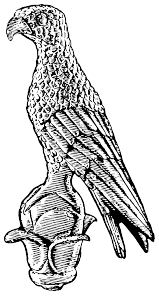


UNIVERSITY OF IOANNINA
SCHOOL OF INFORMATICS AND TELECOMMUNICATIONS
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Advanced Machine Learning Techniques for Data Analysis and Processing

Τίτλος στα Ελληνικά
«Προηγμένες Τεχνικές Μηχανικής Μάθησης για την Ανάλυση και Επεξεργασία Δεδομένων»

by
Lamprini Pappa

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy
in the
School of Informatics and Telecommunications
Department of Informatics and Telecommunications
University of Ioannina

Arta 2026

Application Date of Ms. Lamprini Pappa: 15-11-2022

Approval Date of Advisory Committee: 89/14-12-2022

Advisory Committee Members:

Supervisor

Petros Karvelis, Associate Professor, Department of Informatics and Telecommunications, University of Ioannina, Greece

Members

Chrysostomos Stylios, Professor, Department of Informatics and Telecommunications, University of Ioannina, Greece

Aristides Lalos, Research Director, Industrial Systems Institute (ISI), Athena Research Center in Information, Communication and Knowledge Technologies, Greece

Title: "Advanced Machine Learning Techniques for Data Analysis and Processing"

Appointment of Examination Committee: 189/06-07-2026

Petros Karvelis	Associate Professor, Department of Informatics and Telecommunications, University of Ioannina, Greece
Chrysostomos Stylios	Professor, Department of Informatics and Telecommunications, University of Ioannina, Greece
Aristides Lalos	Research Director, Industrial Systems Institute (ISI), Athena Research and Innovation Center in Information, Communication and Knowledge Technologies, Greece
Dimitrios Iakovidis	Professor, Department of Computer Science and Biomedical Informatics, University of Thessaly, Greece
Georgios Manis	Associate Professor, Department of Computer Science and Engineering, University of Ioannina, Greece
Alexandros Tzallas	Professor, Department of Informatics and Telecommunications, University of Ioannina, Greece
Ioannis Tsoulos	Professor, Department of Informatics and Telecommunications, University of Ioannina, Greece

Approval of the PhD Thesis with Distinction on 29-07-2026

Arta, 29-07-2026

HEAD OF THE DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Alexandros Tzallas
Professor

Department Secretary

Evangelia Christou

This page has been left blank intentionally.

Abstract

Time series classification is a fundamental machine learning task with applications spanning healthcare monitoring, human activity recognition, industrial fault detection, and financial analysis. Among the many approaches developed for this task, symbolic representations occupy a distinctive position: by discretizing continuous signals into sequences of discrete tokens, they offer a combination of computational efficiency, noise robustness, and interpretability that purely numerical or deep-learning approaches rarely match. The central technique in this family is Symbolic Aggregate approXimation (SAX), which reduces a time series to a compact symbolic string via Piecewise Aggregate Approximation (PAA) and Gaussian breakpoint discretization. Despite its two-decade prevalence, SAX and the Bag-of-Patterns (BOP) classifier built on top of it carry several well-documented weaknesses: parameter selection is typically heuristic, trend information is discarded by the piecewise averaging step, cross-channel dependencies are not natively captured in multivariate settings, sequential order between symbolic words is lost in BOP's frequency histogram, and the symbolic vocabulary has not previously been exploited as a first-class input to modern sequence models.

This thesis develops a coherent programme of extensions that address these weaknesses systematically, while preserving the interpretability that motivates the symbolic approach in the first place. The thesis begins by conducting a systematic review of the SAX literature and organizing the identified extensions into a taxonomy structured around the specific weakness each addresses, establishing the conceptual map from which the original contributions depart.

The first original contribution introduces two independent extensions targeting separate weaknesses: Multichannel Intelligent Icons (MII), which captures cross-channel dependencies in multivariate time series, and Slopewise Aggregate Approximation (SAA), which recovers trend information through a parallel angle-based discretization branch. Both are evaluated on a human activity recognition benchmark.

The second contribution addresses the parameter selection problem by applying the DIRECT global optimization algorithm to the joint tuning of SAX parameters across four feature-extraction strategies, evaluated on twenty-four UEA multivariate benchmark datasets. The proposed approach achieves competitive classification accuracy without gradient information or exhaustive search, while maintaining full symbolic interpretability.

The third contribution bridges symbolic time series representations and Large Language Models (LLMs). Observing that SAX already produces ordered discrete token sequences – the native input format of LLMs – the thesis develops SAX_HAR-LLM, a dual-stream framework that pairs a trend-aware case-modulated symbolic encoding with lightweight kinematics-informed descriptors and fine-tunes a causal LLM via QLoRA on structured prompts derived from inertial sensor data. The framework is evaluated on two human activity recognition benchmarks and includes

an attention-based interpretability analysis that links predictions to specific symbolic motifs on specific sensor axes.

The fourth contribution addresses BOP's loss of sequential order by constructing directed graphs over SAX-word transitions and learning graph embeddings via Graph Neural Networks. Two graph construction schemes are proposed and evaluated on twelve UCR univariate benchmark datasets, establishing that graph-based representations complement BOP's frequency information most effectively on series compressed into longer symbolic strings.

Taken together, these contributions constitute a systematic and empirically grounded exploration of symbolic time series representation, advancing the state of the art along four distinct dimensions while maintaining a coherent commitment to interpretability throughout.

Keywords: Time Series Classification, Symbolic Representation, Symbolic Aggregate approXimation (SAX), Bag-of-Patterns (BOP), Multivariate Time Series, Hyperparameter Optimization, DIRECT algorithm, Large Language Models (LLMs), Graph Neural Networks, Interpretability

Εκτεταμένη Σύνοψη

Η ταξινόμηση χρονοσειρών αποτελεί θεμελιώδες πρόβλημα μηχανικής μάθησης με εφαρμογές που εκτείνονται από την παρακολούθηση υγείας και την αναγνώριση ανθρώπινης δραστηριότητας έως τη βιομηχανική ανίχνευση σφαλμάτων και την οικονομική ανάλυση. Μεταξύ των διαθέσιμων προσεγγίσεων, οι συμβολικές αναπαραστάσεις κατέχουν ξεχωριστή θέση: μετασχηματίζοντας συνεχείς χρονοσειρές σε ακολουθίες διακριτών συμβόλων, προσφέρουν συνδυασμό υπολογιστικής αποδοτικότητας, ανθεκτικότητας σε θόρυβο και ερμηνευσιμότητας που αριθμητικές ή βαθιές αρχιτεκτονικές σπάνια επιτυγχάνουν ταυτόχρονα. Η κεντρική τεχνική αυτής της οικογένειας είναι η Συμβολική Συναθροιστική Προσέγγιση (Symbolic Aggregate approXimation - SAX), η οποία συμπυκνώνει μια χρονοσειρά σε συμβολική ακολουθία μέσω της Τμηματικής Συναθροιστικής Προσέγγισης (Piecewise Aggregate Approximation - PAA) και γκαουσιανής διακριτοποίησης. Παρά τη διάδοσή της για δύο δεκαετίες, η SAX και το μοντέλο Bag-of-Patterns (BOP) που βασίζεται σε αυτή παρουσιάζουν γνωστούς περιορισμούς: η επιλογή παραμέτρων γίνεται συνήθως ευρετικά, η πληροφορία τάσης χάνεται κατά την τμηματική προσέγγιση, οι εξαρτήσεις μεταξύ καναλιών δεν αποτυπώνονται στο πολυδιάστατο πλαίσιο, η χρονική διάταξη των συμβολικών λέξεων χάνεται στο ιστόγραμμα συχνότητας του BOP, και το συμβολικό λεξιλόγιο δεν έχει αξιοποιηθεί ως είσοδος σύγχρονων μοντέλων ακολουθιών.

