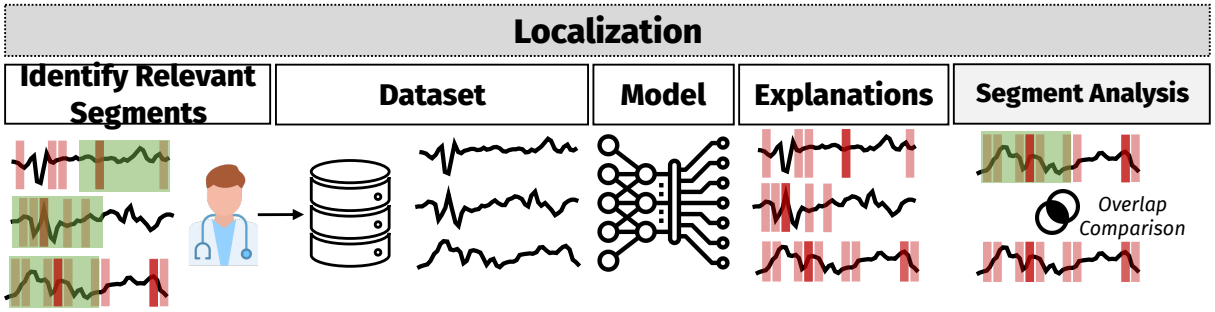


Figure 3.13 – Overview of the Localization Metric Workflow. The process begins with domain experts annotating the relevant segments within the input data. A trained model then generates importance maps for each input instance, which are compared against these known segments.

common techniques used for data visualization:

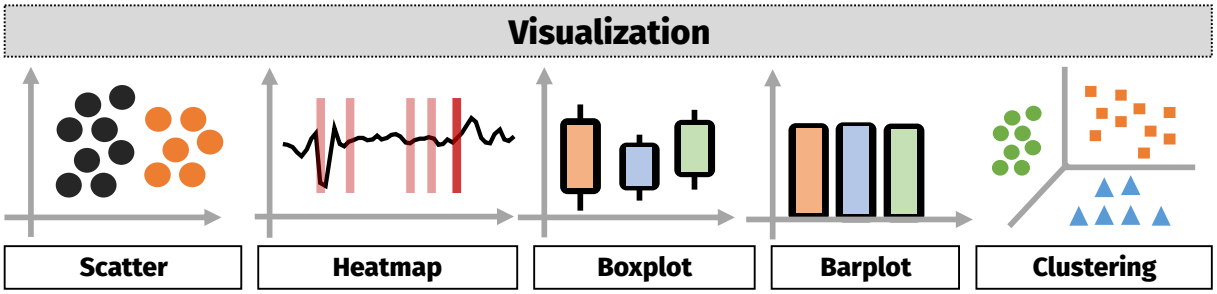


Figure 3.14 – Visualization tools

- Heatmaps are one of the most popular tools for visualizing feature importance, particularly in image-based tabular data and time series data. In the context of XAI, heatmaps overlay a color gradient on the input data to highlight the regions or features most influential in the model's prediction. For instance, in image classification tasks, heatmaps can pinpoint areas of the image that contribute significantly to the predicted label. In time series data, heatmaps can illustrate the relative importance of features across multiple data points. The intuitive color gradients make heatmaps particularly useful for identifying areas of focus in complex models.

- Cluster visualizations are used to group similar data points based on their feature values or model outputs, often using techniques like t-SNE (t-Distributed Stochastic Neighbor Embedding) or UMAP (Uniform Manifold Approximation and Projection). In XAI, clusters help to identify patterns in data or explain model behavior by grouping samples with similar predictions or feature contributions. For exam-

ple, a clustering analysis may reveal that a model treats certain groups of samples similarly, shedding light on potential biases or decision patterns. Visualizing these clusters can help to understand the model performance on subpopulations or identifying outliers.

- Boxplots are statistical charts that summarize the distribution of a feature's impact across a dataset. In XAI, boxplots are used to compare the distribution of feature importance maps for different features. They provide a quick way to visualize the median, quartiles, and outliers of feature contributions, allowing practitioners to understand the variability and central tendency of feature impacts. For example, a narrow boxplot with minimal outliers suggests that a feature has a consistent influence across the dataset, while a wide boxplot with many outliers indicates variability in its impact.

- Barplots are simple yet effective tools for comparing the relative importance of features. In XAI, barplots are often used to display the average of feature importance maps for a set of features, ranked in descending order. Each bar's length corresponds to the average magnitude of the feature's contribution to the predictions. Barplots are particularly useful for providing a global view of feature importance, helping users quickly identify the most influential features across the entire dataset.

3.6.1 Global Interpretable Clustering

In this thesis, we introduce *Global Interpretable Clustering (GIC)*, a new qualitative approach to evaluate the consistency and reliability of explainability methods (e.g. SHAP, LIME) when applied to machine learning models, particularly in the context of time series classification.

As shown in Figure 3.15, the GIC method begins by computing feature-importance maps for each instance in the dataset using one or more explainability methods. Because these importance maps can be high-dimensional, GIC then applies dimensionality reduction algorithms such as *Principal Component Analysis (PCA)*, *t-distributed Stochastic*

Neighbor Embedding (t-SNE), or *Uniform Manifold Approximation and Projection (UMAP)*. This transformation preserves the relative relationships between instances while simplifying visual analysis and subsequent clustering. An example of GIC visualization is shown in Figure 3.16.

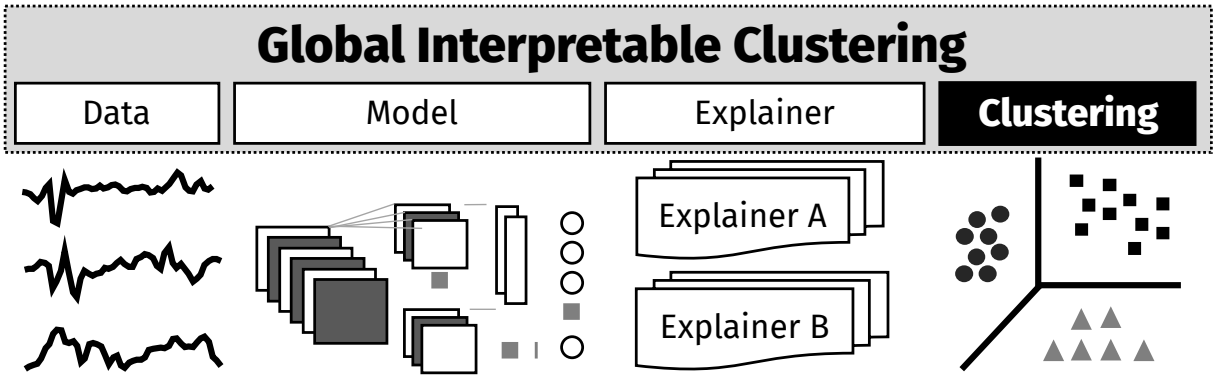


Figure 3.15 – Feature importance values generated by different explainers (e.g., SHAP, LIME) are clustered using dimensionality reduction techniques such as PCA, t-SNE, and UMAP. The resulting clusters shown consistency and coherence across different explainers, revealing patterns in how each method captures significant relationships within the data.

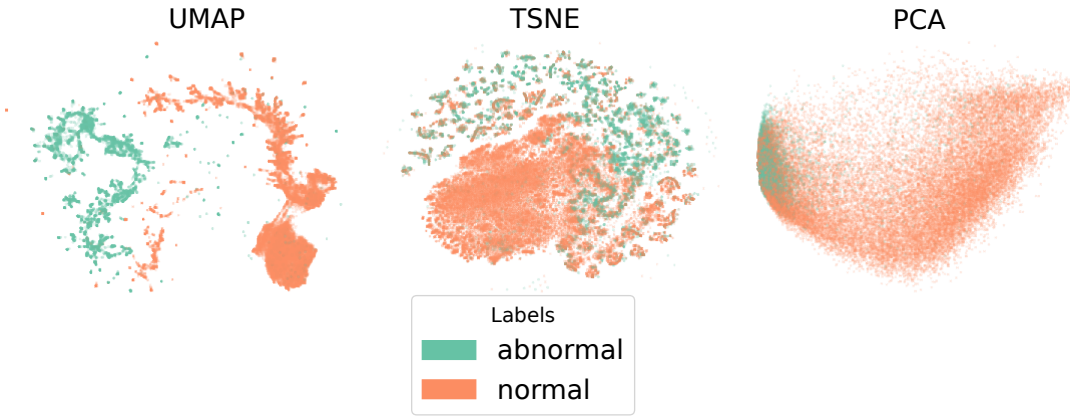


Figure 3.16 – Global Interpretable Clustering method using diferent clustering stratagies such as UMAP, TSNE and PCA.

Algorithm 3.5 outlines the GIC workflow. systematically integrates feature importance computation, dimensionality reduction and visualization, providing a way to assess how well each explainer captures the underlying decision-making process of the model. By comparing the clusters formed by different reduction techniques (PCA, t-SNE, UMAP) and explainers, one can gauge the consistency of the explanations and identify patterns indicating where certain explainers may excel or underperform.

Algorithm 3.5 Global Interpretable Clustering algorithm

Require: Model M, Data D, Explainability Method E
Ensure: Clusters of feature importance for comparison
Step 1: Compute Feature Importance
for each instance $d_i \in D$ **do**
    Apply E to M and d_i
    Compute feature importance $FI_i \in \mathbb{R}^k$
end for
Step 2: Collect Feature Importance Scores
Create feature importance matrix $F \in \mathbb{R}^{m \times k}$, where m is the number of data instances, and k is the number of features.
Step 3: Dimensionality Reduction
Apply PCA, t-SNE, or UMAP to reduce the dimensionality of F
Let $F_{\text{reduced}} \in \mathbb{R}^{m \times d}$ represent the reduced feature importance matrix (d is the reduced dimension)
Step 4: Clustering
Apply clustering algorithm (e.g., K-means) on F_{reduced}
Let $C = \{C_1, C_2, \ldots, C_p\}$ be the resulting clusters
Step 5: Visualization and Analysis
Visualize clusters using scatter plots for each reduction technique
Analyze the consistency of feature importance patterns across clusters

Algorithm 3.5 begins by computing feature importance maps for each instance in the dataset using an explainability methods. For each data instance, explainers like SHAP or LIME are applied to the trained model, producing a feature importance map that highlights which elements of the input data most influence the model's prediction. This step results in a collection of feature importance maps for each instance.

Given the high-dimensional nature of the feature importance data, the next step involves applying dimensionality reduction techniques such as Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP). These techniques are used to reduce the feature importance matrix into a lower-dimensional space, enabling easier analysis while preserving the relationships between instances. The output of this step is a reduced feature importance matrix, where each row corresponds to an instance, but with fewer dimensions, making it more suitable for clustering and visualization.

In the clustering step, a clustering algorithm (e.g., K-means) can be applied to the reduced feature importance matrix to group instances into clusters based on the similarity of their feature importance patterns. Each cluster represents a set of instances

for which the explainability methods yield similar importance values, thus highlighting areas where explainers align or diverge in their interpretations. This clustering step enables the qualitative assessment of how consistent the explainers are in identifying the key features across multiple instances.

Once clustering is completed, the next step involves visualization and analysis of the results. The clusters are visualized using scatter plots corresponding to each dimensionality reduction technique (PCA, t-SNE, UMAP), allowing comparisons between the clusters. These visualizations allows to compare the consistency and stability of the explanations generated by different methods. We can analyze the groupings to see whether the same patterns are captured across different explainers, or whether certain explainers tend to deviate from others in how they assign feature importance.

This type of qualitative assessment of XAI methods has not been thoroughly explored in prior research. By clustering and visualizing feature importance maps, this thesis provides a new dimension to XAI evaluation, offering a tool for community to better understand how different interpretable methods compare in capturing the significant features of the data. Moreover, it enables the detection of patterns in how feature importance varies across methods, instances, and models, thus advancing the field's understanding of the behavior of interpretable AI techniques.

3.7 Discussion and Advantages of a UTS-XAI

Adopting multiple interpretability metrics in a time-series context—namely, sanity, sensitivity, robustness, faithfulness, stability, and localization—offers a well-rounded picture of how effectively and reliably a model's explanations capture its decision-making process. Each metric targets a distinct aspect of explanation quality, and when considered together, they highlight different strengths and vulnerabilities in the model's explanatory outputs.

From the outset, faithfulness confirms whether explanations truly mirror the model's internal logic. If the model claims certain features are critical, altering those features should significantly impact its predictions. Meanwhile, sanity assures us that

the explanations do indeed depend on the model's learned parameters: if randomizing the labels barely changes the saliency maps, it suggests the explanation method may not be reflecting anything substantial about the learned weights. The sensitivity metric then delves into whether the explanation adapts granularly both to different classes and to individual samples, exposing the model's capacity to capture fine-grained patterns rather than offering a generic explanation for all inputs.

Alongside sensitivity, robustness evaluates how much the explanation shifts when small, often adversarial perturbations are introduced into the input. A robust explanation should remain stable for minor variations that do not meaningfully affect the model's overall reasoning. At the training or configuration level, stability checks if explanations hold steady across different runs—varying random seeds, initializations, or hyperparameters. This prevents over-reliance on a single training artifact and underscores whether the explanation is rooted in the data rather than in happenstance training conditions. Finally, localization verifies if the explanation pinpoints the correct temporal (or spatial) segments known to be important, ensuring that highlighted intervals align with domain-specific knowledge or labeled segments.

By synthesizing these six metrics into a unified framework, practitioners can diagnose an interpretability method's adequacy from multiple vantage points. For example, a model might score high on faithfulness yet exhibit low robustness, meaning it genuinely captures relevant features but is highly susceptible to small input tweaks. Conversely, a stable model with strong localization might fail the sanity check if its explanations hardly change even when the labels are randomized. Using all metrics in tandem illuminates such discrepancies, allowing data scientists to pinpoint shortcomings and refine both the model and the interpretability technique. In turn, stakeholders—be they clinicians, financial analysts, or industrial engineers—gain increased confidence that the time-series explanations are not only faithful and robust but also stable, sensitive, and properly localized where it matters most.

4

Experiments and Results

In this chapter, we introduce the experimental protocol and present the results obtained from applying our Unified Time-Series Explainable Artificial Intelligence (UTS-XAI) framework. We begin with experimental protocol by describing the selected datasets, outlining the preprocessing steps, and detailing the model training configurations, thereby providing a robust, reproducible evaluation process. Next, we introduce a comprehensive set of evaluation scenarios, each specifically designed to examine different aspects of the UTS-XAI framework, ranging from predictive accuracy to interpretability robustness. Subsequently, we discuss the experimental results and provide an analysis of UTS-XAI's strengths and limitations. By comparing the framework's performance across varied scenarios and highlighting the interplay between classification accuracy and interpretability, we present the benefits of integrating explainability into time-series classification tasks.

4.1 Experimental Protocol

The experimental protocol presented in this thesis is designed to systematically evaluate both the classification performance of multiple models and the quality of their generated explanations. As illustrated in Figure 4.1, we begin by describing the evaluation scenarios used to rigorously test our framework: 1) train and test well-calibrated classification models used as a solid foundation for 2) subsequent Explainable AI evaluation. We also evaluate our 3) proposed GIC, used to discover feature importance patterns

produced by different XAI techniques, and observe how consistently these methods capture meaningful relationships. In the remainder of this section, we provide an overview of the PhysioNet-based arrhythmia datasets, discussing their composition, class distribution, and preprocessing steps, such as filtering, segmentation, and normalization. We then explain the validation procedures, including how the data was split into training, validation, and test sets. Following this, we outline the evaluation metrics and justify their relevance to the arrhythmia detection task. We also describe baseline classification models, detailing their architectures, hyperparameters, and training configurations. Finally, we introduce the Explainable AI configurations employed to interpret model predictions. We run our experiment using in a hardware with Processador AMD Ryzen 7 5800X, RTX 3090 24GB GDDR6X, 64 GB RAM DDR4 3200mhz,, SSD 512GB M.2 Nvme.