Η παρούσα διατριβή αναπτύσσει ένα συνεκτικό πλαίσιο επεκτάσεων που αντιμετωπίζουν συστηματικά αυτούς τους περιορισμούς, διατηρώντας παράλληλα την ερμηνευσιμότητα που αποτελεί τον κεντρικό λόγο υιοθέτησης της συμβολικής προσέγγισης. Η διατριβή ξεκινά με συστηματική ανασκόπηση της βιβλιογραφίας SAX και οργάνωση των επεκτάσεων σε ταξινομία βάσει της αδυναμίας που η κάθε μία αποσκοπεί να αντιμετωπίσει.

Η πρώτη πρωτότυπη συνεισφορά εισάγει δύο ανεξάρτητες επεκτάσεις: τα Multichannel Intelligent Icons (MII), που αποτυπώνουν συνεξαρτήσεις μεταξύ καναλιών σε πολυδιάστατες χρονοσειρές, και την Slopewise Aggregate Approximation (SAA), που ανακτά πληροφορία τάσης μέσω παράλληλου κλάδου γωνιακής διακριτοποίησης. Και οι δύο αξιολογούνται σε πρόβλημα αναγνώρισης ανθρώπινης δραστηριότητας.

Η δεύτερη συνεισφορά αντιμετωπίζει το πρόβλημα επιλογής παραμέτρων εφαρμόζοντας τον αλγόριθμο καθολικής βελτιστοποίησης DIRECT για κοινή βελτιστοποίηση παραμέτρων SAX σε τέσσερις στρατηγικές εξαγωγής χαρακτηριστικών, με αξιολόγηση σε εικοσιτέσσερα σύνολα δεδομένων UEA.

Η τρίτη συνεισφορά συνδέει τις συμβολικές αναπαραστάσεις χρονοσειρών και τα μεγάλα γλωσσικά μοντέλα (LLM). Παρατηρώντας ότι η SAX παράγει ήδη ταξινομημένες διακριτές ακολουθίες που αποτελούν την εγγενή μορφή εισόδου των LLM, αποφασίσαμε να αξιοποιήσουμε αυτή τη δομική συμβατότητα για αναγνώριση δραστηριότητας από αισθητήρες. Αναπτύξαμε την μέθοδο SAX_HAR-LLM, στην οποία εισαγάγαμε κωδικοποίηση τάσης μέσω εναλλαγής πεζών-κεφαλαίων συμβόλων και συνδυάσαμε τις συμβολικές

ακολουθίες με κινηματικές παραμέτρους σε δομημένες εντολές προς το LLM (prompts) για την εξειδίκευση (fine-tuning) προεκπαιδευμένου LLM. Η μέθοδος αξιολογείται σε δύο σύνολα δεδομένων αναφοράς για αναγνώριση ανθρώπινης δραστηριότητας και περιλαμβάνει ανάλυση ερμηνευσιμότητας, η οποία συνδέει τις προβλέψεις με συγκεκριμένα συμβολικά μοτίβα σε συγκεκριμένους άξονες αισθητήρων.

Η τέταρτη συνεισφορά αντιμετωπίζει την απώλεια αλληλουχιακής πληροφορίας στο BOP κατασκευάζοντας κατευθυνόμενους γράφους μεταβάσεων μεταξύ συμβολικών λέξεων και μαθαίνοντας αναπαραστάσεις γράφου μέσω Γραφικών Νευρωνικών Δικτύων (Graph Neural Networks – GNNs), με αξιολόγηση σε δώδεκα σύνολα δεδομένων UCR.

Στο σύνολό τους, οι συνεισφορές αυτές αποτελούν μια συστηματική και εμπειρικά τεκμηριωμένη διερεύνηση της συμβολικής αναπαράστασης χρονοσειρών, που επεκτείνει τα όρια του πεδίου σε τέσσερις διακριτές κατευθύνσεις χωρίς να θυσιάζει την ερμηνευσιμότητα.

Λέξεις-κλειδιά: Ταξινόμηση χρονοσειρών, Συμβολική Αναπαράσταση, Συμβολική Συναθροιστική Προσέγγιση (SAX), αναπαράσταση Bag-of-Patterns (BOP), Πολυδιάστατες Χρονοσειρές, Βελτιστοποίηση Υπερπαραμέτρων, Αλγόριθμος DIRECT, Μεγάλα Γλωσσικά Μοντέλα (LLMs), Νευρωνικά Δίκτυα Γράφων (GNNs), Ερμηνευσιμότητα

This page has been left blank intentionally.

Acknowledgments

A long journey reaches its end with humility and modesty. This path was never a firework. Patience, persistence, and methodical thinking are the virtues that carried me through to its completion. I emerge feeling more scientifically ignorant than when I began – a testament to the inexhaustible depth of knowledge and research – and I embrace this as a grounding truth that leaves me with the same thirst for exploring new horizons.

This work would not have been possible without the people who accompanied me along the way.

My deepest gratitude goes first to Prof. Chrysostomos Stylios, who was the first to believe in me. His experience, enthusiasm, optimism, and ambition drew me toward higher goals and secured the successful completion of this journey. I will never forget his words of advice: *"Starting a PhD does not mean you will ultimately conclude it"* – a reminder that enrollment confers no guarantee, and that the distance between beginning and finishing must be earned at every step; *"It is your PhD – you trace the research path"*; *"Do not try to provide solutions for everything or to find a panacea method. Just a little piece each time."* These were not instructions but anchors, and I returned to them often.

I am equally grateful to my supervisor, Prof. Petros Karvelis, who has been my mentor from the very first day I set foot in this laboratory – long before the idea of a PhD had taken shape. When the time came, he guided me through the academic paths that followed with a rare combination of freedom and trust. We searched for new research directions together, spoke to each other with honesty, and built a relationship of mutual respect that genuinely fostered my growth. During the hardest periods, he never let me give up. For all of this, I am profoundly grateful.

To my colleagues at the Laboratory of Knowledge and Intelligent Computing – you have become something of a family to me. The daily life of research is shaped by the people you share it with, and I was fortunate in this regard.

I am particularly grateful to Dr. Marios Tyrovolas, whose constructive conversations and invaluable advice on the manuscript writing process proved truly useful. He also generously shared the LaTeX template on which this thesis is built, saving a great deal of time and frustration during the writing stage.

To Mary – she was there for the good and the bad. She listened to my complaints, my fears, my anxieties, and my fatigue, without ever tiring of them. She rejoiced genuinely at my successes, and that kind of presence is rarer and more valuable than it might seem.