4.1.1 Evaluation Scenarios

The following evaluation scenarios are considered in our experiment, as we present in Figure 4.1:

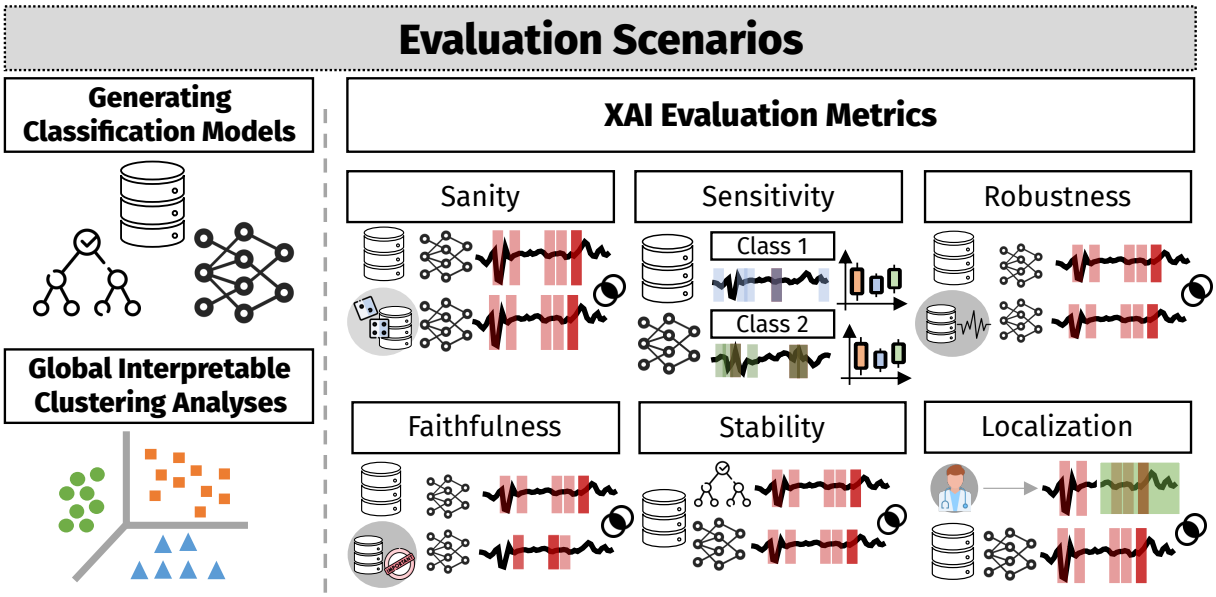


Figure 4.1 – Evaluation Scenarios

1. Generating Classification Models: in this phase, our goal is not to develop state-of-the-art models but rather to obtain well-calibrated models that are not biased by

train and test data. By using different datasets, we ensure that the models generalize appropriately and serve as a foundation for evaluating Explainable AI (XAI) methods. Specifically, we unified the MIT-BIH and SVDB datasets for training and used the INCART dataset exclusively for testing. The classification performance was assessed using three models: XGBoost, FCN, and DeepConvLSTM. These models are subsequently analyzed using XAI methods to obtain explainability results and assess the quality of feature attributions.

2. Global Interpretable Clustering Analysis: in this phase, we analyze feature importance patterns obtained from the XGBoost, FCN, and DeepConvLSTM models using three XAI methods: SHAP, LIME, and Saliency Maps. We apply dimensionality reduction techniques, namely PCA, t-SNE, and UMAP, to visualize and interpret the feature importance representations. The proposed Global Interpretable Clustering approach allows us to identify consistent patterns across different XAI methods, facilitating a qualitative comparison of their ability to capture meaningful relationships in the data. This enables us to assess the alignment of explanations with model behavior and determine the robustness of feature attributions.

3. Explainable AI Evaluation: the models considered for XAI evaluation metrics include DeepConvLSTM, FCN, and XGBoost, while the XAI methods used are SHAP Explainer (EX), LIME Tabular Explainer (LTE), Saliency Map (SM), and SHAP Tree Explainer (TEX). The performance of these methods is assessed using multiple metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Cosine Similarity, Euclidean Distance, and Structural Similarity Index (SSIM).

    - Sanity: evaluates whether the importance maps focus is significantly influenced by the model learned parameters. By randomizing labels and training a model, one expects the importance maps to become less meaningful, demonstrating that the saliency map indeed depends on the learned parameters rather than being a generic or arbitrary visualization.

- Faithful: evaluates how well the importance maps correlates with the model's output changes in response to input perturbations. A faithful importance map would highlight parts of the input that, when altered, significantly affect the prediction. This is usually tested by perturbing parts of the input sequence and observing the impact on the output, expecting that areas marked as highly salient should cause greater changes in the output.
- Sensitivity: checks if the importance map can distinguish between different samples, especially focusing on the predicted class. A sensitive importance map would generate different patterns for inputs belonging to different classes or even for different instances within the same class, reflecting the model's sensitivity to the unique aspects of each input.
- Robustness: check consistency under small changes to the input data. Minor modifications to the input should not drastically change the importance map, indicating that the importance map is focusing on genuinely relevant features rather than noise or irrelevant variations.
- Stability: evaluate the ability to consistently identify similar features or patterns as important for similar or identical classes across different samples. This means that for inputs classified into the same category, the importance maps should highlight similar features as being important.
- Localization: evaluates if an importance map is localized around the time segments most relevant to the prediction. This ensures the interpretability of the model's focus and decisions, particularly in temporal contexts where the timing of events can be crucial.

4.1.2 Datasets Description

The datasets used in this study come from the PhysioNet collection, which provides a variety of electrocardiogram recordings, including data from the MIT-BIH Arrhythmia Database (PHYSIONET, 2005), MIT-BIH Supraventricular Arrhythmia Database (PHYSIONET, 1999), and the St. Petersburg INCART Arrhythmia Database (PHYSIONET,

2008). Each dataset contains detailed annotations for different types of heartbeat, classified according to the clinical diagnosis of arrhythmias.

Table 4.1 lists the symbols used to represent various types of heartbeats in the PhysioNet arrhythmia dataset. Each symbol corresponds to a specific heartbeat type, including normal beats, ectopic beats (such as atrial and ventricular premature beats), and other specialized beats such as paced beats or escape beats. Symbols for events such as the start and end of ventricular flutter-fibrillation are also included. Figure 4.2 shown examples of ECG heartbeats from PhysioNet arrhythmia dataset.

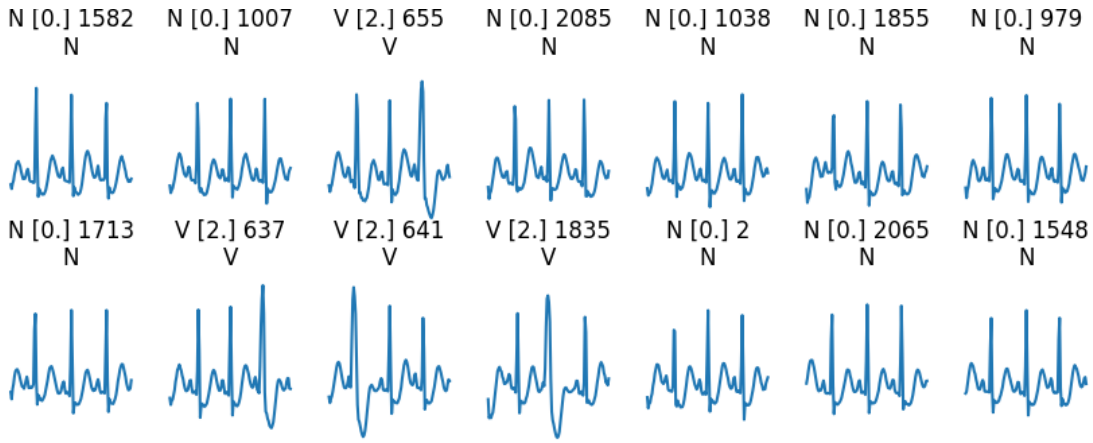


Figure 4.2 – ECG heartbeats samples from PhysioNet arrhythmia dataset.

Table 4.1 – Symbol Definitions for ECG Beat Types.

Symbol	Description
N	Normal beat
.	Normal beat
L	Left bundle branch block beat
R	Right bundle branch block beat
A	Atrial premature beat
a	Aberrated atrial premature beat
J	Nodal (junctional) premature beat
S	Supraventricular premature beat
V	Premature ventricular contraction
F	Fusion of ventricular and normal beat
[	Start of ventricular flutter-fibrillation
!	Ventricular flutter wave
]	End of ventricular flutter-fibrillation
e	Atrial escape beat
j	Nodal (junctional) escape beat
E	Ventricular escape beat
/	Paced beat
f	Fusion of paced and normal beat
x	Non-conducted P-wave (blocked APB)
Q	Unclassifiable beat

Table 4.2 categorizes ECG beat symbols into five classes according to the AAMI standard: normal (N), ventricular (V), supraventricular (S), fusion (F), and unknown

(Q). Although the standard defines a fusion (F) class to capture beats that exhibit characteristics of both normal and ventricular patterns, the F and Q classes are not sufficiently represented in our dataset. Consequently, to ensure a robust and balanced analysis, we opted to exclude these underrepresented classes.

Table 4.2 – Beat Classifications accordind to AAMI standard: normal (N), ventricular (V), supraventricular (S), fusion of normal and ventricular (F) and unknown beats (Q).

Class	**Symbols**	**Description**
N	N, ., L, R, e, j	Non-ectopic beats
S	A, a, J, S, x	SVEB (Supraventricular ectopic beat)
V	V, E, !	VEB (Ventricular ectopic beat)
F	F	Fusion beat
Q	P, /, f, u, Q	Unknown beat

For the purpose of our binary classification task, we combined the ventricular (V) and supraventricular (S) to create a Arrhythmia class (Abnormal) and a normal (N) class, which constitute the most representative categories in the data set. This binary grouping allows us to focus on the discrimination between normal and abnormal rhythms while avoiding potential biases introduced by classes with very few instances.

4.1.2.1 MIT-BIH Arrhythmia Database

The MIT-BIH Arrhythmia Database (MIT-BIH) contains 48 half-hour samples of two-channel ambulatory ECG recordings acquired from the BIH Arrhythmia Laboratory's 47 patients investigated between 1975 and 1979. Twenty-three recordings were chosen at random from a set of 4000 24-hour ambulatory ECG recordings collected at Boston's Beth Israel Hospital from a mixed population of inpatients (about 60%) and outpatients (about 40%); the remaining 25 recordings were chosen from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample. The recordings were digitalized over a 10 mV range at 360 samples per second, each channel. Each record was separately annotated by two or more cardiologists; differences were settled to produce the computer-readable reference annotations for each beat, which are supplied with the database and total over 110,000

annotations.

4.1.2.2 MIT-BIH Supraventricular Arrhythmia Database

The MIT-BIH Supraventricular Arrhythmia Database (SVDB) supplements the MIT-BIH Arrhythmia Database by providing examples of less common supraventricular arrhythmias. It includes 78 half-hour ECG recordings, with annotations for various types of supraventricular arrhythmic events.

4.1.2.3 St Petersburg INCART Arrhythmia Database

The St Petersburg INCART Arrhythmia Database (INCART) contains 75 annotated recordings culled from 32 Holter recordings. Each recording lasts 30 minutes and contains 12 standard leads sampled at 257 Hz with gains ranging from 250 to 1100 analog-to-digital converter units per millivolt. Gains are specified in each record's.hea file. There are over 175,000 beat annotations in total in the reference annotation files. The original records were obtained from patients undergoing coronary artery disease testing (17 men and 15 women, ages 18 to 80; mean age: 58). None of the patients had pacemakers, and the majority of them had ventricular ectopic beats. Subjects with ECGs consistent with ischemia, coronary artery disease, conduction abnormalities, and arrhythmias were prioritized for inclusion in the database.

4.1.2.4 Dataset Summarization

Table 4.3 outlines the three PhysioNet-based ECG arrhythmia databases used in this study. It includes the MIT-BIH Arrhythmia Database (MIT-BIH), the MIT-BIH Supraventricular Arrhythmia Database (SVDB), and the St. Petersburg INCART Arrhythmia Database (INCART). These datasets vary in signal frequency, lead configurations, and the number of subjects.

The MIT-BIH and SVDB databases are merged to form a binary classification

dataset (normal vs. abnormal), used exclusively for training and validation. In contrast, the INCART database, which includes entirely different subjects, is reserved for testing, thus minimizing overfitting and mitigating potential biases in model evaluation. The Table 4.3 shown the key characteristics of the datasets used in this study.

Table 4.3 – Summary of the ECG arrhythmia datasets and their class distributions. The combined MIT-BIH and SVDB datasets form a binary classification set (normal vs. abnormal) used for training and validation, while the INCART dataset (composed of different subjects) is used solely for testing.

Database	**Freq. (Hz)**	**Signals**	**N. Subj.**	**Set**	**Binary Dist.**	**Multi-Class Dist.**
MIT-BIH Arrhythmia Database	128 (360)	MLII (MLII, V5)	45	-		
MIT-BIH Supraventricular Arrhythmia Database	128	ECG1 (ECG1, ECG2)	78	-		
Combined (MIT-BIH + SVDB)	*128*	*MLII + ECG1*	123	Train	Normal: 206.992 Abnormal: 65.265	Normal: 206.992 Ventricular: 34.797 Supraventricular: 30.468
St Petersburg INCART Arrhythmia Database	128 (257)	Lead II (12 leads)	75	Test	Normal: 124.742 Abnormal: 50.365	Normal: 124.742 Ventricular: 45.925 Supraventricular: 4.440

Combining the MIT-BIH and SVDB databases for model training, and subsequently validating on the INCART Database broadens the diversity of the training data, covering a extensive range of arrhythmia patterns and ECG morphologies. By exposing the model to different patient populations and a variety of arrhythmic events, this approach enhances the model's robustness and ability to generalize beyond a single dataset. Furthermore, merging multiple sources of data helps mitigate biases that may arise from using a single source. In addiction, by validating on the INCART dataset an entirely separate and potentially more varied data source, the resulting performance metrics more accurately reflect the model's practical clinical utility. This independent validation step supports the claim that the model can maintain reliable accuracy when faced with real-world variations in patient demographics, recording conditions, and device specifications. Ultimately, this multi-dataset training and external validation strategy increases the credibility of the model and strengthens its reliability.

4.1.2.5 Dataset Processing

Given the heterogeneity of the datasets differences in signal frequencies, lead configurations, and subject characteristics, there important step to standardize and preprocess

the data to create a unified dataset. We aim to maintain consistency and compatibility for downstream analysis, model training and validation. We present the steps taken by Algorithm 4.1 to align sampling frequencies, normalize signal characteristics, and harmonize labels across datasets, addressing challenges posed by their diverse formats while preserving the integrity of the original data.

The $build\_classification$ function is designed to process and classify electrocardiogram (ECG) signals. It takes as input the raw ECG signal (p_signal), annotation indices (ann_idx), annotation symbols (ann_sym), sampling frequency (fs), a time window in seconds (num_sec), a classification dictionary ($physionet\_dict$), and several optional parameters for filtering, normalization, resampling, and logging. The function starts by initializing matrices for storing processed signals and their labels, as well as a dataframe for easy querying of annotations.

During the processing phase, if enabled, the ECG signal is filtered to remove noise. The function then iterates over each annotation. For each annotation, it processes a segment of the ECG signal determined by the given time window around the annotation index. The classification of each segment is done using the $label\_choice$ function, which determines the appropriate label based on the annotation symbols and the $physionet\_dict$ dictionary.

The function also handles various conditions like ignoring specified classes and checks if the processed signal segments meet the expected size. If normalization or resampling is required, these operations are performed on the processed signal data.

Finally, the function returns the processed signal data (X), the corresponding labels (Y), and additional arrays containing symbols, indices, label lists, positions of indices, and true labels for each processed annotation. Auxiliary functions like intersection, $label\_choice$, and $process\_label$ support the main functionality by performing tasks such as finding the intersection of lists and determining classification labels for given sets of annotation symbols. This function is a complex tool likely used in medical or research settings for analyzing heart rhythm data from ECG recordings.

Algorithm 4.1 Build Physionet Dataset for Classification

Require: p_signal, ann_idx, ann_sym, fs, num_sec, $physionet\_dict$, $ignore\_classes$, $filtering$, $norm$, $resample$, $outliers$, log

$num\_cols \leftarrow 2 \times num\_sec \times fs$
$num\_rows \leftarrow$ **length**(ann_idx)
$X \leftarrow$ zero matrix of size $num\_rows \times num\_cols$
$Y \leftarrow$ zero matrix of size $num\_rows \times 1$
Initialize sym, $index$, $labels$, $index\_pos$, $true\_labels$ as empty lists
$row\_ \leftarrow 0$
$df\_ \leftarrow$ DataFrame with columns 'idx' and 'symbols' from ann_idx and ann_sym
for $(idx\_, sym\_, pos\_)$ in **zip**(ann_idx, ann_sym, **range**(**len**(ann_idx))) **do**
    $left \leftarrow \max(0, idx\_ - num\_sec \times fs)$
    $right \leftarrow \min($**len**$(p\_signal), idx\_ + num\_sec \times fs)$
    $idx\_range \leftarrow$ **range**$(left, right)$
    $query\_ \leftarrow$ rows in $df\_$ where 'idx' is in idx_range
    $label\_list \leftarrow$ values of 'symbols' in $query\_$
    $index\_list \leftarrow$ indices in idx_range where 'idx' is in $query\_$
    $ammi\_label$, $true\_label \leftarrow$ label_choice($label\_list$, $physionet\_dict$, **log** = **False**)
    if not $ammi\_label$ **or not** $true\_label$ **then**
        continue
    end if
    if $ignore\_classes$ **and** $ammi\_label$ in $ignore\_classes$ **then**
        continue
    end if
    $x \leftarrow$ segment of p_signal from $left$ to $right$
    if len(x) = num_cols **then**
        $X[row\_, :] \leftarrow x$
        $Y[row\_, :] \leftarrow$ class mapping of $ammi\_label$ in $physionet\_dict$
        Append $ammi\_label$ to sym
        Append other details to respective lists
        $row\_ \leftarrow row\_ + 1$
    else
    end if
end for
$X \leftarrow X[: row\_, :]$
if $norm$ **then**
    $X \leftarrow$ NormalizeData(X)
end if
if $resample$ **then**
    $X \leftarrow$ resample X with $resample$ parameters
end if
$Y \leftarrow Y[: row\_]$ **return** X, Y, and other collected arrays

4.1.3 Evaluation Procedures

In supervised machine learning, evaluating a model's performance typically involves splitting the dataset into training and test sets. This can be done using methods such as

hold-out, k-fold cross-validation (k-CV), leave-one-out cross-validation (LOOCV), or leave-one-subject-out cross-validation (LOSO). The classifier is trained on the training set and tested on the unseen test set to gauge its accuracy. The following sections outline these evaluation methods.

In this work, we adopted a hybrid approach tailored to our ECG datasets. The MIT-BIH and SVDB databases were merged and split into a training and validation set (using a hold-out method) to develop and fine-tune the classification model. This step leverages a broad range of arrhythmia samples while ensuring that the model parameters are properly calibrated. Subsequently, the INCART database, containing entirely different subjects, is set aside as the exclusive test set. This design choice provides a robust assessment of generalization performance by evaluating the model on genuinely novel subjects. Using an entirely separate database for testing, we minimize overfitting risks and mitigate potential biases that could arise from reusing subjects or signals in both training and testing phases.

4.1.4 Metrics for Evaluating Classification Models

In the context of binary arrhythmia classification, the model's performance can be examined through a confusion matrix that tracks how often the model correctly identifies arrhythmic (positive) and normal (negative) cases. Specifically, normal events are considered the "positive" class, while arrhythmic events are treated as the "negative" class. Consequently, a true positive (TP) corresponds to correctly identifying a normal event as normal, whereas a false positive (FP) arises when an arrhythmic event is mistakenly labeled as normal. Likewise, a true negative (TN) means that a correct classification of an arrhythmic event as arrhythmic, and a false negative (FN) indicates that a normal event has been incorrectly classified as arrhythmic. From these four quantities, we derive common metrics: accuracy, precision, recall, and F-measure to gain a comprehensive view of the effectiveness of a model. Table 4.4 summarizes the definitions of these metrics, illustrating how each offers unique perspectives into model performance for the binary task of distinguishing arrhythmic signals from normal signals.

Table 4.4 – Summarization of accuracy, recall, precision and F-measure. TP means true positives, TN true negatives, FP false positives and FN means false negatives.

Metric	Equation	Description
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Accuracy is the ratio of correct predictions divided by the total predictions.
Precision	$\frac{TP}{TP+FP}$	Precision is the ratio of true positives and total positives predicted.
Recall	$\frac{TP}{TP+FN}$	Recall is the ratio of true positives to all the positives in ground truth.
F Measure	$2 \times \frac{Precision \times Recall}{Precision+Recall}$	The F-measure is the harmonic mean of precision and recall.

4.1.5 Machine Learning models used as Baselines

To evaluate the effectiveness of our approach, we use three baseline models: DeepConvLSTM, Fully Convolutional Networks (FCN), and XGBoost. While DeepConvLSTM excels at capturing both spatial (via convolutional layers) and temporal (via LSTM layers) relationships, FCN provides a simpler alternative with reduced training times by leveraging only convolutional layers. XGBoost, on the other hand, offers a strong baseline with minimal feature engineering but cannot natively model temporal or spatial correlations without additional preprocessing. By comparing these three models, we can evaluate the strengths and weaknesses of different approaches to time series classification in the context of system performance data.

4.1.5.1 DeepConvLSTM

Manual feature extraction from high-dimensional ECG time-series data is often laborious and requires substantial domain expertise to identify relevant morphological and temporal characteristics. To address these challenges in arrhythmia classification, we adopt the DeepConvLSTM architecture introduced by Ordóñez and Roggen (ORDÓÑEZ; ROGGEN, 2016). This model leverages convolutional neural networks (CNNs)

for automated feature extraction and Long Short-Term Memory (LSTM) layers for capturing sequential dependencies in the cardiac signal.

In this setup, convolutional layers learn to detect important ECG patterns—such as QRS complexes, P waves, and T waves—by scanning local regions of the input signal. These extracted features are then passed to LSTM layers, which model longer-term temporal dependencies crucial for identifying subtle arrhythmic patterns that span multiple beats or time intervals. This design is effective when minute variations in signal morphology and timing can significantly impact classification decisions—precisely the case for arrhythmia detection.

The DeepConvLSTM model comprises three convolutional layers followed by two recurrent LSTM layers. Its output layer is a dense neuron with a sigmoid activation function, which estimates the probability that a given ECG segment is normal (positive class) versus arrhythmic (negative class). Figure 4.3 provides a conceptual view of this architecture, while Table 4.5 details the layer configurations and parameters. This end-to-end structure alleviates the need for manual feature engineering and offers a robust framework for learning both local (via CNNs) and temporal (via LSTMs) patterns in ECG signals, making it well suited for the binary classification of arrhythmias.

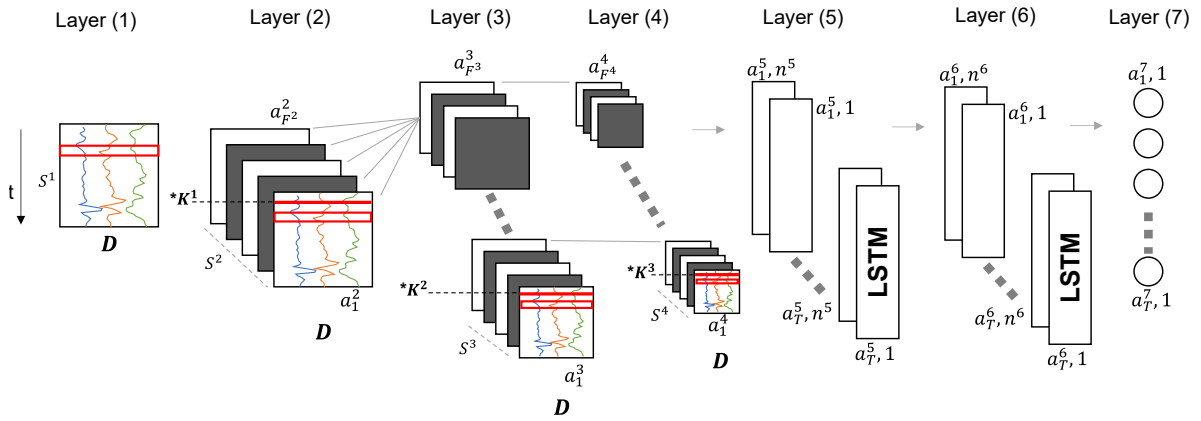


Figure 4.3 – Architecture for continuous authentication based on DeepConvLSTM neural network. Three convolutional layers process the operational system performance counter data. Two recurrent layers produce the classification result with an output layer. A dense layer of 1 unit with the sigmoidal activation function contains the probability that the sample belongs to the genuine user or imposter.

Layer (type)	**Output Shape**	**Param #**
input_1 (InputLayer)	(None, 128, 1)	0
conv_1 (Conv1D)	(None, 128, 64)	256
batch_normalization (BatchNormalization)	(None, 128, 64)	256
activation (Activation)	(None, 128, 64)	0
conv_2 (Conv1D)	(None, 128, 64)	12,352
batch_normalization_1 (BatchNormalization)	(None, 128, 64)	256
activation_1 (Activation)	(None, 128, 64)	0
conv_3 (Conv1D)	(None, 128, 64)	12,352
batch_normalization_2 (BatchNormalization)	(None, 128, 64)	256
activation_2 (Activation)	(None, 128, 64)	0
conv_4 (Conv1D)	(None, 128, 64)	12,352
batch_normalization_3 (BatchNormalization)	(None, 128, 64)	256
activation_3 (Activation)	(None, 128, 64)	0
lstm (LSTM)	(None, 128, 64)	33,024
dropout (Dropout)	(None, 128, 64)	0
lstm_1 (LSTM)	(None, 128, 64)	33,024
dropout_1 (Dropout)	(None, 128, 64)	0
time_distributed (TimeDistributed)	(None, 128, 3)	195
activation_4 (Activation)	(None, 128, 3)	0
lambda (Lambda)	(None, 3)	0

Table 4.5 – Deep Model Summary: A detailed breakdown of the model's architecture, including each layer's type, output shape, and parameter count.

4.1.5.2 Fully Convolutional Network (FCN)

In addition to our DeepConvLSTM approach, we also employ a Fully Convolutional Network (FCN), which offers a purely convolutional architecture for binary arrhythmia classification. Unlike recurrent-based models, FCNs rely solely on convolutional layers to learn features, eliminating the overhead of processing sequential data step by step. This architecture was used in various studies that involved time series data (FAWAZ et al., 2018; FAWAZ et al., 2019).

This design makes FCNs more computationally efficient during both training and inference. Multiple stacked convolutional layers detect morphological patterns associated with normal and arrhythmic beats at different scales, while global average pooling aggregates these learned features into a low-dimensional representation. This final representation is then passed to a dense output layer (e.g., with sigmoid activation) to classify the signal as normal (positive) or arrhythmic (negative). Because FCNs process the entire input in parallel rather than unrolling it over time steps, they are par-

ticularly advantageous when local morphological cues are more critical than capturing long-range temporal dependencies. Consequently, FCNs can excel in scenarios where rapid detection of ECG anomalies is essential, and the cost of maintaining recurrent layers may outweigh the benefits of modeling longer temporal contexts.

Layer (type)	**Output Shape**	**Param #**
input_layer_1 (InputLayer)	(None, 128)	0
reshape_1 (Reshape)	(None, 128, 1)	0
conv_1 (Conv1D)	(None, 128, 128)	1,152
batch_normalization_4 (BatchNormalization)	(None, 128, 128)	512
activation_5 (Activation)	(None, 128, 128)	0
conv_2 (Conv1D)	(None, 128, 256)	164,096
batch_normalization_5 (BatchNormalization)	(None, 128, 256)	1,024
activation_6 (Activation)	(None, 128, 256)	0
conv_3 (Conv1D)	(None, 128, 128)	98,432
batch_normalization_6 (BatchNormalization)	(None, 128, 128)	512
activation_7 (Activation)	(None, 128, 128)	0
GAP (GlobalAveragePooling1D)	(None, 128)	0
predictions (Dense)	(None, 2)	258

Table 4.6 – Summary of the model architecture, including each layer's type, output shape, and the number of parameters. The table also lists the total, trainable, and non-trainable parameter counts for the entire model.

4.1.5.3 XGBoost

As a non-neural alternative, we include XGBoost, a powerful tree-based ensemble learning technique known for its accuracy and efficiency in various classification tasks. XGBoost does not automatically capture temporal dependencies or spatial correlations as neural networks do. But, XGBoost can be advantageous for a variety of reasons: it is typically faster to train on tabular data, easier to interpret at a feature level, and less sensitive to hyperparameter tuning than many deep learning approaches. Although XGBoost does not inherently capture temporal dependencies or spatial correlations as neural networks do, it compensates by effectively combining multiple weak learners (decision trees). Each new tree iteratively refines the predictions of the previous trees, leading to a powerful, gradient-boosted ensemble.

In this way, it can still be applied effectively to raw ECG segments by treating

each segment as a feature vector. Arrhythmia signals likely contain distinct, discriminative patterns, such as specific waveform shapes or abrupt changes, that decision trees can readily identify and separate, even without extensive feature engineering. Additionally, the relatively low dimensionality (n=128) of the data means that the model does not suffer from the curse of dimensionality as much as it might with higher-dimensional raw data, allowing it to perform well without needing hierarchical feature extraction typically associated with deep learning models.

In practice, this often involves flattening each ECG segment into a one-dimensional array so that it can be fed into the tree-based model. Each entry in the vector corresponds to the amplitude of a particular time step in the raw ECG signal. We present the parameters used for XGBoost classifier in Table 4.7

Parameter	**Default Value**
max_depth	3
learning_rate	0.1
n_estimators	100
objective	binary:logistic
booster	gbtree
tree_method	auto
n_jobs	1
gamma	0
min_child_weight	1
subsample	1
colsample_bytree	1
reg_alpha	0
reg_lambda	1
random_state	0

Table 4.7 – Parameters used for XGBoost classifier.

4.1.6 Explainable AI Parameters

In the proposed experimental protocol, three explainable AI techniques —SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model-agnostic Explanations), and Saliency Maps—are employed alongside standard performance metrics.

We adjust the class distributions in the background data used to generate explanations with an equal distribution of classes as sugested by (LIU et al., 2022), which

demonstrates that this strategy mitigates the negative effects of data imbalance. Additionally, with the balancing strategy, the top-ranked variables from the corresponding importance ranking demonstrated improved discrimination power. In this way, a balanced dataset of 30,000 normal heartbeats and 30,000 arrhythmic heartbeats is used to mitigate potential bias arising from class imbalance. The LIME parameter $num_samples$ is set to 2,000, which balances computational overhead with the level of detail provided in local explanations.

SHAP quantifies overall feature contributions to the classifier's predictions, thereby offering a global perspective on which aspects of the heartbeat signals are most relevant. LIME targets individual predictions by constructing local surrogate models, enabling an examination of the factors that drive specific classification outcomes. Saliency Maps highlight the segments of the raw time series that most strongly influence the model's output, which can be compared against known clinical patterns. Taken together, these methods facilitate our evaluation of both global and local properties of the model, thereby clarifying how it discriminates between normal and arrhythmic heartbeats.

4.2 Results

In this section, we present the results of our study, organized according to three main stages of the experimental pipeline. First, we describe the process of generating classification models in which the MIT-BIH and SVDB databases are merged to form a training set, while the INCART dataset is reserved for testing. By doing this, our models remain well-calibrated and minimize overfitting and bias. We evaluated the classification performance on three architectures, XGBoost, FCN, and DeepConvLSTM, laying the foundation for further interpretability analysis.

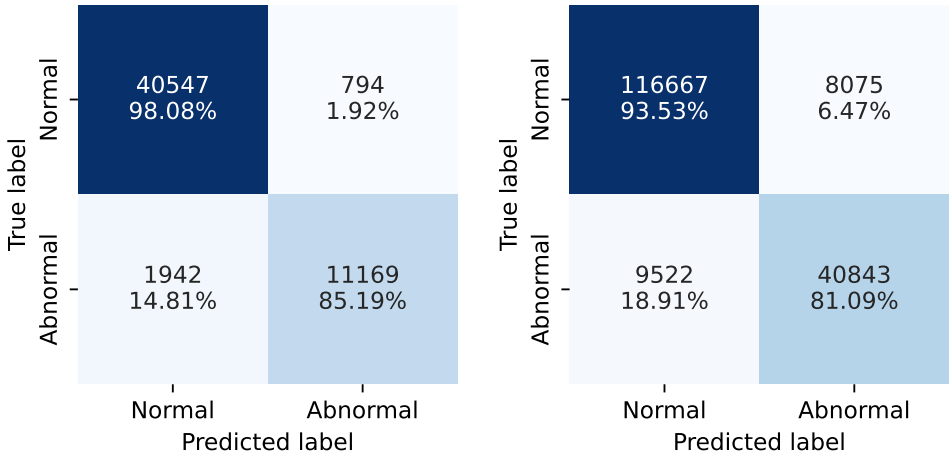
Next, in the global interpretable clustering analysis results, we investigate the feature importance patterns produced by the three models using a range of XAI techniques: the SHAP Explainer (EX), the LIME Tabular Explainer (LTE), Saliency Maps (SM), and the SHAP Tree Explainer (TEX). By applying dimensionality reduction methods such as

PCA, t-SNE, and UMAP, we visualize these importance representations and identify commonalities and differences across XAI approaches.

Finally, we present results for an explainable AI evaluation associated with each model–explanation pairing. These metrics include DTW, MSE, MAE, RMSE, cosine similarity, Euclidean distance, and the SSIM. In addition to these quantitative measures, we consider six metrics: Sanity, Faithful, Sensitivity, Robustness, Stability, and Localization, designed to probe the deeper characteristics and reliability of feature importance maps.

4.2.1 Generating Models for Arrhythmia Classification

The confusion matrices in Figure 4.4 present the performance of XGBoost model for the classification of heartbeats as normal or abnormal (arrhythmia) using two different datasets: MITBIH + SVDB and INCART. For the MITBIH+SVDB dataset, the model demonstrates a high accuracy in identifying normal heartbeats, with a true positive rate (TP) of 98.08% and a false negative rate (FN) of 14.81%. In the INCART dataset, the model's performance slightly decreases, achieving a TP of 93.53%. Nevertheless, the model generates 8,075 false positives, indicating a higher rate of misclassification for normal heartbeats as abnormal compared to the MITBIH+SVDB dataset.