To my brother – the most lasting thing he gave me was a way of thinking, and I doubt he even knows he gave it. He has been truly proud of me throughout, and that too has meant more than he knows.

To my little friend Orestes – he showed me, at exactly the right moment of

indecision, that starting a PhD was the right choice. In his own way, he set things in motion.

Finally, to all my friends who patiently answered my countless questions and endured my endless *whys* – thank you for indulging my curiosity.

Lamprini Pappa
July 2026

Contents

This page has been left blank intentionally.

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
BFS	Breadth-First Search
BOP	Bag-of-Patterns
CD	Critical Difference
CNN	Convolutional Neural Network
CoT	Chain-of-Thought
DGE	Distance-Aware Graph Embeddings
DTW	Dynamic Time Warping
GAN	Generative Adversarial Network
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GPU	Graphics Processing Unit
HAR	Human Activity Recognition
IoT	Internet of Things
LLM	Large Language Model
LoRA	Low-Rank Adaptation
LSTM	Long Short-Term Memory
MII	Multichannel Intelligent Icons
ML	Machine Learning
MTS	Multivariate Time Series
MTSC	Multivariate Time Series Classification
NLP	Natural Language Processing
PAA	Piecewise Aggregate Approximation
QLoRA	Quantized Low-Rank Adaptation
ResNet	Residual Network
RF	Random Forest
SAA	Slopewise Aggregate Approximation
SAX	Symbolic Aggregate approXimation

SFA	Symbolic Fourier Approximation
SOTA	State of the Art
SVM	Support Vector Machine
TGE	Transition Graph Embeddings
TPR	True-Positive Rate
TSC	Time Series Classification
UCR	University of California, Riverside
UEA	University of East Anglia
UTS	Univariate Time Series
UTSC	Univariate Time Series Classification
XAI	Explainable Artificial Intelligence

This page has been left blank intentionally.

Chapter 1

Introduction

1.1 Machine Learning: Definition, Motivation, and Scope

Artificial Intelligence (AI) has been a scientific aspiration since the mid-twentieth century, but it was the emergence of *Machine Learning* (ML) as its dominant paradigm that transformed this aspiration into a practical, engineering discipline. ML is the subfield of AI that studies algorithms and statistical models enabling computer systems to perform tasks without being explicitly programmed for each one; instead, they *learn* from data. The most widely cited operational definition remains that of Mitchell: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" [1]. This simple characterisation captures the paradigm shift at the heart of ML: rather than a human expert encoding every rule that maps inputs to outputs, the rules are *induced* automatically and iteratively from observations.

The motivation for this approach is straightforward. A vast class of practically important problems — recognising speech, translating text, detecting anomalies in sensor streams, recommending content, or diagnosing disease — involves mappings that are too complex, too high-dimensional, or too context-dependent for explicit rule-based solutions [2, 3]. In each of these settings, data carries information that is difficult or impossible to express as a closed-form specification but can be extracted statistically given sufficient examples. The defining characteristic of ML, therefore, is not merely the use of statistics but the systematic use of experience to *generalise*: a model trained on seen examples should produce useful predictions on unseen ones.

The economic and scientific importance of machine learning has grown dramatically over the past two decades. Three concurrent trends have been the main drivers [2,4]: (i) the exponential growth of digitally collected data, arising from social networks, sensor networks, scientific instruments, and the broader digitalisation of economic activity; (ii) dramatic increases in computational power, especially through graphics-processing units (GPUs) and, more recently, specialised AI accelerators; and (iii) the maturation of algorithmic ideas that, though often decades old in conception, only became practically exploitable at the scale afforded by (i) and (ii). The conjunction of these forces has elevated ML from an academic sub-discipline to a general-purpose technology with applications spanning healthcare, finance, autonomous systems, climate modelling, natural language communication, and scientific discovery [2, 3].

This introductory chapter places the work of this thesis within the broader context of machine learning research. Section 1.2 reviews the principal data modalities to

which ML is applied and the distinct challenges each poses. Section 1.3 provides a taxonomy of learning paradigms and traces the historical development of the field. Section 1.4 surveys the current state of the art at a high level. Section 1.5 identifies open challenges that motivate ongoing research. Section 1.6 defines the scope of this thesis.

1.2 Types of Data in Machine Learning

A central insight of modern ML is that the structure of the data should dictate the inductive biases of the model. Different data modalities present qualitatively different challenges for learning, and the history of the field can in part be read as a succession of representational innovations matched to each modality. The most prominent data types are surveyed below and summarised in Table 1.1.

Tabular (structured) data. The classical setting for statistical learning consists of instances described by a fixed set of heterogeneous features arranged in rows and columns. Tabular data is ubiquitous in business analytics, clinical records, and financial modelling. The principal challenges are the mixture of continuous and categorical features, missingness, and the absence of the strong spatial or temporal structure that facilitates deep learning. Tree-based ensemble methods — random forests [5] and gradient-boosted trees — consistently rank among the strongest performers on tabular benchmarks, often outperforming deep models that lack appropriate inductive biases for this setting.

Time-series data. A time series is an ordered sequence of observations indexed by time, arising wherever a quantity is measured repeatedly at successive instants. Time series data is particularly prevalent in scientific monitoring (EEG - ElectroEncephalo-Gram , ECG - ElectroCardioGram), industrial sensing, finance, and meteorology. The key distinguishing property is temporal ordering: the position of a value in the sequence carries information, giving rise to phenomena such as autocorrelation, seasonality, non-stationarity, and temporal warping that are absent in unordered tabular data. Time series may be *univariate* (UTS) — a single channel of observations — or *multivariate* (MTS) — several co-evolving channels recorded simultaneously. The multivariate case introduces the additional challenge of inter-channel dependencies, which may be as informative as individual channel dynamics. A rich body of specialised methodology has developed for time series classification, ranging from distance-based methods and dictionary approaches to modern deep architectures [6–8].

Text and natural language data. Natural language is discrete, sequential, compositional, and deeply contextual. A sentence's meaning depends not only on the words it contains but on their order, their syntactic relationships, pragmatic context, and the vast background knowledge a human reader brings. The vocabulary is open-ended, morphological variation is large, and the same concept may be expressed in innumerable ways. These properties make Natural Language Processing (NLP) one of the most challenging — and consequential — application domains of ML. Early approaches relied on bag-of-words representations and handcrafted features; distributed word representations [9] introduced dense, geometry-respecting embeddings that captured semantic similarity; and contextualised representations from pre-trained transformer encoders [10] have since become the standard substrate for nearly all NLP tasks.

Image data. Images are high-dimensional, spatially structured arrays of pixel intensities. The dominant inductive bias is that useful features are locally connected

and translation-invariant: the same edge detector or object part may be useful at any spatial location. This bias is directly encoded in convolutional neural networks (CNNs) [11], which have been the defining architecture for computer vision for over a decade [12]. Challenges include scale, viewpoint, and illumination variation, occlusion, and fine-grained recognition.

Graph data. Many real-world systems are most naturally represented as graphs: molecules, social networks, knowledge bases, and citation networks all consist of entities (nodes) connected by relations (edges). Unlike images or sequences, graphs have irregular, non-Euclidean topology and no canonical ordering of nodes or neighbours, which invalidates the standard convolution operation. Graph neural networks (GNNs) address this by performing message passing over the neighbourhood structure and have become the dominant paradigm for graph-structured learning [13].

Audio data. Audio signals are one-dimensional waveforms with both temporal and spectral structure. Spoken language, music, and environmental sounds are routinely processed by first computing time-frequency representations (spectrograms or mel-frequency cepstral coefficients) and then applying CNN or recurrent architectures. End-to-end learning directly from raw waveforms has become increasingly practical as models scale [3].

Multi-modal data. Increasingly, state-of-the-art systems combine multiple modalities. Vision-language models ingest images and text simultaneously; medical AI systems combine imaging, genomics, and clinical notes. Multi-modal learning introduces the additional challenge of aligning representations across heterogeneous input spaces [14].

1.3 A Categorical and Historical Overview of Machine Learning

1.3.1 Learning Paradigms

ML methods are conventionally grouped according to the nature of the supervisory signal available during training. Table 1.2 summarises the principal paradigms; their properties are discussed below.

Supervised learning. The most widely studied paradigm: the algorithm is provided with a training set of input–output pairs $\{(\mathbf{x}_i, y_i)\}$ and learns a function $f : \mathcal{X} \to \mathcal{Y}$ that generalises to new inputs. When the output space $\mathcal{Y}$ is a finite set of class labels the task is *classification*; when it is a continuous quantity the task is *regression* [1]. The vast majority of practical ML applications — spam detection, medical diagnosis, speech recognition, image labelling — fall into this category. Its central challenge is the availability of labelled data, which is often expensive to obtain.

Unsupervised learning. Only inputs are available; the goal is to discover latent structure in the data. Clustering algorithms partition the input space into coherent groups; dimensionality reduction methods such as principal component analysis find compact representations; generative models learn the probability density $p(\mathbf{x})$ from which the data were drawn. Unsupervised learning is valuable precisely because unlabelled data is abundant and cheap to collect.

Semi-supervised learning. A small amount of labelled data is combined with a large pool of unlabelled data. Under the assumption that the marginal distribution $p(\mathbf{x})$ carries information about $p(y|\mathbf{x})$ — e.g., because the class boundary avoids regions of high density — the unlabelled examples can substantially improve performance

Table 1.1: Overview of common data modalities encountered in machine learning, with their structural properties, principal challenges, and representative model families.

Modality	**Structure**	**Principal challenges**	**Representative models**
Tabular	Heterogeneous features, unordered	Mixed types, missingness, no spatial prior	Random forests [5], gradient boosting
Time series	Ordered, temporal (uni-/multivariate)	Warping, non-stationarity, inter-channel dependencies	ROCKET, InceptionTime, LSTMs [7]
Text / NLP	Discrete, sequential, compositional	Open vocabulary, long-range context, ambiguity	Transformers, BERT [10]
Image	Spatial, translation-invariant	High dimensionality, scale, viewpoint variation	CNNs [11], Vision Transformers [15]
Graph	Non-Euclidean, relational	Irregular topology, permutation invariance	Graph neural networks [13]
Audio	Temporal + spectral	Noise, time–frequency resolution trade-off	CNN/RNN on spectrograms, wave models
Multi-modal	Heterogeneous, cross-modal	Modality alignment, missing modalities	Contrastive pretraining, fusion models

relative to supervised learning with labels alone [3].

Reinforcement learning. An agent interacts sequentially with an environment, observing states and selecting actions; the environment returns scalar rewards. The agent's objective is to learn a policy that maximises cumulative discounted reward. Reinforcement learning is the natural framework for sequential decision-making, robotics, and game-playing. Its first modern instance was Samuel's self-improving checkers player, which used a form of temporal-difference learning that would later be formalised [16].

Self-supervised learning. A variant of unsupervised learning in which pseudo-labels are generated automatically from the raw data by defining prediction tasks that require no human annotation — for example, predicting a masked token from its context, or predicting the relative positions of image patches. Self-supervised pretraining has become the dominant strategy for learning general-purpose representations at scale, underpinning BERT [10] and the broader family of large language and vision–language models [17].

1.3.2 Historical Development

The development of machine learning spans roughly eight decades and can be organised into four overlapping eras, illustrated in Figure 1.1.

Table 1.2: Comparison of the principal machine learning paradigms along four axes.

Paradigm	**Supervision**	**Typical tasks**	**Representative methods**
Supervised	Labelled pairs	Classification, regression	SVMs, random forests, CNNs
Unsupervised	None	Clustering, dimensionality reduction, anomaly detection	k-means, PCA, autoencoders
Semi-supervised	Few labels + unlabelled	Classification with scarce labels	Self-training, label propagation
Reinforcement	Reward signal	Sequential decision making	Q-learning, PPO, AlphaGo
Self-supervised	Labels from data	Representation learning, pretraining	BERT masking, contrastive learning

Foundations: Neural and Statistical Beginnings (1940s–1980s)

The intellectual genealogy of ML begins with McCulloch and Pitts, who showed in 1943 that networks of idealised neurons — each computing a weighted threshold over its inputs — could represent any logical proposition, establishing the theoretical plausibility of computation in neural tissue [18]. Rosenblatt's Perceptron (1958) made the leap from logic to learning: a single-layer network of binary threshold units whose weights could be updated by a simple error-correction rule to correctly classify linearly separable patterns [19]. Samuel's checkers program (1959) independently demonstrated that a computer could improve its performance through play, coining the term "machine learning" in doing so [16].

Early optimism was tempered by Minsky and Papert's 1969 analysis showing that the Perceptron could not solve problems requiring non-linear boundaries, such as the exclusive-or function, contributing to a prolonged period of reduced funding and interest subsequently dubbed the first "AI winter." The decisive technical response came in 1986, when Rumelhart, Hinton, and Williams demonstrated that the back-propagation algorithm — efficient computation of gradients through multi-layer networks via the chain rule — could train hidden units to learn useful intermediate representations, enabling multi-layer perceptrons to represent and learn non-linear functions [20]. This work underpins virtually all subsequent neural network research.

Statistical Learning and Kernel Methods (1990s–Early 2000s)

The 1990s saw a shift toward methods grounded in statistical learning theory, providing principled generalisation bounds and tractable optimisation. The support-vector machine (SVM), introduced by Cortes and Vapnik, performs maximum-margin linear classification in a high-dimensional feature space implicitly defined by a kernel function, achieving strong generalisation even in high dimensions [21]. SVMs dominated many benchmarks for over a decade and remain competitive in low-data regimes. Ensemble methods emerged as a complementary paradigm: Breiman's random forests [5] construct many decision trees on bootstrap samples with randomised feature selection, producing robust classifiers whose predictions are aggregated by majority vote;

gradient-boosted trees fit successive trees to the residuals of the ensemble, and both families remain among the strongest methods for tabular data.