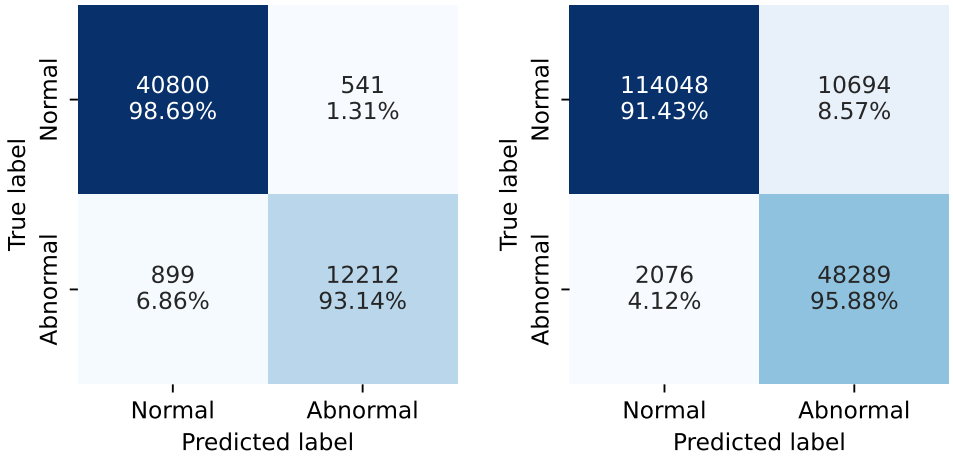
(a) MITBIH + SVDB (b) INCART

Figure 4.4 – XGBoost model results on MITBIH+SVDB (training) and INCART (test) datasets for anomaly heartbeats classification.

The comparative analysis of the two datasets reveals that the XGBoost model

exhibits higher accuracy in identifying normal heartbeats in both datasets, despite a decrease in TP. Also, the model's performance in identifying abnormal heartbeats is reasonably high in both datasets but shows a slight decline in the INCART dataset. Considering that training was performed on the MITBIH+SVDB dataset and tested on INCART, this result proved to be adequate, whose model can consistently learn the features of heartbeats anomalies and classify them correctly in a new data set.

In the next results, presented in Figure 4.5, the confusion matrices illustrate the performance of a DEEPCONVLSTM model. For the MITBIH+SVDB dataset, the model demonstrates high accuracy in identifying normal heartbeats, achieving a TP of 98.69%, resulting in a FP of only 1.31%. For abnormal heartbeats, the model achieves a TN of 93.14%. The FN is relatively low at 6.86%, with 899 abnormal heartbeats misclassified as normal. In the INCART dataset, the DEEPCONVLSTM model maintains high accuracy but shows a slight decrease compared to the MITBIH+SVDB dataset. For abnormal heartbeats, the model achieves a TN of 95.88%, and the FN is 4.12%, with only 2,076 abnormal heartbeats misclassified as normal.



(a) MIT-BIH - SVDB (b) INCART

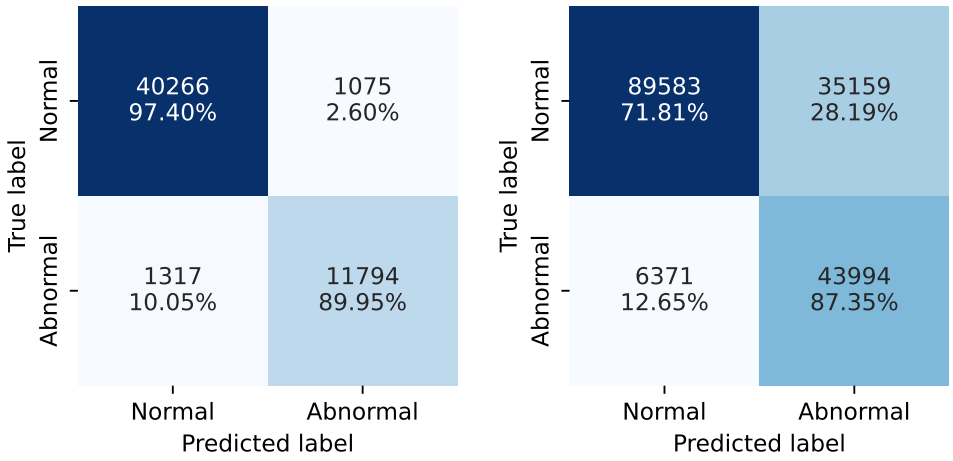
Figure 4.5 – DEEPCONVLSTM model results for training in MITBIH+SVDB dataset and test on INCART dataset.

This result presented in Figure 4.5 shows that the DeepConvLSTM model performance in identifying abnormal heartbeats is better in the INCART dataset, with a higher TN and lower FN compared to the MIT-BIH SVDB dataset. DeepConvLSTM model performs exceptionally well across both datasets, but its performance is influenced by

the features of the data.

Finally, the confusion matrices in Figure 4.6 show the performance of an FCN model. The FCN model shows high accuracy in identifying normal heartbeats in MIT-BIH+SVDB dataset, achieving a TP of 97.40%. For abnormal heartbeats, the model achieves a TN of 89.95%, and FN is 10.05%. For INCART dataset, the performance of the FCN model declines, particularly in identifying normal heartbeats if we compare with XGBoost and DeepConvLSTM models. For abnormal heartbeats, the model achieves a TN of 87.35%, and FN is 12.65%. The substantial decline in performance for normal heartbeat classification in the INCART dataset highlights the need for further refinement of this baseline model.

The higher number of false positives and false negatives in the INCART dataset indicates greater challenges in accurately classifying heartbeats in this dataset.



(a) MIT-BIH - SVDB (b) INCART

Figure 4.6 – FCN results on MITBIH+SVDB (training) and INCART datasets.

4.2.2 Global Interpretable Clustering Analysis

In this section, we present the results of our Global Interpretable Clustering approach using three-dimensionality reduction techniques: UMAP, t-SNE, and PCA applied to the outputs of XGBoost, DeepConvLSTM and FCN models for classifying heartbeats as "normal" or "abnormal." The visualizations are evaluated on varying thresholds (0.5

to 0.9). The threshold filters importance maps, ranging from 0 to 1, to maintain only important features above the defined threshold.

Starting with XGBoost, Figure 4.7 shows that at lower thresholds, UMAP shows distinct clustering, with a stronger separation between normal and abnormal. t-SNE results show more evenly distributed data points compared to UMAP. t-SNE may struggle to retain global structure while emphasizing local neighbor relationships at higher thresholds. The PCA visualizations show smooth, continuous distributions with some visible class separation. The reliance of PCA on linear transformations probably contributes to less effective separability for nonlinear relationships in the data.

Figure 4.8 shows that at lower thresholds (e.g., 0.5 and 0.6), UMAP exhibits distinct clusters, but with some fragmentation between normal and abnormal samples. t-SNE and PCA present highly diffuse clusters across thresholds, with significant overlaps, struggling with global separability, making them less suitable for interpretability in this case.

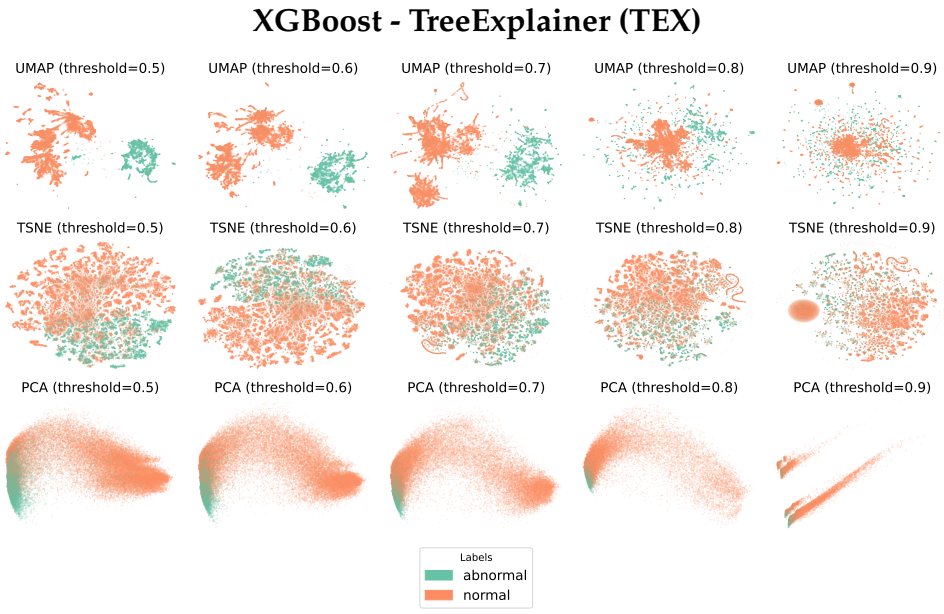


Figure 4.7 – Global Interpretable Clustering method results for XGBoost classifier using TreeExplainer for generate importance maps. We show the comparison of clustering techniques (UMAP, t-SNE, and PCA) across thresholds (0.5–0.9).

The contrasting results observed between the TEX and LTE visualizations stem from fundamental differences in how these techniques calculate and represent feature importance. TEX, which is based on SHAP values, computes feature attributions globally considering all possible combinations of features, providing a consistent and holistic

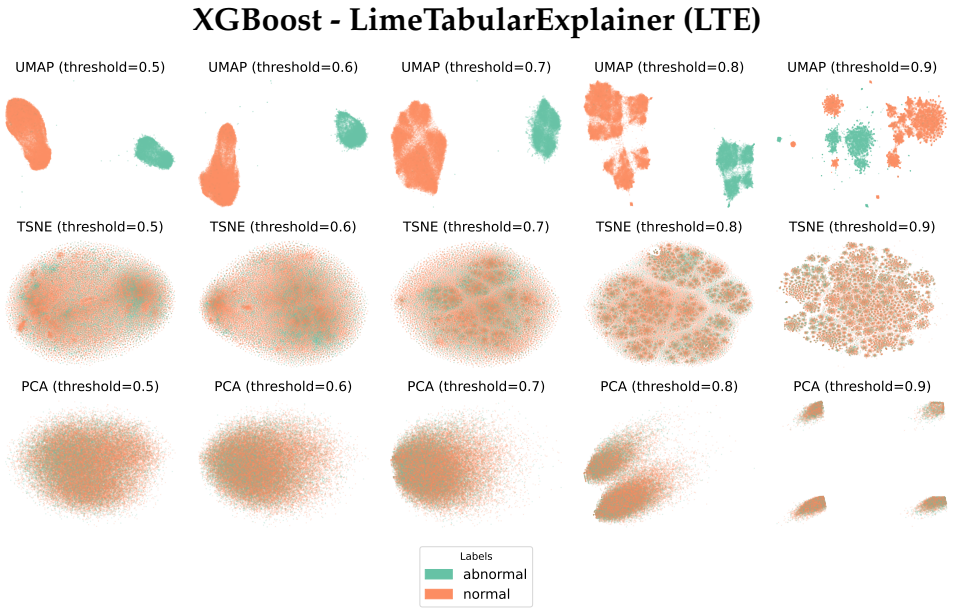


Figure 4.8 – Global Interpretable Clustering method results for XGBoost classifier using LimeTabularExplainer for generate importance maps. We show the comparison of clustering techniques (UMAP, t-SNE, and PCA) across thresholds (0.5–0.9).

explanation of the model's behavior. This global perspective allows TEX to generate feature importance maps that effectively highlight both local and global relationships in the data. In contrast, LTE generates local explanations by approximating the model with a simpler surrogate, such as a linear model, for specific instances. This inherently local focus can result in noisier feature importance maps, which do not capture the broader patterns in the dataset as effectively as SHAP.

These methodological differences have a direct impact on the clustering results. TEX clustering exhibits stronger class separation, particularly when using UMAP and PCA. This is because SHAP provides globally consistent importance values that align well with dimensionality reduction techniques, enabling clearer clustering and better-defined decision boundaries.

In contrast, the LTE shows weaker class separation and increased overlap between clusters. The localized nature of LIME explanations introduces variability and noise in feature importance maps, which impacts the clustering performance of techniques such as t-SNE, UMAP, and PCA. Although higher thresholds improve separation to some extent, the lack of global consistency in LTE feature importance maps limits its effectiveness for global interpretability. Even with UMAP, which performs relatively

well compared to other methods, the clustering remains less distinct than in the TEX results.

The sensitivity of dimensionality reduction techniques further amplifies these differences. UMAP and PCA, which are designed to capture both local and global structures, benefit from SHAP consistent global patterns. Conversely, LIME locally focused importance maps make it harder for these methods to establish clear separations. t-SNE, which emphasizes local neighbor relationships, struggles with both TEX and LTE, but the differences are more pronounced with LIME.

We find similar results for the DeepConvLSTM model, as shown in Figure 4.9 for EX, Figure 4.10 for LTE and Figure 4.11 for SM. The similarity between the results of SM and LTE is derived from their shared focus on localized feature importance. Both methods prioritize explanations based on specific instances rather than providing a globally consistent view of feature relevance across the entire dataset, which inherently limits their ability to capture broader patterns.

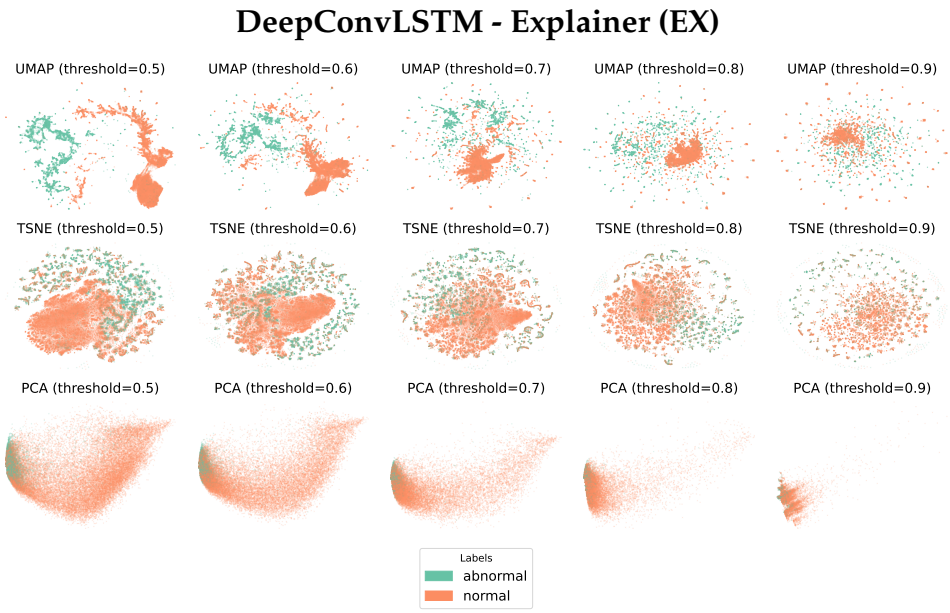


Figure 4.9 – Global Interpretable Clustering method results for DeepConvLSTM classifier using Explainer for generate importance maps. We show the comparison of clustering techniques (UMAP, t-SNE, and PCA) across thresholds (0.5–0.9).

SM, being gradient-based, derives feature importance by analyzing how small changes in input features affect the model's output. This method identifies the features that are most influential for individual predictions, but it does not account for interac-

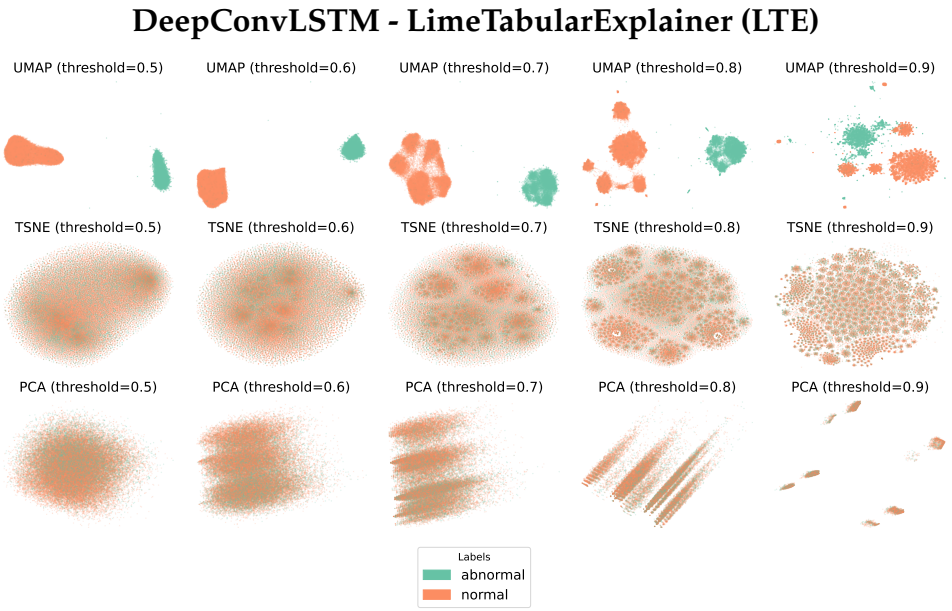


Figure 4.10 – Global Interpretable Clustering method results for DeepConvLSTM classifier using LimeTabularExplainer for generate importance maps. We show the comparison of clustering techniques (UMAP, t-SNE, and PCA) across thresholds (0.5–0.9).

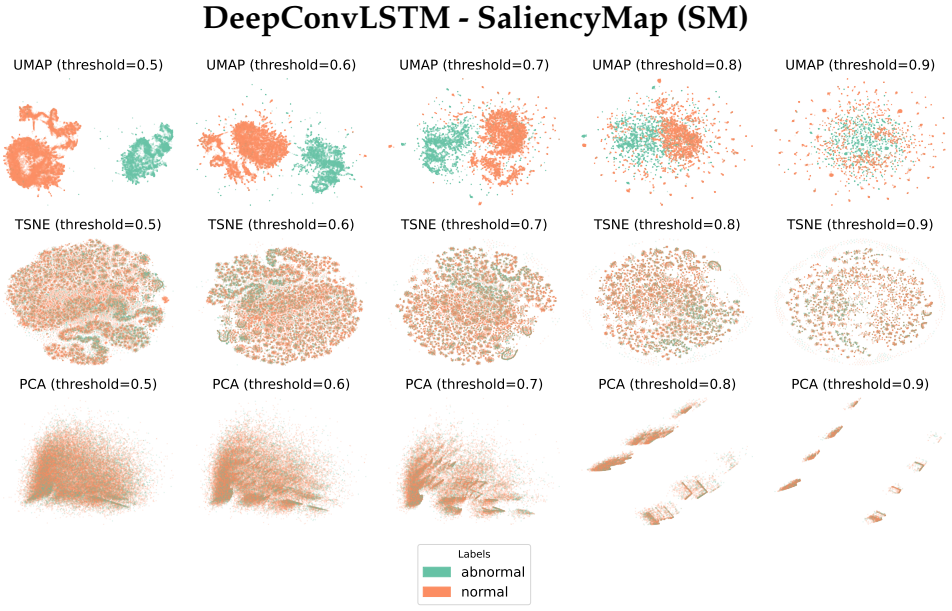


Figure 4.11 – Global Interpretable Clustering method results for DeepConvLSTM classifier using SaliencyMap for generate importance maps. We show the comparison of clustering techniques (UMAP, t-SNE, and PCA) across thresholds (0.5–0.9).

tions between features or the broader global decision structure of the model. Similarly, LTE approximates the model locally by constructing surrogate linear models around specific instances. These surrogate models capture the local decision boundaries but fail to reflect the global relationships in the dataset.

We also notice an interesting trend related to thresholding: as we increase the importance filter, features that are globally relevant but not locally prominent become more influential in defining cluster structure. This is evident in the UMAP and PCA plots, where clusters at higher thresholds appear more compact, showing that the thresholding operation highlights globally consistent feature contributions, refining cluster definitions as the threshold increases.

4.2.3 Explainable AI Evaluation

In this section, we present six important metrics to evaluate XAI methods: sanity, faithfulness, sensitivity, robustness, stability, and localization. We measure how similar or different importance maps are using metrics such as MSE, MAE, RMSE, cosine similarity, Euclidean distance, SSIM and DTW. We use boxplots and confusion matrices to support our analysis.

4.2.3.1 Sanity Evaluation

In this study, sanity metric is used to evaluate the reliability of XAI methods by comparing their importance maps against random importance maps. These random maps are generated using models trained on the same dataset, but with randomized labels to break any meaningful relationships between features and target variables. If an XAI method is truly explaining model behavior, its importance maps should significantly differ from those generated by a model with random labels. Otherwise, the method is likely to capture spurious correlations rather than meaningful explanations.

The models considered for this evaluation include DeepConvLSTM, FCN, and XGBoost, while the XAI methods used are EX, LTE, SM, and TEX. The performance

of these methods is assessed using MSE, MAE, RMSE, Cosine Similarity, Euclidean Distance, and SSIM. Table 4.8 shows the interpretation of the high value and the low value interpretation for each metric. Higher values of MSE, MAE, RMSE, and Euclidean distance indicate that the importance maps generated by the XAI method are significantly different from those of a randomly trained model, suggesting meaningful and reliable explanations. Conversely, higher values of Cosine Similarity and SSIM imply that the explanations closely resemble those of a model trained on randomized labels, indicating unreliable or spurious feature attributions.

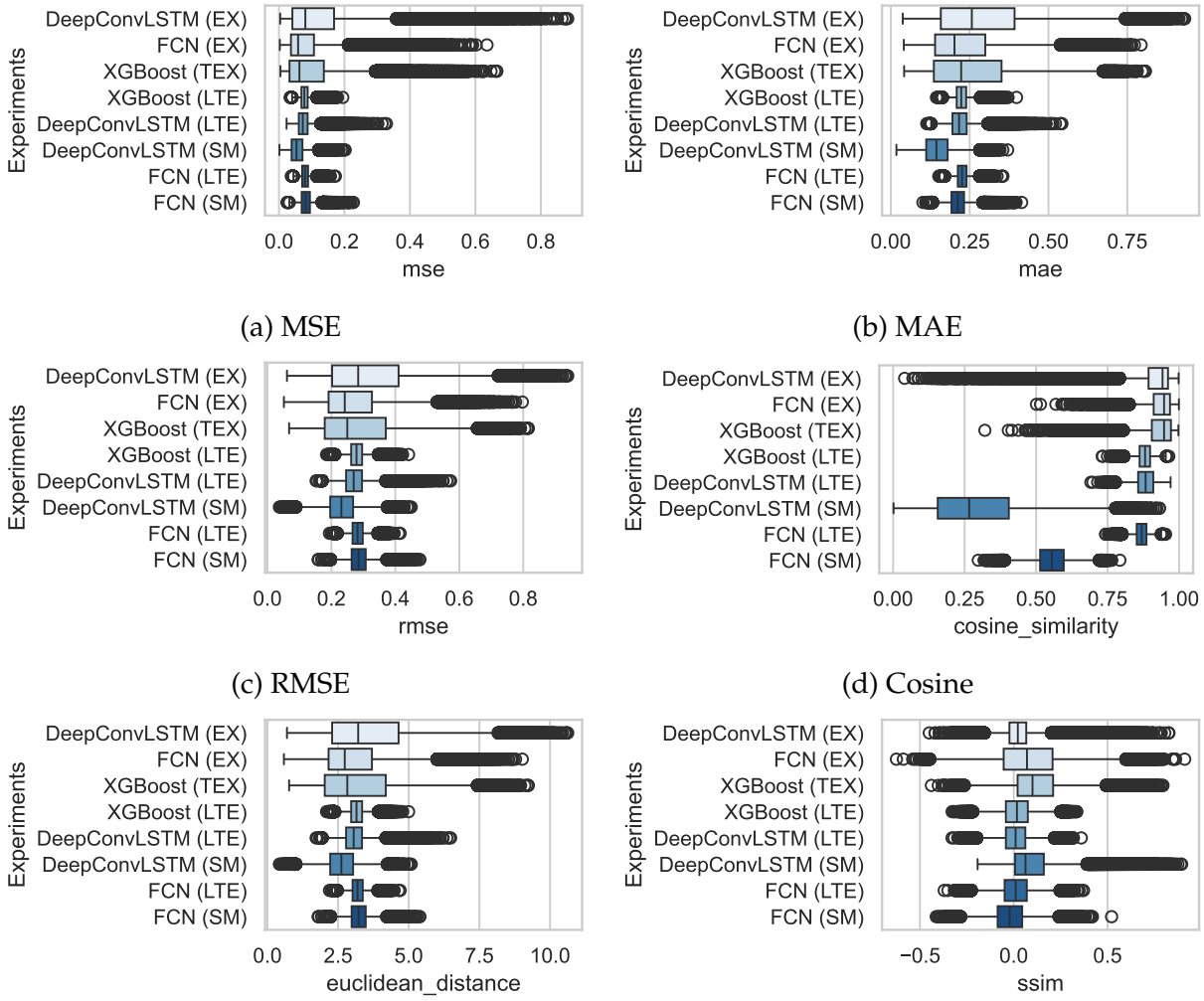
Table 4.8 – Interpretation of high and low values for XAI sanity metric.

Sanity Metric	High Value Meaning	Low Value Meaning
MSE	Explanation differs significantly from random model (good)	Explanation is similar to random model (bad)
MAE	Feature importance maps are distinguishable (good)	Feature attributions are close to random (bad)
RMSE	XAI method produces reliable, distinct explanations (good)	Explanations resemble random attributions (bad)
Cosine Similarity	Explanation aligns with random model (bad)	Explanation differs from random model (good)
Euclidean Distance	Importance maps from real and random models are very different (good)	XAI method produces explanations close to random (bad)
SSIM	Explanation structure is similar to random model (bad)	Explanation structure is different from random model (good)

Boxplots visualization are employed to visualize the distribution and variability of the sanity metric across different setups. We chose boxplot because it provides an overview of the data distribution, central tendency, and variability, making it easier to assess the stability and reliability of experiments.

The results presented in Figure 4.12 show that SM tends to exhibit higher variability for neural models, potentially reflecting the increased complexity of capturing time-dependent or convolutional representations in electrocardiogram signals. On the other hand, TEX shows comparatively robust numerical attributions for XGBoost, indicating a strong match between tree-based decision processes and SHAP's underlying additive feature attribution framework, yet similarity measures (e.g., cosine similarity, SSIM) can fluctuate more, showing that while attributions may be stable in magnitude, their spatial or directional alignment is not always consistent.

LTE shows moderate performance in both neural and tree-based models, but its variance often increases with complexity, consistent with LIME's reliance on local approximation neighborhoods that can become sensitive to small changes in model or data. Meanwhile, the EX displays a stable error profile for neural models showing that it can capture overarching feature importance even as it occasionally struggles to



(a) MSE

(b) MAE

(c) RMSE

(d) Cosine

(e) Euclidean

(f) SSIM

Figure 4.12 – Boxplots of the six similarity metrics —MSE, MAE, RMSE, Cosine Similarity, Euclidean Distance, and SSIM — across DeepConvLSTM, FCN, and XGBoost model architectures and four XAI methods (SHAP Explainer, LIME Tabular Explainer, Saliency Map, SHAP Tree Explainer) on the MIT-BIH Arrhythmia dataset.

preserve fine-grained structural consistency.

Our sanity results confirms that SHAP-based methods (EX, TEX) provide the most robust and reliable explanations, as they consistently differentiate between real and randomized importance maps across all six metrics. LIME (LTE) shows moderate reliability, but its local perturbation-based approach introduces variability, making it less consistent. Saliency Maps (SM) perform the worst, as their feature attributions resemble those from a randomly trained model, indicating that they fail to provide meaningful explanations in this context.

While XAI methods (e.g. SHAP) are popular explanation methods, their output

alone does not inherently evaluate the quality of the explanations. XAI methods produce importance maps but do not quantify how meaningful these scores are compared to random maps. The sanity evaluation allows us to objectively measure whether the importance maps generated are meaningful or just noise. Sanity can expose cases where explanation techniques fail to generate robust feature importance maps. For instance, a high similarity to random maps (low MSE or low Euclidean Distance) indicates that the explanation may lack meaningful information. For deep models , sanity metrics can demonstrate that traditional techniques such as Lime or Saliency Maps often fail to generate reliable explanations due to the models' complexity.

These metrics collectively reveal a consistent pattern: explanation techniques for simpler, structured models such as XGBoost (e.g., TreeExplainer) are robust, while deep models require more sophisticated approaches to generate meaningful feature importance maps. However, no single approach emerges as the universal best performer across every model architecture and every metric.

4.2.3.2 Faithfulness Evaluation

Figure 4.13 shows the faithfulness evaluation for XGBoost which highlights a sharp decline in arrhythmia detection as ablation intensity increases (TEX). The original performance shows high classification accuracy, correctly distinguishing normal and abnormal classes. However, for the TEX method at ablation $intensity = 0.1$ we observe a significant reduction in the number of correctly classified abnormal cases, indicating that key decision-splitting features are being removed. As the intensity of the ablation increases, the rate of misclassification increases, with abnormal rhythms increasingly predicted as normal, demonstrating that the features identified by SHAP methods are critical for model decision making. In contrast, LTE shows a more gradual decline, indicating that their selected features may not fully capture the most discriminative ECG patterns for the detection of arrhythmias. This raises concerns that LIME might be missing key features necessary for reliable classification. The results confirm that SHAP explanations provide a more faithful representation of feature importance, as

their removal leads to severe classification errors.

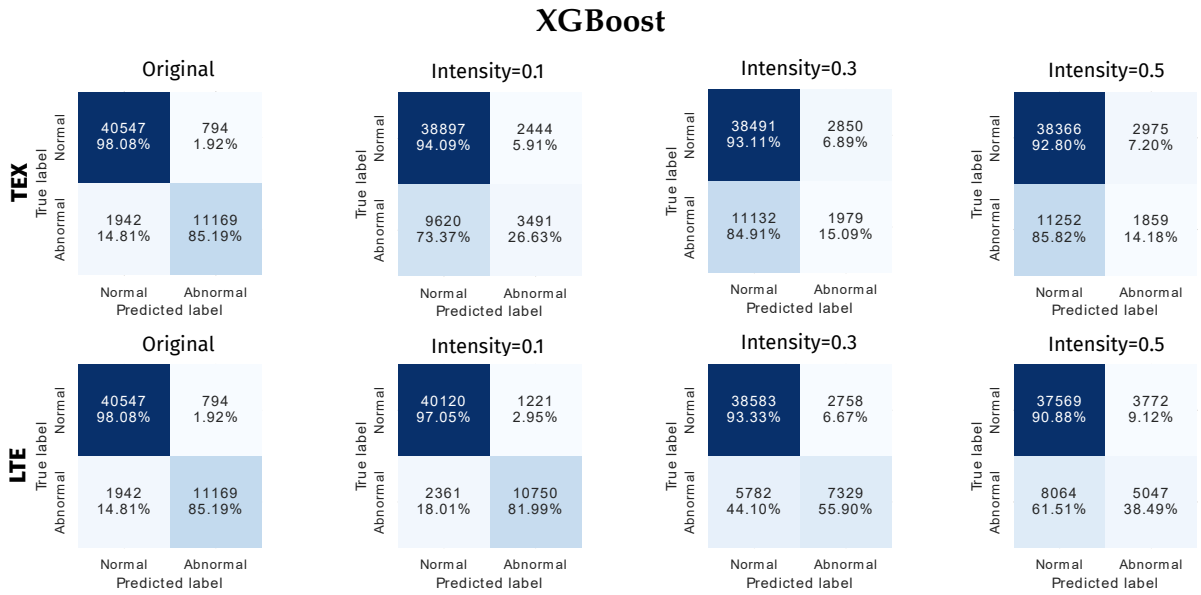


Figure 4.13 – Faithfulness evaluation for the XGBoost model on the MIT-BIH arrhythmia dataset, showing the impact of feature ablation at intensities of 0.1, 0.3, and 0.5. The explainability methods used include SHAP Tree Explainer (TEX), LimeTabularExplainer (LTE), SHAP Explainer (EX), and Saliency Map (SM). As ablation intensity increases, the model's performance deteriorates, with a sharp decline in correctly classified abnormal cases, indicating reliance on the most important features identified by the XAI methods.

Following the next result, Figure 4.14 show that FCN model suffers a greater performance deterioration when high-importance features are removed using (EX). This reinforces the fact that SHAP identifies highly relevant ECG segments. Interestingly, SM ablation shows less performance loss, implying that saliency maps may distribute importance across broader regions rather than pinpointing precise diagnostic features. These results show that while FCN may generalize better than XGBoost at lower ablation intensities, it still critically depends on SHAP-identified features for arrhythmia detection. The fact that LTE and SM ablation cause milder degradation demonstrates that these methods might not fully capture the most essential discriminative features.