In parallel, the foundations of deep learning were being quietly laid. LeCun and colleagues published the seminal demonstration of a CNN applied end-to-end to document recognition, learning hierarchical edge and shape detectors directly from pixel inputs [11]. Hochreiter and Schmidhuber introduced the long short-term memory (LSTM) network, which used gating mechanisms to selectively write, read, and erase a cell state, resolving the vanishing-gradient problem that had prevented recurrent networks from learning long-range temporal dependencies [22].

The Deep Learning Revolution (2006–2016)

Despite these early results, deep networks remained difficult to train reliably with the methods available at the time. A decisive shift began in 2006 when Hinton, Osindero, and Teh introduced a fast greedy layer-wise pre-training procedure for deep belief networks based on restricted Boltzmann machines, demonstrating that initialising deep networks with unsupervised pre-training enabled effective fine-tuning with gradient descent [23]. This reignited interest in deep architectures across the community.

The breakthrough that made deep learning mainstream was the AlexNet result at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. Krizhevsky, Sutskever, and Hinton trained a deep CNN with five convolutional layers and three fully connected layers, using ReLU activations, dropout regularisation, and GPU parallelism; the network achieved a top-5 error rate of 15.3%, compared with 26.2% for the second-best entry, a margin so large that it persuaded the computer vision community to abandon hand-engineered features almost overnight [12]. Subsequent years saw rapid progress: residual networks (ResNets) [24] introduced shortcut connections that enabled training of networks over one hundred layers deep, and the decade was retrospectively synthesised in an influential review that identified three ingredients — big data, powerful computation, and generic learning algorithms — as sufficient to learn essentially any function of practical interest [4].

Transformers, Pre-training, and Foundation Models (2017–Present)

The introduction of the Transformer architecture in 2017 marked the beginning of a fourth era. Vaswani and colleagues proposed replacing the recurrent processing of sequences with a *self-attention* mechanism that computes, for each position in a sequence, a weighted sum of all other positions based on their relevance, enabling fully parallel computation and the modelling of arbitrary-range dependencies [25]. The Transformer rapidly displaced LSTMs in sequence-to-sequence tasks and proved far more amenable to scaling.

The pre-training–fine-tuning paradigm was established at scale by BERT, which pre-trained a large transformer encoder on two self-supervised tasks — masked language modelling and next-sentence prediction — and then fine-tuned on eleven NLP benchmarks, setting new state-of-the-art results on all of them [10]. The complementary GPT line of work pursued autoregressive pre-training on ever-larger corpora: GPT-3, with 175 billion parameters and pre-training on approximately 300 billion tokens, demonstrated that scale alone suffices for strong few-shot performance on a diverse range of tasks, without any gradient updates at test time [26].

These results gave rise to the concept of *foundation models* — large models trained on broad data at massive scale whose representations and capabilities can be adapted

to a wide spectrum of downstream applications [17]. Vision transformers extended the paradigm to images by treating image patches as tokens and achieved competitive performance against CNNs at sufficient scale [15]. Instruction tuning and reinforcement learning from human feedback (RLHF) subsequently improved the alignment and usability of large language models [27], while empirical scaling laws have begun to characterise how performance improves predictably as a function of model size, training data, and compute [28, 29].

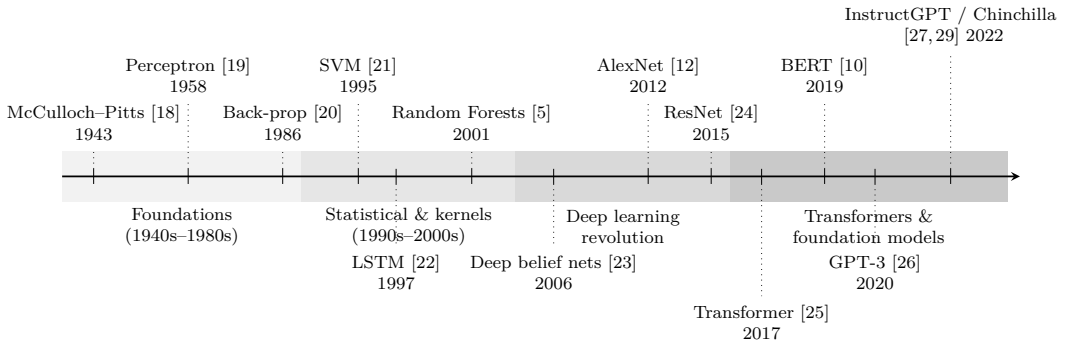


Figure 1.1: A condensed timeline of machine learning milestones spanning the four development eras described in Section 1.3.2. Shading intensity increases with era recency. Milestone positions are approximate.

1.4 Current trends in Machine Learning

The contemporary ML landscape can be characterised as a layered stack of model families, each dominant in a particular regime of data modality, dataset size, and computational budget.

Classical methods for structured data. For tabular and small-to-medium datasets, kernel machines [21] and tree ensembles [5] remain strong, frequently hard-to-beat baselines. Their advantages include explicit handling of heterogeneous feature types, robustness to irrelevant features, and interpretability relative to deep networks. Gradient-boosted trees (XGBoost, LightGBM, CatBoost) consistently win structured-data competitions and remain the method of choice across much of applied ML.

Convolutional architectures for spatial data. Deep CNNs, from the original LeNet [11] through AlexNet [12], VGGNet, Inception, and ResNet [24], defined the standard for image classification across the 2010s. Transfer learning from models pre-trained on large image corpora (ImageNet) enables strong performance on tasks with limited labelled data — a pattern that prefigures the large-scale pre-training paradigm later popularised in NLP. CNNs have also been successfully adapted to one-dimensional sequential data, including audio and time series [7].

Sequence models. The LSTM [22] and gated recurrent unit addressed the vanishing-gradient problem in recurrent networks and dominated sequence modelling tasks throughout the 2010s. They have largely been supplanted by the Transformer [25] for tasks where parallel training is feasible, but retain relevance in settings where sequential processing is constrained by latency or memory.

The Transformer and self-attention. The Transformer's self-attention mechanism computes pairwise interactions between all positions in a sequence, enabling the learning of arbitrary-range dependencies with full parallelism. Its adoption spread

from NLP to computer vision [15], audio, protein structure prediction (AlphaFold), and time series forecasting, supporting a view of the Transformer as a general-purpose sequence model. The scaling properties of the architecture are particularly favourable: performance improves predictably and sub-linearly in loss as a power law of model parameters, training tokens, and compute [28], which has motivated the construction of progressively larger models.

Pre-trained language models and LLMs. BERT [10] established that pre-training a large transformer encoder via self-supervised objectives yields a general-purpose text representation that can be fine-tuned to state-of-the-art performance across diverse NLP tasks with relatively little labelled data. GPT-3 [26] extended this to autoregressive generation and demonstrated emergent few-shot learning at 175 billion parameters. Instruction tuning [27] further improved alignment with human intent through reinforcement learning from human feedback (RLHF). Recent open-weight large language models such as LLaMA [30] have made these capabilities accessible for research and fine-tuning. The Chinchilla scaling analysis [29] refined the understanding of optimal resource allocation, showing that many existing models were over-parameterised relative to their training data and advocating for a balanced scaling of parameters and tokens.