Lastly, Figure 4.15 shows that at the original performance levels, DeepConvLSTM achieves high accuracy in detecting arrhythmias, supported by sequential feature learning from the LSTM layers. However, at ablation$intensity = 0.1$, EX begins to show classification errors, with an increasing number of arrhythmias being misclassified as normal. By ablation $intensity = 0.3$, the misclassification rate of abnormal

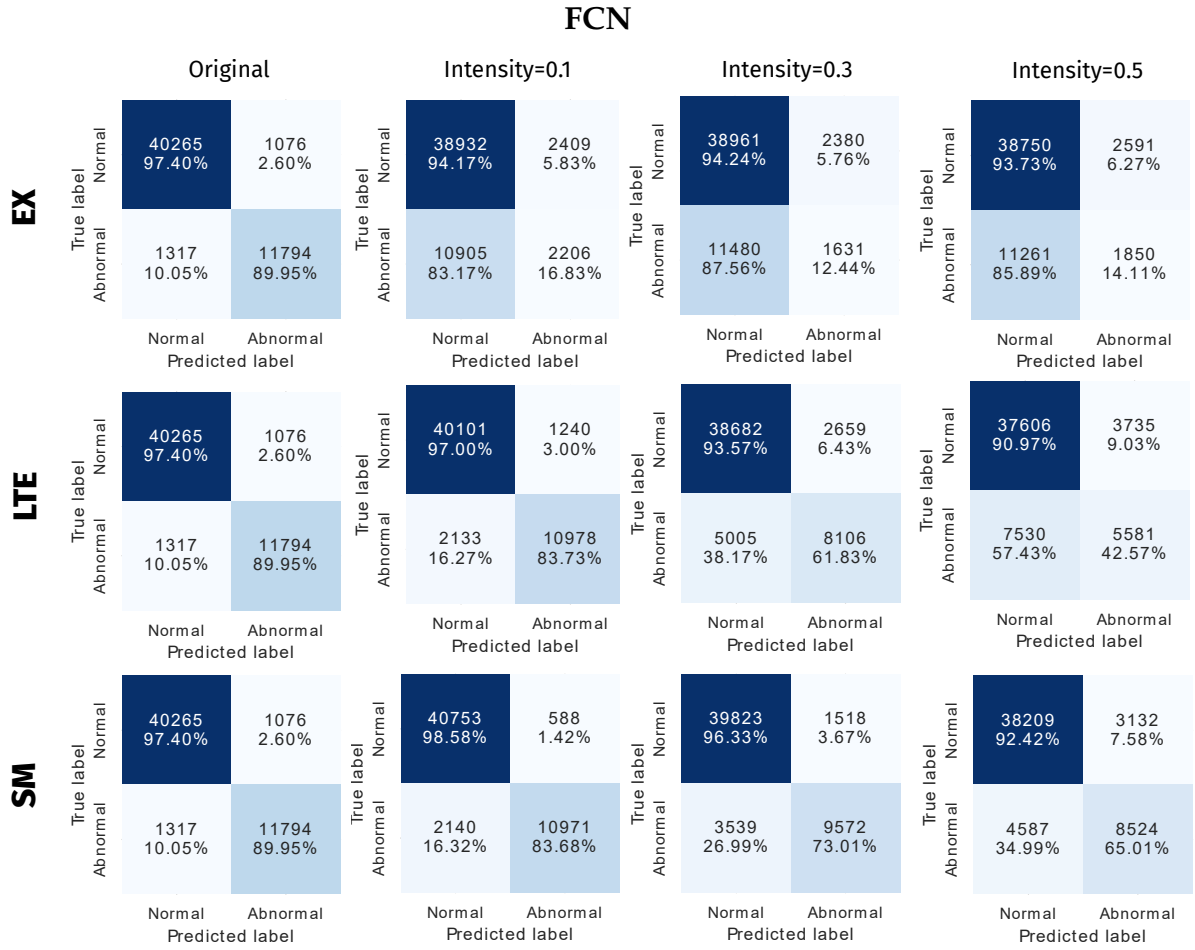


Figure 4.14 – Faithfulness evaluation for the FCN (Fully Convolutional Network) model on the MIT-BIH arrhythmia dataset, demonstrating performance degradation at ablation intensities of 0.1, 0.3, and 0.5. The explainability methods used are SHAP Explainer (EX), LimeTabularExplainer (LTE), and Saliency Map (SM). The performance decline, especially for abnormal rhythm classification, validates the critical role of features highlighted by the explainability techniques.

ECGs increases significantly, particularly for EX, confirming that SHAP-based methods effectively identify highly discriminative features.

4.2.3.3 Robustness Evaluation

The results presented in Figure 4.16 show that XGBoost is the most sensitive to noise, with higher values of MSE, RMSE, and Euclidean distance, demonstrating that its feature attributions are significantly altered when input noise is introduced. The wider spread in cosine similarity provides evidence of inconsistent preservation of feature importance, meaning that in some instances, XGBoost retains its focus on important

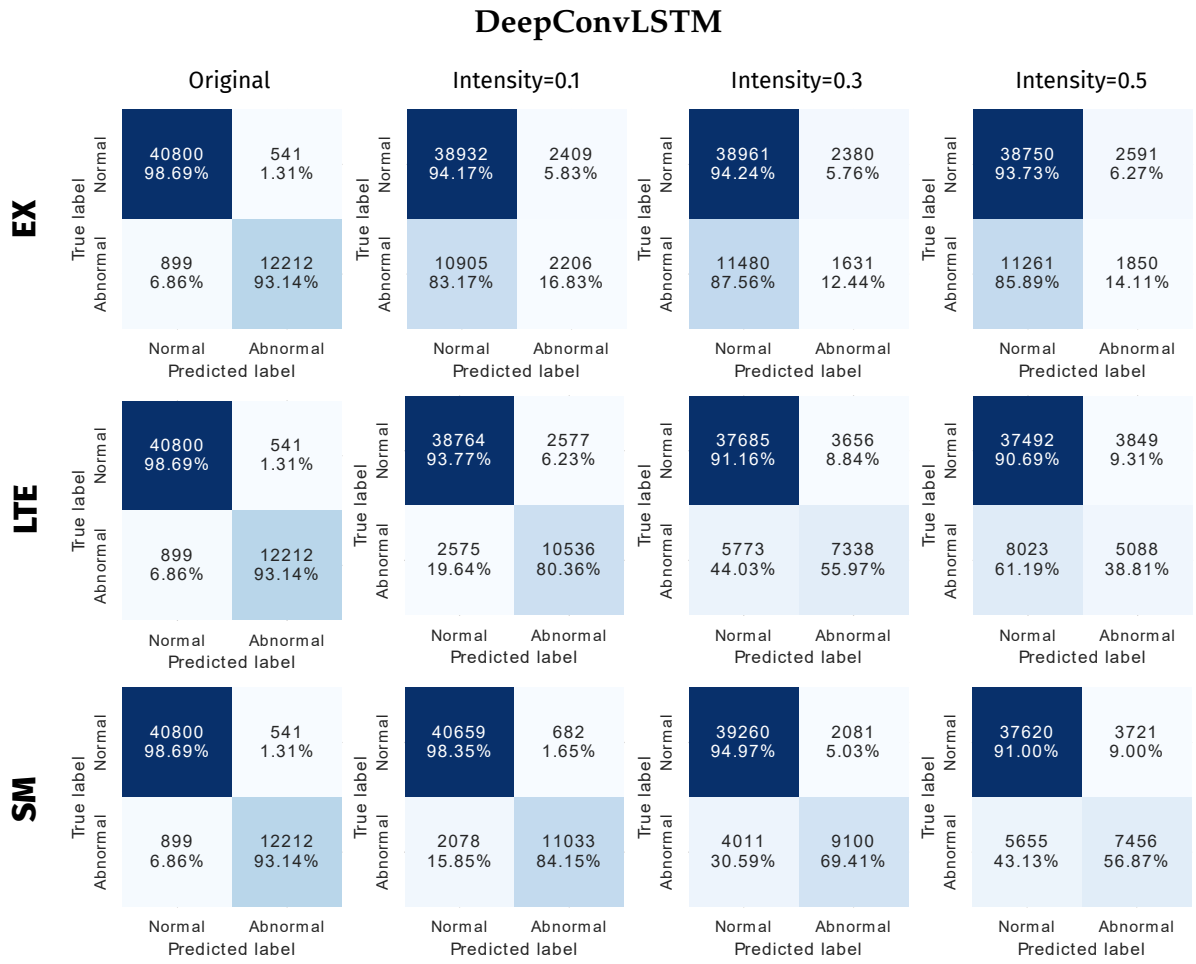


Figure 4.15 – Faithfulness evaluation for the DeepConvLSTM model on the MIT-BIH arrhythmia dataset, highlighting the effect of feature ablation at intensities of 0.1, 0.3, and 0.5. The XAI methods used include SHAP Explainer (EX), LimeTabularExplainer (LTE), and Saliency Map (SM). The model exhibits initial robustness at low intensity but suffers significant performance loss, particularly in abnormal rhythm detection, as more key features are ablated, confirming the importance of the identified features.

features, while in others, noise shifts the importance landscape. Additionally, lower SSIM values in LTE and SM imply that the overall structure of feature importance maps changes substantially, highlighting the model's reliance on discrete feature splits that make it vulnerable to minor input variations.

The FCN model exhibits lower MSE and RMSE values, implying that its feature importance maps remain more stable under noise. The Euclidean distance distribution is more compact, indicating that its attributions do not shift dramatically and that the model distributes feature importance across multiple regions rather than relying on a few key features. Higher Cosine Similarity scores confirm that the model preserves

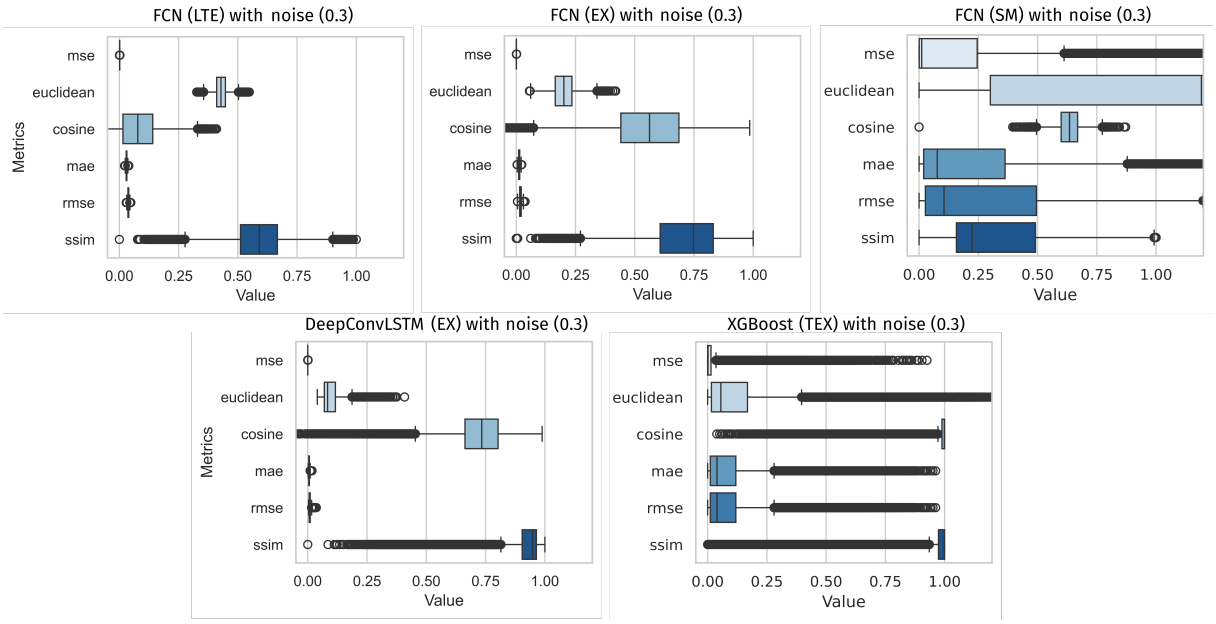


Figure 4.16 – Robustness evaluation of XGBoost, FCN, and DeepConvLSTM models. Noise was applied to the input, and differences in feature importance maps were measured using multiple similarity and distance metrics, including MSE, Euclidean Distance, Cosine Similarity, MAE, RMSE, and SSIM. Higher stability in these metrics indicates that a model's explanations remain consistent despite noise. Results show that DeepConvLSTM is the most robust, maintaining similar feature attributions, while XGBoost is highly sensitive to perturbations, with significant shifts in feature importance.

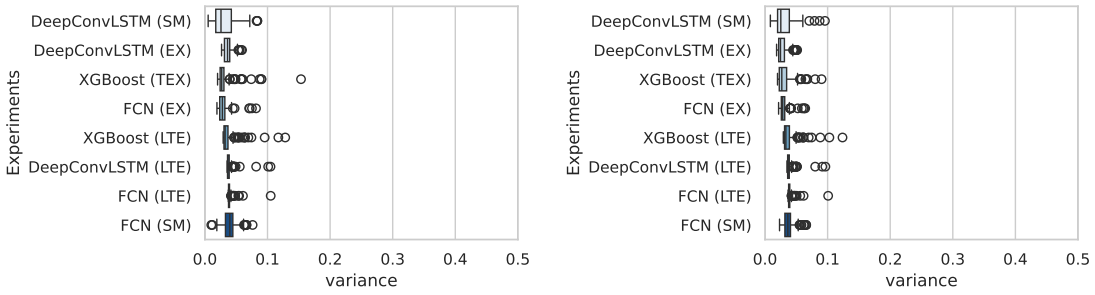
the relative ranking of important features despite noise, while moderate SSIM values indicate that while the structure of importance maps is retained to some extent, localized shifts still occur.

Lastly, DeepConvLSTM demonstrates the highest robustness to noise, as evidenced by lower MSE, RMSE, and Euclidean distance values. This result shows that noise does not significantly alter its feature attributions. The cosine similarity values remain close to 1, confirming that the importance ranking of features is highly preserved, confirming that DeepConvLSTM learns stable sequential patterns that are less affected by local perturbations. Additionally, higher SSIM values indicate that the structural alignment of feature importance maps remains largely unchanged, and the model captures more distributed and resilient features across time-series sequences.

4.2.3.4 Sensitivity Evaluation

The results presented in Figure 4.17 illustrate the variance in feature importance maps separately for normal (a) and abnormal (b) ECGs, measured across different XAI methods and models.

For normal class (Figure 4.17(a)), we observe low variance across all models and explainability methods, indicating that feature importance maps remain fairly stable when identifying normal rhythms. DeepConvLSTM (SM) and (EX) exhibit the lowest variance, indicating that deep learning models assign similar importance to features when analyzing normal heart rhythms. XGBoost (TEX) and FCN (EX) show slightly higher variance, but the overall range remains small, showing that all models have a consistent understanding of normal heartbeats, relying on a relatively fixed set of features for classification.



(a) Normal (b) Abnormal

Figure 4.17 – Class-sensitivity analysis of XGBoost, FCN, and DeepConvLSTM models using different XAI methods. Variance in feature importance maps is measured separately for normal (a) and abnormal (b) classes. Lower variance in the normal class indicates stable feature attributions, while higher variance in the abnormal class suggests increased model adaptation to diverse arrhythmic patterns. SHAP-based explanations (EX, TEX) show more stable class sensitivity, whereas LIME (LTE) and Saliency Maps (SM) introduce greater variability, reflecting more localized feature importance shifts.

For the abnormal class, there is an increase in variance for DeepConvLSTM (SM), XGBoost (LTE), and FCN (LTE). This indicates that feature importance maps for arrhythmia detection are less stable, likely because abnormal heart rhythms exhibit more diverse and complex patterns, which requires the model to adapt its explanations based on the specific ECG characteristics. The higher variance is a sigh that different segments

of the ECG become important depending on the type of arrhythmia, reinforcing the need for models to exhibit class sensitivity.

Interestingly, DeepConvLSTM (EX) and XGBoost (TEX) maintain lower variance compared to other methods, implying that these explainability methods generate more consistent attribute of importance of features for the classification of arrhythmias, potentially capturing robust patterns across different abnormal cases. However, FCN and DeepConvLSTM with Saliency Maps (SM) show the highest variance, which may indicate that these methods dynamically adapt their explanations based on the specific arrhythmia subtype, rather than relying on a fixed set of discriminative features.

4.2.3.5 Stability Evaluation

The results presented in Figure 4.18 show pairwise comparisons of feature maps obtained by XAI methods with different setups. DeepConvLSTM model demonstrates the highest stability, particularly in comparisons between DeepConvLSTM (SM) and DeepConvLSTM (EX), which exhibit lower DTW values. This indicates that feature importance maps remain relatively stable within the same model in different explainability methods. However, compared to XGBoost and FCN, the DTW distances increase, reflecting differences in how deep learning models distribute feature attributions across sequential representations. In contrast, XGBoost shows the highest variability in feature importance attributions, particularly in comparisons such as XGBoost (TEX) vs. XGBoost (LTE) and XGBoost (TEX) vs. DeepConvLSTM (TEX), where higher DTW values indicate that the model's reliance on certain features is highly dependent on the chosen XAI method. This shows that XGBoost's feature attributions are less stable, making its interpretability more sensitive to the choice of the explainability technique.

FCN exhibits moderate stability with relatively consistent importance across different XAI methods, particularly in comparisons of FCN (EX), FCN (SM), and FCN (LTE). However, FCN vs. XGBoost comparisons reveal greater DTW distances, reinforcing the idea that tree-based models and convolutional architectures derive feature importance differently. Furthermore, pairwise comparisons of XAI methods highlight

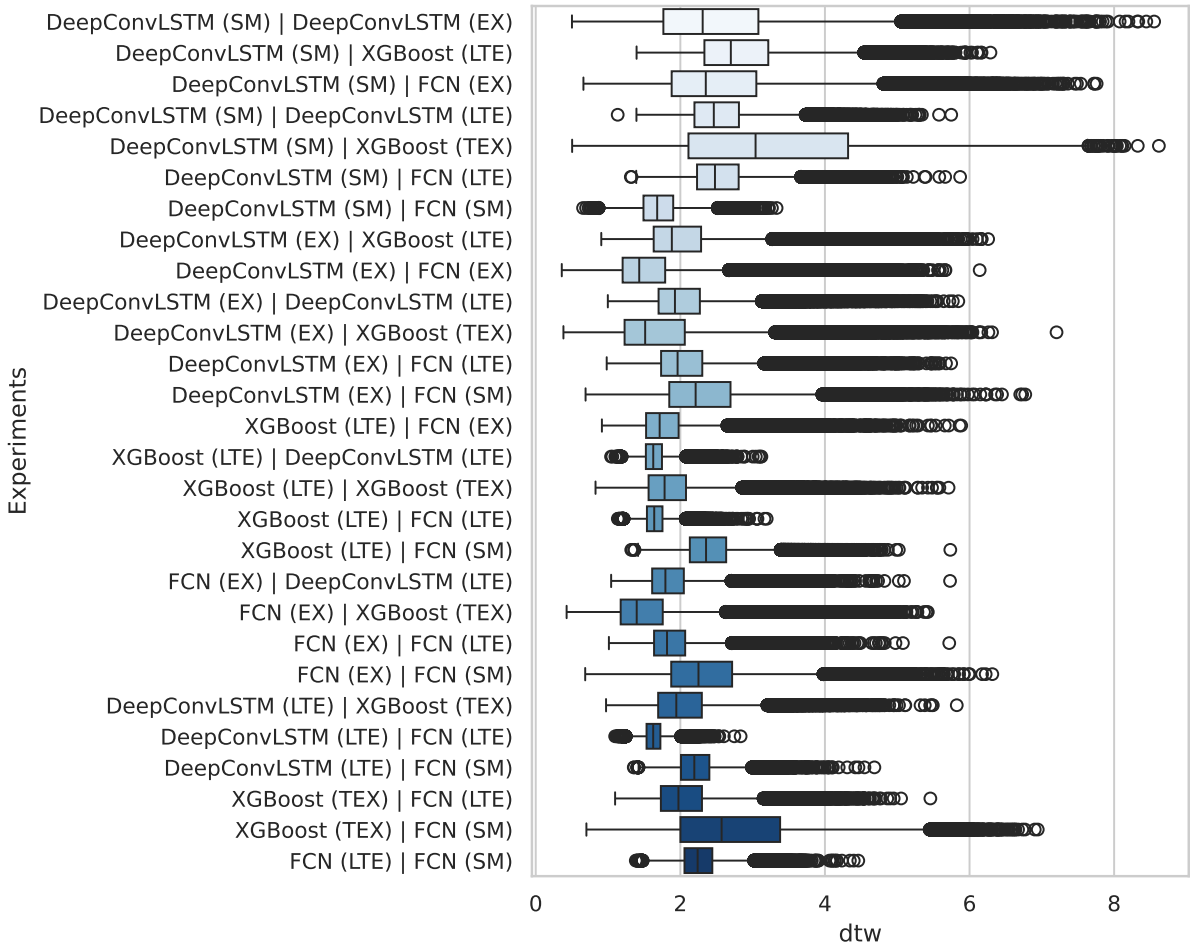


Figure 4.18 – Pairwise comparison of feature importance maps across XGBoost, FCN, and DeepConvLSTM models using different XAI methods (SHAP Tree Explainer (TEX), SHAP Explainer (EX), LimeTabularExplainer (LTE), and Saliency Map (SM)). Dynamic Time Warping (DTW) is used to measure the distance between feature maps, with lower values indicating greater stability. DeepConvLSTM exhibits the highest stability across different XAI methods, whereas XGBoost shows the highest variability, suggesting that explainability results are model-dependent.

that SHAP-based methods (TEX and EX) generate more consistent feature maps across models, whereas LIME (LTE) introduces greater variability, suggesting that LIME's local approximations may be less stable in cross-model comparisons.

4.2.3.6 Localization Evaluation

We use Localization to evaluate whether feature importance maps correctly align with the temporal segments of the input that are most relevant to the predicted classes. This alignment is particularly important in time series data, where domain-specific

knowledge often defines key temporal segments that carry class-specific information. The analysis evaluates the alignment of feature importance maps at two thresholds ($t = 0$ and $t = 0.5$) to assess how well each method captures temporal relevance.

Table 4.9 – Localization results for various models and explanation techniques at two thresholds ($t = 0$ and $t = 0.5$). Higher localization scores indicate better alignment of feature importance maps with the relevant temporal segments of the input.

Experiment	(%) $t = 0$	(%) $t = 0.5$
DeepConvLSTM (SM)	70.02	71.54
DeepConvLSTM (EX)	69.81	77.06
XGBoost (TEX)	69.64	83.05
FCN (EX)	69.69	81.23
XGBoost (LTE)	69.67	70.96
DeepConvLSTM (LTE)	69.64	69.73
FCN (LTE)	69.74	70.44
FCN (SM)	69.93	61.62

The localization results presented in Table 4.9 show that between the evaluated models and the explanation techniques, significant differences were observed. XGBoost (TEX) achieved a significant improvement in localization scores when the threshold was raised from ($t = 0$ to $t = 0.5$), with scores increasing from 69.64 to 83.05. This demonstrates the ability of the method to adapt to the thresholding mechanism and accurately highlight critical temporal features. By focusing on high-importance features, TEX effectively captures relevant data regions, making it a reliable explanation technique for interpretable models. On the other hand, XGBoost (LTE) showed only a modest improvement in scores, from 69.67 at $t = 0$ to 70.96 at $t = 0.5$.

DeepConvLSTM (EX) exhibited strong localization performance. This result suggests that the Explainer method is effective in identifying the most important features within the temporal segments as the threshold increases, making it well-suited for deep learning architectures. In contrast, the SM for DeepConvLSTM showed only a slight improvement in scores reflecting its moderate ability to align with threshold-based localization criteria. DeepConvLSTM (LTE) exhibited minimal differences in scores between. This consistency highlights its difficulty in adapting to thresholding and prioritizing high-impact features.

For FCN (EX), there is a substantial improvement, with scores rising from 69.69

at $t = 0$ to 81.23 at $t = 0.5$. This indicates its strong alignment with the threshold-based localization mechanism, allowing it to focus on the most critical features effectively. However, the SM for FCN performed poorly, with localization scores deteriorating from 69.93 at $t = 0$ to 61.62 at $t = 0.5$. This significant drop highlights the limitations of SM in handling thresholding, as their explanations become less aligned with meaningful temporal features when noise is removed. FCN (LTE) showed minimal improvements.

4.2.4 Final Remarks

Training Appropriated Classification Models: The performance of three different models, XGBoost, DeepConvLSTM, and FCN, for arrhythmia classification reveals some differences and trends. The results reveal several important aspects of model selection, dataset variability, and the potential for clinical application in the arrhythmia classification. The DeepConvLSTM model demonstrated robustness in comparison with XGBoost and FCN models. Its ability to maintain high performance on both the MITBIH+SVDB and INCART datasets shows that the model's architecture and learning mechanisms may be better suited to capture the complex temporal dependencies and nuanced signal patterns inherent in ECG data. DeepConvLSTM low false negative rates indicate that it is less likely to miss truly abnormal heartbeats, an attribute that is critical in clinical practice where missed diagnoses can have severe patient consequences. In contrast, the XGBoost model, while showing strong performance on the MITBIH+SVDB dataset, experiences a measurable performance dip when confronted with the more heterogeneous and possibly more challenging INCART dataset. This discrepancy is expected since tree-based approaches, although powerful and interpretable, may be sensitive to shifts in data distributions or subtle differences in patient populations, signal acquisition methods, or arrhythmia morphologies.

Visualizing Feature Importance Maps with GIC: Our proposed GIC method offers a novel approach for visualizing and analyzing the feature importance maps generated by different XAI methods in time-series classification. By using UMAP, t-SNE, and PCA, GIC facilitates the exploration of how different XAI methods encode and

differentiate feature importance patterns across model architectures. As a visualization tool, we can analyze the stability, coherence, and interpretability of feature importance maps. An analysis of dimensionality reduction techniques shows significant differences in how these methods handle feature importance visualizations. UMAP consistently outperformed PCA and t-SNE, particularly in preserving class separability at higher thresholds. t-SNE, while effective at capturing local relationships, often struggled to retain global structure, leading to less meaningful visual separations. PCA, relying on linear transformations, provided smoother transitions between clusters but exhibited limitations in handling non-linear feature distributions, making it less suitable for complex deep learning models. These results emphasize the importance of selecting appropriate dimensionality reduction techniques based on the properties of the XAI method and the model architecture being analyzed.

A deeper comparison of SHAP, LIME, and Saliency Maps further underscores the advantages of globally consistent feature attribution methods. SHAP-based methods consistently produced the most reliable and well-structured feature importance maps, resulting in clear class separability across all clustering techniques. In contrast, LIME and Saliency Maps demonstrated significant variability, particularly at lower thresholds, suggesting that localized explanations are less stable and less reliable for global interpretability. The ability of SHAP to provide consistent attributions across multiple instances allows for clearer and more interpretable clustering outcomes, whereas LIME and Saliency Maps, by focusing on localized feature importance, introduce noise that impacts the clustering performance of dimensionality reduction techniques. The results confirm that GIC serves as an effective tool for evaluating and comparing XAI methods in time series classification. By integrating GIC into the broader XAI evaluation framework, we gain a deeper understanding of how different explainability methods behave across models, allowing for more informed selection and refinement of interpretability techniques.

Integrating Sanity, Faithfulness, Robustness, Stability, Sensitivity and Localization in XAI Evaluation: we demonstrated how different explainability methods (SHAP, LIME, and Saliency Maps) interact with machine learning models (XGBoost,

FCN, and DeepConvLSTM) applied to our use case dataset of ECG data. Although each metric captures a different aspect of explainability, their combined results highlight key trends in the reliability and interpretability of feature importance attributions.

- The **sanity evaluation** further reinforced the strength of SHAP-based methods, confirming that EX and TEX consistently differentiated real feature importance from randomized maps. LTE showed moderate reliability, but its perturbation-based approach introduced variability, reducing its consistency. SM performed the worst, producing explanations that resembled those of randomly trained models, indicating that they fail to provide meaningful explanations in high-complexity models such as DeepConvLSTM.

- The **faithfulness evaluation** has shown how strongly the removal of high-importance features affected model predictions. The results confirmed that SHAP-based methods (EX, TEX) consistently produced the most faithful explanations, as removing these features led to significant performance degradation, indicating that they truly influenced the model's decision-making process. LTE and SM exhibited lower sensitivity to feature removal, demonstrating that their identified features may be less representative of the model's actual decision process.

- For **robustness evaluation**, the introduction of noise in the input data revealed clear differences in the stability of feature importance maps. DeepConvLSTM exhibited the highest robustness, with minimal changes in feature attributions across explainability methods, suggesting that sequential models learn more resilient patterns. XGBoost showed the greatest sensitivity to noise, particularly under SHAP-based methods (TEX, EX), reflecting its reliance on a small subset of highly important features. LIME and Saliency Maps produced more variable feature attributions, indicating lower robustness to minor input perturbations. These results shows that SHAP is faithful in identifying critical features and this implies in a more sensitivity to perturbations, whereas LIME and SM produce more stable but potentially less informative explanations.

- Class-**sensitivity** analysis shows that feature importance maps differed significantly between normal and abnormal ECG classifications, particularly in deep models. The results showed that SHAP-based methods exhibited lower variance in feature importance across classes, implying that they provided more consistent and generalized explanations. LTE and SM, however, displayed greater variability, particularly in the abnormal ECG class, suggesting that their explanations are more context-dependent. This highlights an important consideration: while some explainability methods remain consistent across all predictions, others dynamically adapt to the complexity of the data, which can be advantageous for detecting irregular patterns such as arrhythmias.

- The **stability** metric examined how well different models and XAI methods produced similar importance maps when trained on the same dataset. DeepConvLSTM demonstrated the highest stability, particularly across SHAP-based methods (EX, TEX), with consistently aligned feature importance maps. XGBoost, however, exhibited high variance, which means that tree-based models depend heavily on the choice of explainability technique. LIME (LTE) showed the greatest inconsistency, implying that its explanations are more dependent on local perturbations rather than globally consistent feature importance.

- The **localization** results demonstrated that because anomaly windows are broad, even modest or noisy feature importance maps are likely to overlap substantially—yielding moderate localization scores (70%) by default.

While SHAP-based techniques emerge as the most reliable across structured and deep models, their robustness must be considered in noisy environments. LIME offers flexibility but suffers from instability, whereas Saliency Maps are largely ineffective in generating meaningful explanations.

5

Conclusions and Future Works

This chapter concludes the thesis by reviewing the work that has been presented, summarizing its primary contributions, and proposing future research directions. The chapter concludes by exploring the broader implications of this work and suggesting future research directions to address these limitations.

5.1 Summary and Contributions

This thesis has contributed to the advancement of the field of explainable artificial intelligence in the time series domain by proposing the Unified Time Series Framework for Explainable Artificial Intelligence (UTS-XAI). The central research question that this thesis addressed was how to improve effective explainability evaluation through the use of domain-specific XAI metrics. Through extensive evaluations, conceptual frameworks, and implementations, this has been addressed and answered.

The primary research question that motivated this thesis was

> *"How can we integrate advanced explainable artificial intelligence methods and improve explainability evaluation into time-series classification to develop robust, trustworthy, and interpretable machine learning models for real-world applications?"*

The responses to this thesis's central question are multifaceted. First, we propose and evaluate the *UTS-XAI* framework, grounded in extensive literature on both explain-

able machine learning and time series classification analysis. The framework provides a structured approach to integrating explainability and evaluation directly into the time series classification pipeline, rather than treating these components as afterthoughts. This principled structure, developed and refined through state-of-the-art research offers a solid theoretical foundation for building future ML systems that demand high predictive accuracy and reliable interpretability.

Second, the thesis introduced and validated a series of specialized tools and methods that manifest the UTS-XAI framework in practice. These include a time series quantitative evaluation methodology for XAI methods, designed to capture the temporal nature of data, and the *Global Interpretable Clustering (GIC)* visualization tool, which allows us to qualitatively compare different explainability approaches.

The results presented in this thesis confirm that integrating advanced explainability methods and rigorous explainability evaluation metrics significantly improves the interpretability and reliability of time series classification models. Through the UTS-XAI framework, we demonstrated that different XAI methods behave differently depending on the model architecture and interpretability criteria, emphasizing the importance of choosing the right explainability approach based on the application domain. We also established that explainability evaluation must go beyond generating feature importance maps: it requires quantitative assessment using faithfulness, robustness, stability, sensitivity, localization, and sanity metrics.

Taken together, the UTS-XAI framework and its associated methods not only serve for the time series classification tasks presented in this thesis but also offer a generalizable foundation for future research.

5.2 Achieved Research Objectives

We present the achieved objectives of this thesis as follows.

- Proposing a Robust Time-Series Classification Pipeline: we successfully implemented and tested XGBoost, FCN, and DeepConvLSTM, covering a diverse range of model architectures suitable for time-series classification. The models were eval-

uated using three real-world datasets from Physionet repository (MIT-BIH, SVDB and INCART), demonstrating their effectiveness in handling medical time-series data. We also validate this pipeline in (BRAGANÇA et al., 2022)

- Developing a Time-Series Explainability Evaluation Methodology: we introduce a domain-specific XAI evaluation framework that quantifies explainability performance using metrics such as faithfulness, sensitivity, robustness, stability, localization, and sanity. We demonstrated that SHAP-based methods provide the most faithful and meaningful explanations, while LIME and Saliency Maps showed limitations in various interpretability criteria. We integrate time-series similarity and distance measures (e.g. MSE, RMSE, Euclidean distance, DTW, and cosine similarity) to assess explanation consistency, further strengthening the evaluation framework.

- Integrating Classification and Explainability into a Unified Framework (UTS-XAI): We successfully combine the classification pipeline with advanced explainability methodology in such a way that model explanations are not only generated, but also systematically validated. Our experiments demonstrated that SHAP-based methods were the most faithful (accurately identifying influential features) but also more sensitive to noise, meaning that their feature importance changed significantly under perturbations. LIME and Saliency Maps were less sensitive to noise but also less faithful, meaning that their explanations remained stable but potentially less meaningful. This trade-off highlights that no single explainability method is universally superior across all criteria.