Foundation models and multi-modal systems. Bommasani et al. introduced the term *foundation model* to describe large pre-trained models that serve as a generalist substrate for a wide array of downstream tasks [17]. This paradigm has extended beyond language to vision–language systems (CLIP, Flamingo), audio, video, and scientific applications, blurring the historically clean separation between modality-specific research communities. The broad, transferable representations learned during pre-training reduce the need for domain-specific labelled data and represent one of the most significant structural shifts in ML methodology in recent years.

Generative models. Generative adversarial networks (GANs) [31] introduced a game-theoretic framework for training implicit generative models, producing high-fidelity image synthesis. Diffusion models subsequently surpassed GANs on image quality and diversity and have been extended to audio, video, and scientific data generation.

Detailed treatment of domain-specific state-of-the-art results for time series classification and natural language processing — including benchmark datasets, evaluation protocols, and per-method performance — is deferred to the respective chapters of this thesis, where the relevant prior art is reviewed in full.

1.5 Open Challenges in Machine Learning

Despite substantial progress, a number of fundamental challenges constrain the reliable and equitable deployment of machine learning systems and motivate sustained research investment.

Data scarcity and label efficiency. State-of-the-art models are notoriously data-hungry. Obtaining labelled examples is expensive, time-consuming, and in some domains — clinical medicine, rare event detection, low-resource languages — simply not possible at the scale modern deep learning requires. Few-shot and zero-shot learning [32], active learning, data augmentation, and transfer learning [33] are the primary responses, but closing the gap between deep-learning data requirements and realistic annotation budgets remains an open problem.

Interpretability and explainability. The most accurate models — large neural

networks and high-cardinality ensembles — are opaque: their decisions cannot readily be inspected or explained in terms a domain expert would find meaningful. This is not merely an aesthetic concern; in high-stakes domains such as healthcare, criminal justice, and autonomous systems, the inability to explain a prediction can be a legal and ethical barrier to deployment. The field of Explainable AI (XAI) seeks methods and taxonomies for making model decisions interpretable while preserving accuracy [34]. Techniques range from post-hoc local approximations (LIME, SHAP) to inherently interpretable architectures (linear models, decision trees, symbolic representations). Balancing accuracy and interpretability is a recurring theme in the design choices made throughout this thesis.

Computational cost and environmental impact. Training large foundation models requires millions of GPU-hours, with correspondingly large financial and carbon footprints [3, 28]. While scaling laws [29] characterise how performance improves with compute, they also imply that further progress at the frontier will become ever more expensive. The gap between the resources available to large industrial laboratories and to academic or public-sector researchers has widened considerably, raising questions about reproducibility and equitable participation in frontier research.

Generalisation and distribution shift. A model that performs well on its training distribution may degrade severely when deployed in a different context: a clinical model trained on one hospital's data may fail on another's; a sentiment classifier trained on product reviews may not transfer to social media. Understanding generalisation in over-parameterised models remains theoretically incomplete [4], and practical solutions — domain adaptation, robust optimisation, and causal inference — remain active research areas.

Bias and fairness. Models trained on historical data inherit the biases present in that data, potentially producing discriminatory outcomes across demographic groups. A substantial literature catalogues the sources of bias (measurement, representation, aggregation) and the competing formal definitions of fairness (demographic parity, equalised odds, calibration) [35]. These definitions are often mutually incompatible, and reconciling them with predictive accuracy is an active area of work in the ML theory and policy communities.

Privacy. Models can inadvertently memorise and leak sensitive information from their training data, posing risks of re-identification and data disclosure [36]. Differential privacy provides a rigorous framework for bounding information leakage but often comes with accuracy costs, especially in low-data regimes.

Robustness and adversarial vulnerability. Deep neural networks are susceptible to adversarial examples — inputs imperceptibly modified from natural examples in ways that cause confident misclassification [37]. This brittleness poses risks in safety-critical applications and suggests that current models have learned superficial statistical correlations rather than the robust, abstract representations a human would form. Certified robustness, adversarial training, and randomised smoothing are among the defences under active investigation.

1.6 Scope of This Thesis

Situated within the landscape described above, this thesis advances machine learning methodology for time series data — one of the most structured, sequential, and practically consequential data modalities identified in Section 1.2 — addressing several of the challenges identified in Section 1.5.

Time series classification (TSC) constitutes one of the most studied tasks in the ML literature, with applications spanning healthcare monitoring, industrial fault detection, human activity recognition, and financial analysis. Despite the richness of existing methodology, open problems remain in the design of efficient, interpretable, and generalisable classifiers — particularly in the multivariate setting, where inter-channel dependencies and high dimensionality compound the challenges of temporal modelling. This thesis contributes novel methods and extensive empirical evaluations for both Univariate Time Series Classification (UTSC) and Multivariate Time Series Classification (MTSC), building upon and critically comparing against the state of the art as surveyed in dedicated reviews [6–8].

Within this domain, the thesis commits to a specific representational choice: symbolic time series representations, and in particular the Symbolic Aggregate approXimation (SAX) and the feature models built on top of it. This choice is deliberate. Symbolic methods offer a combination of computational efficiency, robustness to noise, and — most importantly — interpretability that few competing families of methods provide simultaneously, and these properties grow in importance as XAI becomes a practical requirement rather than an academic aspiration. The contributions of this thesis systematically extend what symbolic representations can achieve: enriching them with cross-channel and trend information, optimizing their parameters in a principled manner, and pairing them with more expressive downstream models — including pre-trained large language models (LLMs) [25], whose native token-sequence interface the symbolic representation is shown to match structurally, and GNNs, which recover the sequential structure that frequency-based symbolic features discard. In this thesis, transformer-based architectures thus enter not as a separate research thread on natural language data, but as a modelling tool applied to symbolic time series, investigated under realistic data and compute constraints through parameter-efficient fine-tuning.

Across all contributions, the recurring themes are *computational efficiency*, *interpretability*, and *generalisation*. The specific datasets, model architectures, experimental protocols, and results are presented in full in the chapters that follow, where the relevant domain-specific literature is also reviewed in greater depth.

1.7 Thesis Outline

The remainder of this thesis is organized as follows.

Chapter 2 establishes the technical foundation on which all subsequent chapters build. It reviews the landscape of time series representation methods, presents the SAX transformation and the Bag-of-Patterns (BOP) model in full detail, and surveys the taxonomy of univariate and multivariate TSC methods, characterizing each algorithmic family by its typical trade-off between accuracy, computational cost, and interpretability.

Chapter 3 presents a systematic review of the SAX literature, conducted against the Scopus database, and organizes the identified extensions into an original taxonomy structured around the specific weakness of the base method that each variant addresses. The recurring gaps this taxonomy surfaces — trend loss, limited multivariate support, heuristic parameter selection, and the sequential order discarded by frequency-based aggregation — form the motivating thread for the original contributions that follow.

Chapter 4 introduces the first original contributions: two independent extensions of SAX, each targeting one weakness identified in the taxonomy. The Multichannel

Intelligent Icon (MII) capture cross-channel symbolic co-occurrence in multivariate time series, while the Slopewise Aggregate Approximation (SAA) recovers trend information through a parallel, angle-based discretization branch. Both are evaluated jointly on a human activity recognition benchmark against the BOP baseline and a classical feature-engineering approach.