- Introducing the Global Interpretable Clustering (GIC) Methodology: we developed GIC, which uses dimensionality reduction techniques (PCA, t-SNE, UMAP) to visualize feature importance maps and compare different XAI methods. This methodology allowed us to group explanation patterns and identify similarities and inconsistencies in explainability techniques, providing an intuitive way to assess interpretability at scale.

5.3 Broader Impact and Future Research Perspectives

Although this thesis has made significant progress in integrating explainability into time series classification through the UTS-XAI, several challenges remain open. These challenges highlight opportunities for further refinement, expansion, and real-world applicability of explainable AI techniques for time-series data.

5.3.1 Extending UTS-XAI to Multiclass Time-Series Classification

UTS-XAI framework has been successfully validated in a binary time series classification setting but real-world applications often involve multiclass classification tasks, where models must distinguish between multiple categories rather than just two. Examples include multi-condition medical diagnosis, human activity recognition, fault detection in industrial systems, and financial trend classification, all of which require models to identify and explain differences between three or more possible outcomes. The transition from binary to multiclass classification introduces several challenges, including increased complexity in feature importance attributions, scalability issues in explainability evaluations, and ambiguity in localization metrics, all of which require further research. Future research should focus on extending UTS-XAI to explicitly support multiclass classification by integrating four elements as follows.

- Class-specific faithfulness, sensitivity, and stability evaluations – developing metrics that assess feature importance separately for each class transition rather than averaging across all predictions.
- Multiclass-aware feature attribution and robustness testing – introducing techniques that account for shared and distinct feature importance across multiple categories.
- Improved visualization and interpretability – developing interactive explainability tools that allow users to explore and compare feature attributions across multiple class labels in an intuitive way.

5.3.2 Standardizing Explainability Benchmarks and Community Adoption

Despite growing interest in XAI for time series models, there are no widely accepted benchmark datasets, evaluation metrics, or standard protocols to assess the explainability in temporal applications. This study introduces a novel evaluation methodology, but future research should focus on establishing standardized benchmarks to facilitate fair comparisons between different explainability methods. Developing publicly available time-series explainability datasets and creating community-driven challenges to evaluate XAI metrics would significantly accelerate the adoption of explainable AI in both research and industry.

5.3.3 Adapting UTS-XAI to Other Use Cases and Domains

Although the UTS-XAI framework is demonstrated in the classification of arrhythmias from ECG signals, it is designed to generalize to a wide range of time-series applications. Future work should explore its applicability to sensor-based predictive maintenance, climate modeling, human activity recognition, and fraud detection. Furthermore, integrating new explainability that better suits for its domain should be explored.

5.4 Final Thoughts on Future Directions

The results of this thesis lay the foundation for a more rigorous and structured approach to explainability in time series classification. However, achieving truly trustworthy and universally interpretable AI systems requires continued advancements in adaptive explainability techniques, real-time evaluation, ethical considerations, and human-centered visualization approaches. By addressing these challenges, future research can bridge the gap between theoretical explainability metrics and real-world decision-making that allows time series models to be not only accurate but also interpretable, reliable, and fair for practical deployment.

6

ACKNOWLEDGEMENT

Data Availability

The datasets used in this study come from the PhysioNet collection, which provides a variety of electrocardiogram (ECG) recordings. In particular, we used data from the following databases: the MIT-BIH Arrhythmia Database (PHYSIONET, 2005), the MIT-BIH Supraventricular Arrhythmia Database (PHYSIONET, 1999), and the St. Petersburg INCART Arrhythmia Database (PHYSIONET, 2008). Each dataset contains detailed annotations for different types of heartbeats, classified according to the clinical diagnosis of arrhythmias.

Funding

This research was partially funded by the following sources. The funding bodies had no role in the study design; in the collection, analysis, or interpretation of data; or in the writing of the manuscript.

- **CAPES-PROEX and FAPEAM**: This work was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES-PROEX) – Finance Code 001, and partially supported by the Amazonas State Research Support Foundation (FAPEAM) through the POSGRAD project 2024/2025.
- **Samsung Electronics of Amazônia Ltda.** : In accordance with Article 48 of Decree No. 6.008/2006, this research received partial funding from Samsung Electronics

of Amazônia Ltda. under Federal Law No. 8.387/1991, through Agreement No. 003/2019, signed with ICOMP/UFAM.

- **BASEGRANT Program**: This study was supported by the BASEGRANT program, which aims to train high-quality young talents at the Ph.D. level in Brazil by providing financial support to full-time Ph.D. students in the Computer Science graduate program at the Federal University of Amazonas (PPGI/UFAM).

- **ICOMP/UFAM, Flextronics da Amazônia Ltda., and Motorola Mobility**: Partial funding was also provided as stipulated in Arts. 21 and 22 of Decree No. 10.521/2020, under Federal Law No. 8.387/1991, through Agreement No. 003/2021, signed among ICOMP/UFAM, Flextronics da Amazônia Ltda., and Motorola Mobility Comércio de Produtos Eletrônicos Ltda.

BIBLIOGRAPHY

ADADI, A.; BERRADA, M. Explainable ai for healthcare: from black box to interpretable models. In: SPRINGER. *Embedded systems and artificial intelligence: proceedings of ESAI 2019, Fez, Morocco*. [S.l.], 2020. p. 327–337. 28

ADEBAYO, J. et al. Sanity checks for saliency maps. *Advances in neural information processing systems*, v. 31, 2018. 30, 60, 61

ALIKHADEMI, K. et al. Can explainable ai explain unfairness? a framework for evaluating explainable ai. *arXiv preprint arXiv:2106.07483*, 2021. 53

ANGUITA, D. et al. A public domain dataset for human activity recognition using smartphones. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, v. 3, p. 3, 2013. 48

ARLOT, S.; CELISSE, A. et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, The author, under a Creative Commons Attribution License, v. 4, p. 40–79, 2010. 45, 51

BAER, G. et al. Why do class-dependent evaluation effects occur with time series feature attributions? a synthetic data investigation. *arXiv preprint arXiv:2506.11790*, 2025. 60, 61, 62, 63

BAGNALL, A. et al. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, Springer, v. 31, n. 3, p. 606–660, 2017. 43, 50

BETTINI, C.; CIVITARESE, G.; FIORI, M. Explainable activity recognition over interpretable models. In: IEEE. *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. [S.l.], 2021. p. 32–37. 58

BRAGANÇA, H. et al. How validation methodology influences human activity recognition mobile systems. *Sensors*, MDPI, v. 22, n. 6, p. 2360, 2022. 44, 53, 58, 64, 65, 133

BRAGANçA, H. et al. A smartphone lightweight method for human activity recognition based on information theory. *Sensors*, v. 20, n. 7, 2020. 42, 44, 48

BULLING, A.; BLANKE, U.; SCHIELE, B. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 46, n. 3, p. 1–33, 2014. 52

CEREKCI, E. et al. Quantitative evaluation of saliency-based explainable artificial intelligence (xai) methods in deep learning-based mammogram analysis. *European Journal of Radiology*, Elsevier, v. 173, p. 111356, 2024. 60, 61

CHADDAD, A. et al. Survey of explainable ai techniques in healthcare. *Sensors*, MDPI, v. 23, n. 2, p. 634, 2023. 28, 55

CHEN, R. J. et al. Algorithmic fairness in artificial intelligence for medicine and healthcare. *Nature biomedical engineering*, Nature Publishing Group UK London, v. 7, n. 6, p. 719–742, 2023. 28

DANG, L. M. et al. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, Elsevier, v. 108, p. 107561, 2020. 42

DAS, A.; RAD, P. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371*, 2020. 55

DAS, D. et al. Explainable activity recognition for smart home systems. *arXiv preprint arXiv:2105.09787*, 2021. 58

DEHGHANI, A.; GLATARD, T.; SHIHAB, E. Subject cross validation in human activity recognition. *arXiv preprint arXiv:1904.02666*, 2019. 42

DEMBINSKY, D. et al. Unifying vxai: A systematic review and framework for the evaluation of explainable ai. *arXiv preprint arXiv:2506.15408*, 2025. 61, 62

DOSHI-VELEZ, F.; KIM, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. 54

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern classification*. [S.l.]: John Wiley & Sons, 2000. v. 2. 688 p. 45

FAWAZ, H. I. et al. Transfer learning for time series classification. In: IEEE. *2018 IEEE international conference on big data (Big Data)*. [S.l.], 2018. p. 1367–1376. 104

FAWAZ, H. I. et al. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, Springer, v. 33, n. 4, p. 917–963, 2019. 42, 50, 104

FERRARI, A. et al. Trends in human activity recognition using smartphones. *Journal of Reliable Intelligent Environments*, Springer, v. 7, n. 3, p. 189–213, 2021. 42

GHOLAMIANGONABADI, D.; KISELOV, N.; GROLINGER, K. Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection. *IEEE Access*, IEEE, v. 8, p. 133982–133994, 2020. 45

GUIDOTTI, R. et al. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 51, n. 5, p. 1–42, 2018. 28, 30, 54, 55

HEDSTRÖM, A. et al. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, v. 24, n. 34, p. 1–11, 2023. 61, 62

HOHMAN, F. et al. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE transactions on visualization and computer graphics*, IEEE, v. 25, n. 8, p. 2674–2693, 2018. 28, 29

KNOF, H.; BOERGER, M.; TCHOLTCHEV, N. Quantitative evaluation of xai methods for multivariate time series-a case study for a cnn-based mi detection model. In: SPRINGER. *World Conference on Explainable Artificial Intelligence*. [S.l.], 2024. p. 169–190. 60, 61, 62

KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MONTREAL, CANADA. *Ijcai*. [S.l.], 1995. v. 14, n. 2, p. 1137–1145. 45

LI, F. et al. Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 18, n. 2, p. 679, 2018. 48

LIMA, W. S. et al. Human Activity Recognition based on Symbolic Representation Algorithms for Inertial Sensors. *Sensors*, v. 18, n. 11, p. 1–26, 2018. 42, 48

LIMA, W. S. et al. Human activity recognition using inertial sensors in a smartphone: An overview. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 19, n. 14, p. 3213, 2019. 52

LINARDATOS, P.; PAPASTEFANOPOULOS, V.; KOTSIANTIS, S. Explainable ai: A review of machine learning interpretability methods. *Entropy*, Multidisciplinary Digital Publishing Institute, v. 23, n. 1, p. 18, 2021. 28, 53, 54

LIPTON, Z. C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, ACM New York, NY, USA, v. 16, n. 3, p. 31–57, 2018. 54

LIU, M. et al. Balanced background and explanation data are needed in explaining deep learning models with shap: An empirical study on clinical decision making. *arXiv preprint arXiv:2206.04050*, 2022. 106

LOEFFLER, C. et al. Don't get me wrong: How to apply deep visual interpretations to time series. *arXiv preprint arXiv:2203.07861*, 2022. 60, 61, 62, 63

LONGO, L. et al. Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, Elsevier, v. 106, p. 102301, 2024. 30

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. In: *Proceedings of the 31st international conference on neural information processing systems*. [S.l.: s.n.], 2017. p. 4768–4777. 29, 58, 72

MELIS, D. A.; JAAKKOLA, T. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, v. 31, 2018. 61, 62

MIRZAEI, S. et al. Explainable ai evaluation: a top-down approach for selecting optimal explanations for black box models. *Information*, MDPI, v. 15, n. 1, p. 4, 2023. 30

NWEKE, H. F. et al. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, Elsevier, v. 105, p. 233–261, 2018. 49

ORDÓÑEZ, F. J.; ROGGEN, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, MDPI, v. 16, n. 1, p. 115, 2016. 102

PAWLICKA, A. et al. How explainable is explainability? towards better metrics for explainable ai. In: SPRINGER. *The International Research & Innovation Forum*. [S.l.], 2023. p. 685–695. 28, 30

PAWLICKI, M. et al. Evaluating the necessity of the multiple metrics for assessing explainable ai: A critical examination. *Neurocomputing*, Elsevier, v. 602, p. 128282, 2024. 61, 62

PHYSIONET. *MIT-BIH Supraventricular Arrhythmia Database*. 1999. <https://physionet.org/content/svdb/1.0.0/>. Cambridge, MA, USA. Accessed on 2019. Disponível em: <https://physionet.org/content/svdb/1.0.0/>. 94, 136

PHYSIONET. *MIT-BIH Arrhythmia Database*. 2005. <https://www.physionet.org/content/mitdb/1.0.0/>. Cambridge, MA, USA. Accessed on 2019. Disponível em: <https://www.physionet.org/content/mitdb/1.0.0/>. 94, 136

PHYSIONET. *St. Petersburg INCART Arrhythmia Database*. 2008. <https://physionet.org/content/incartdb/1.0.0/>. St. Petersburg, Russia. Accessed on 2019. Disponível em: <https://physionet.org/content/incartdb/1.0.0/>. 95, 136

REBUFFI, S.-A. et al. There and back again: Revisiting backpropagation saliency methods. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 8839–8848. 30

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. " why should i trust you?" explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 1135–1144. 71

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016. 29, 30, 58

ROJAT, T. et al. Explainable artificial intelligence (xai) on timeseries data: A survey. *arXiv preprint arXiv:2104.00950*, 2021. 28, 30, 55

ROY, C. et al. Explainable activity recognition in videos: Lessons learned. *Applied AI Letters*, Wiley Online Library, p. e59, 2021. 58

RUDIN, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, Nature Publishing Group, v. 1, n. 5, p. 206–215, 2019. 28, 29

RUIZ, A. P. et al. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, Springer, v. 35, n. 2, p. 401–449, 2021. 43, 50

SAMEK, W.; WIEGAND, T.; MÜLLER, K.-R. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017. 30

SARKER, I. H. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, Springer, v. 2, n. 3, p. 160, 2021. 42

SCHÄFER, P. Scalable time series classification. *Data Mining and Knowledge Discovery*, Springer, v. 30, n. 5, p. 1273–1298, 2016. 43

SCHLEGEL, U. et al. Towards a rigorous evaluation of xai methods on time series. In: IEEE. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. [S.l.], 2019. p. 4197–4201. 30, 60, 61, 62, 63

SCHLEGEL, U.; KEIM, D. A. Time series model attribution visualizations as explanations. In: IEEE. *2021 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*. [S.l.], 2021. p. 27–31. 29, 30

SCHLEGEL, U.; KEIM, D. A. A deep dive into perturbations as evaluation technique for time series xai. In: SPRINGER. *World conference on explainable artificial intelligence*. [S.l.], 2023. p. 165–180. 61, 63

SCHLEGEL, U. et al. An empirical study of explainable ai techniques on deep learning models for time series tasks. *arXiv preprint arXiv:2012.04344*, 2020. 30

SERRAMAZZA, D. I. et al. Evaluating explanation methods for multivariate time series classification. In: SPRINGER. *International Workshop on Advanced Analytics and Learning on Temporal Data*. [S.l.], 2023. p. 159–175. 60, 61, 62, 63

SHEU, R.-K.; PARDESHI, M. S. A survey on medical explainable ai (xai): Recent progress, explainability approach, human interaction and scoring system. *Sensors*, MDPI, v. 22, n. 20, p. 8068, 2022. 55

SHOAIB, M. et al. Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 14, n. 6, p. 10146–10176, 2014. 48

SHOAIB, M. et al. A survey of online activity recognition using mobile phones. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 15, n. 1, p. 2059–2085, 2015. 42, 44, 48

SIMONYAN, K.; VEDALDI, A.; ZISSERMAN, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013. 69

SOKOL, K.; VOGT, J. E. What does evaluation of explainable artificial intelligence actually tell us? a case for compositional and contextual validation of xai building blocks. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. [S.l.: s.n.], 2024. p. 1–8. 28, 30, 61, 62

THEISSLER, A. et al. Explainable ai for time series classification: A review, taxonomy and research directions. *IEEE Access*, IEEE, 2022. 55

TOREINI, E. et al. The relationship between trust in ai and trustworthy machine learning technologies. In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. [S.l.: s.n.], 2020. p. 272–283. 53

UDDIN, M. Z.; SOYLU, A. Human activity recognition using wearable sensors, discriminant analysis, and long short-term memory-based neural structured learning. *Scientific Reports*, Nature Publishing Group, v. 11, n. 1, p. 1–15, 2021. 58

VEALE, M.; BINNS, R. Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. *Big Data & Society*, SAGE Publications Sage UK: London, England, v. 4, n. 2, p. 2053951717743530, 2017. 53

WANG, J. et al. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, Elsevier, v. 119, p. 3–11, 2019. 42, 44

WONG, T. T. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, Elsevier, v. 48, n. 9, p. 2839–2846, 2015. 45, 51

YEH, C.-K. et al. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, v. 32, 2019. 60, 61

YIN, M.; VAUGHAN, J. W.; WALLACH, H. Understanding the effect of accuracy on trust in machine learning models. In: *Proceedings of the 2019 chi conference on human factors in computing systems*. [S.l.: s.n.], 2019. p. 1–12. 53