Chapter 5 addresses the parameter selection problem by applying the DIRECT global optimization algorithm to the joint tuning of SAX parameters across four feature-extraction strategies for multivariate TSC. The approach is evaluated on twenty-four benchmark datasets from the UEA archive against six state-of-the-art baselines, with additional analyses of runtime, statistical significance, and interpretability.

Chapter 6 establishes a structural bridge between symbolic time series representations and large language models. It develops SAX_HAR-LLM, a dual-stream framework that pairs a trend-aware, case-modulated symbolic encoding with kinematics-informed descriptors, fusing both into structured prompts on which a causal LLM is fine-tuned for activity recognition. The framework is evaluated on two HAR benchmarks and complemented by an attention-based interpretability analysis.

Chapter 7 addresses BOP's loss of sequential order by constructing directed graphs over SAX-word transitions and learning graph embeddings through GNNs. Two graph-construction schemes are proposed and evaluated, alone and in combination with BOP, on twelve benchmark datasets from the UCR archive.

Chapter 8 concludes the thesis, summarizing its contributions, tracing the threads that connect them, acknowledging their limitations, and outlining the directions in which this line of research continues.

This page has been left blank intentionally.

Chapter 2

Time Series Representation and Classification

Understanding the methodological landscape underlying this work requires examining three interconnected topics: how time series data are represented, how those representations are exploited for classification, and how classification methods scale to the multivariate setting. These three concerns are not independent—the choice of representation fundamentally constrains what patterns a classifier can detect, and the transition from univariate to multivariate data amplifies both the representational and computational demands on the model. This section reviews each topic in turn, beginning with an overview of time-series representation techniques, followed by a survey of UTSC methods, and concluding with the specific challenges and approaches associated with MTSC. Throughout, particular attention is given to the trade-off between predictive accuracy, computational efficiency, and interpretability—a tension that motivates the design choices of the proposed method. The growing importance of XAI [38] makes this tension increasingly consequential: in domains such as healthcare, industrial monitoring, and finance, a model that cannot account for its decisions is of limited practical value regardless of its benchmark performance [39, 40].

2.1 Time Series Representation

The proliferation of IoT devices has made the mass generation of time series data a defining characteristic of the modern technological landscape, drawing considerable attention from the research community. Time series data consists of observations collected sequentially over time, setting it apart from other data types such as cross-sectional or spatial data. The analysis of time series is central to understanding and modeling complex processes across a broad range of fields, including Finance and Economics [41], Health and Medicine [42], Environmental Monitoring [43], Engineering and Manufacturing [44], and Transportation and Logistics [45]. The sheer volume of such data presents a significant challenge to the data science community, necessitating the development of new methodologies for effectively modeling, representing, retrieving, processing, interpreting, and visualizing it. As interconnected devices and sensors continue to multiply, the volume, velocity, and variety of generated time series data grow at an exponential rate. This data abundance simultaneously creates enormous challenges and opportunities for data management, storage, processing, and analytics, calling for scalable infrastructure and innovative methods capable of

This chapter is based on the following publications: [J1], [J2], [C1].

extracting meaningful insights and enabling real-time decision-making within data-generating ecosystems. The pervasive nature of time series data demands robust analytical tools able to capture and examine temporal interdependencies. The rich patterns and dynamics embedded within such data present distinctive challenges and opportunities for researchers seeking to uncover hidden correlations, generate accurate predictions, and develop deeper understanding of the underlying processes.

The core tasks in time-series data processing include content-based querying, clustering, classification, prediction, anomaly detection, motif detection, and rule discovery [46–48]. These problems have been extensively studied within the fields of pattern recognition and data mining over many years. However, the direct application of conventional pattern recognition and data mining techniques to time series data encounters several inherent difficulties, stemming from three main sources [49]. First, time series data tends to be highly dimensional, with the number of data points potentially reaching tens of thousands. Second, corresponding elements across two time series may not align cleanly due to differences in length, scale, translation, shift, or irregular spacing between adjacent observations. Third, the concept of similarity in time series analysis differs fundamentally from its conventional usage in pattern recognition. Rather than relying on all elements of a pattern vector to determine similarity, time series analysis often draws on only subsets of elements to establish correspondence. Two distinct time series may be regarded as similar if they share sufficiently long common subsequences or exhibit analogous patterns occurring in the same temporal order.

To address these challenges—and the high dimensionality of time series data in particular—a variety of representation techniques and analytical approaches have been investigated. Time series data can be represented in numerous ways, each offering a distinct perspective on the underlying temporal patterns and dynamics. By organizing data in structured formats, recurring patterns, trends, and relationships can be identified and leveraged across applications including forecasting, anomaly detection, and decision support. Temporal dependencies and patterns become more tractable through appropriate representations, facilitating the detection of seasonality, trends, and anomalies. Representation also supports data visualization, enabling users to comprehend and draw insights from observed fluctuations over time.

The existing approaches to time series representation are commonly grouped into four categories, illustrated in Figure 2.1 [50–52]. *Model-based* representations fit parametric models—such as autoregressive or hidden Markov models—to the observed data, capturing global generative structure but at increased computational cost. *Data-dictated* representations derive their structure directly from the statistical properties of the specific series being encoded, such as clipping each value relative to the series mean. *Data-adaptive* representations adjust their structure, such as segment or breakpoint placement, in response to the characteristics of each individual series; symbolic methods fall within this class. Finally, *non-data-adaptive* representations apply a fixed transformation regardless of the input—such as Fourier and wavelet transforms, or the equal-width segmentation used in Piecewise Aggregate Approximation—trading adaptability for computational simplicity and reproducibility.

Among data-adaptive representations, symbolic methods occupy a particularly prominent position in the literature. These approaches discretize continuous signals into sequences of symbols drawn from a finite alphabet, yielding compact representations that support efficient storage, indexing, and analysis while preserving essential temporal patterns. A well-designed symbolic representation should satisfy two funda-

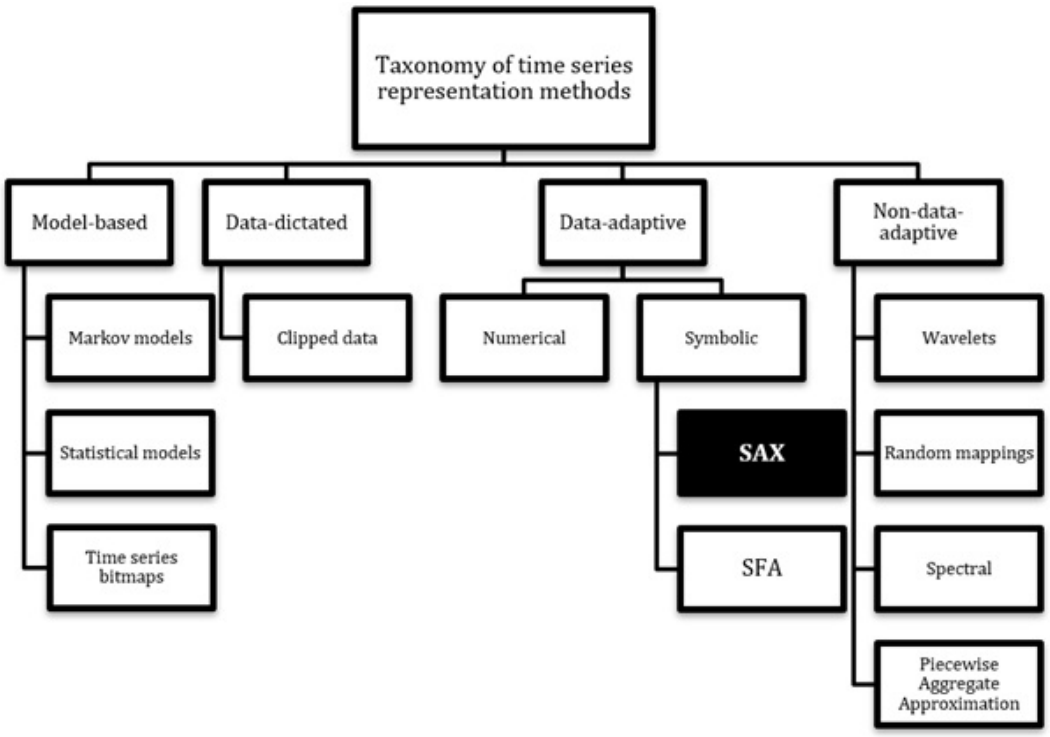


Figure 2.1: The main categories of time-series representation methods.

mental properties [53]: *perfect mapping*, whereby similar data points are consistently assigned the same symbol; and *tight bounding*, whereby the distance between symbolic representations provides a close approximation to the Euclidean distance in the original space.

The most widely adopted symbolic representation is SAX [54], which first reduces dimensionality via PAA and then discretizes each segment into a symbol using breakpoints derived from a standard Gaussian distribution. SAX satisfies both design requirements above and has been successfully applied across a broad range of time series analysis tasks, including similarity search, clustering, anomaly detection, and classification [55]. Its interpretability, computational efficiency, and well-understood theoretical properties make it a natural foundation for transparent and scalable analytical pipelines. The choice of representation is therefore not merely a preprocessing step; it fundamentally shapes what patterns are accessible to a classifier and how interpretable the resulting model will be. This motivates the present work's focus on symbolic representations as the basis for lightweight and explainable multivariate time series classification.

2.2 The Symbolic Aggregate ApproXimation Method

SAX was introduced by Lin et al. [54,56] and has since become a foundational technique in time-series data mining. SAX transforms a continuous, real-valued time series into a compact sequence of discrete symbols, enabling efficient storage, indexing, and analysis while preserving the essential temporal structure of the original signal. Its simplicity, computational efficiency, and well-understood theoretical properties have made it a reference point for symbolic time series representation, and it has been successfully

applied to a broad range of tasks, including clustering [57], anomaly detection [58], motif discovery [59], visualization [60], and time series classification [55, 61].

A key theoretical property of SAX is its support for *lower bounding*: the distance between two SAX representations provides a guaranteed lower bound on the Euclidean distance between the original time series [54]. This property allows large portions of the search space to be pruned during similarity queries without computing exact distances, significantly reducing computational cost while preserving retrieval correctness.

The SAX transformation proceeds in two sequential steps: dimensionality reduction via PAA, followed by discretization of the resulting coefficients into symbols drawn from a finite alphabet. Both steps are described in detail below.

2.2.1 Piecewise Aggregate Approximation

The first step of the SAX pipeline is dimensionality reduction via PAA [62]. Given a time series $X = \{x_1, x_2, \ldots, x_n\}$ of length n, PAA partitions it into m equal-length, non-overlapping segments and replaces each segment with its mean value, producing a reduced representation $\bar{X} = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m\}$, where $m \ll n$. Formally, each PAA coefficient is computed as:

$$\bar{x}_i = \frac{m}{n} \sum_{j=\frac{n}{m}(i-1)+1}^{\frac{n}{m}i} x_j \tag{2.1}$$

where n is the original series length and m is the user-defined number of segments. Each coefficient $\bar{x}_i$ therefore summarizes the average behavior of the signal over the i-th interval of length n/m.

PAA achieves substantial compression while retaining the coarse temporal shape of the signal. By capturing the mean level of each segment rather than individual sample values, it discards fine-grained noise while preserving the dominant trends and level shifts that are most discriminative for downstream tasks. Importantly, the PAA representation maintains a tight lower bound on the Euclidean distance between the original series, making it well-suited for efficient similarity search [62].

Note: Before applying PAA, the time series is z-normalized (zero mean, unit variance) over the window of interest. This normalization step is essential to ensure that the resulting PAA coefficients are compatible with the standard Gaussian breakpoints used in the subsequent discretization step, and to enable meaningful comparison across windows, subjects, and channels.

2.2.2 Discretization via Gaussian Breakpoints

Following PAA, each coefficient $\bar{x}_i$ is mapped to a discrete symbol drawn from a predefined alphabet Σ of size a. We first introduce the key definitions:

Definition 1 *Alphabet* (Σ, a): The ordered set of symbols used for discretization, typically the first a lowercase letters {a, b, ...}. The alphabet size a is a user-defined parameter.

Definition 2 *Breakpoints* $B = \{\beta_1, \beta_2, \ldots, \beta_{a-1}\}$: A set of $a - 1$ threshold values that partition the real line into a equiprobable regions under the standard Gaussian distribution $\mathcal{N}(0, 1)$. Because the input series is z-normalized, the PAA coefficients approximately follow this distribution, ensuring that each symbol is assigned with roughly equal probability.

Definition 3 *SAX Representation*: The sequence of symbols $S = \{s_1, s_2, \ldots, s_m\}$ obtained after discretizing all m PAA coefficients, constituting the symbolic representation of the time series window. Short subsequences of this representation, referred to as *SAX words*, are later extracted via a sliding window in the Bag-of-Patterns model (Section 2.2.3).

The discretization rule assigns a PAA coefficient $\bar{x}_i$ to symbol α_j if and only if $\beta_{j-1} \leq \bar{x}_i < \beta_j$, where $\beta_0 = -\infty$ and $\beta_a = +\infty$. The breakpoint values are fixed and obtained from a statistical lookup table, an excerpt of which is given in Table 2.1 for alphabet sizes $a = 3$ through 10.

Table 2.1: Gaussian breakpoints β_i defining equiprobable bins for alphabet sizes $a = 3$ to 10. Values are obtained from the standard normal distribution lookup table.

$a \rightarrow$	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
β_1	−0.43	−0.67	−0.84	−0.97	−1.07	−1.15	−1.22	−1.28
β_2	0.43	0	−0.25	−0.43	−0.57	−0.67	−0.76	−0.84
β_3	—	0.67	0.25	0	−0.18	−0.32	−0.43	−0.52
β_4	—	—	0.84	0.43	0.18	0	−0.14	−0.25
β_5	—	—	—	0.97	0.57	0.32	0.14	0
β_6	—	—	—	—	1.07	0.67	0.43	0.25
β_7	—	—	—	—	—	1.15	0.76	0.52
β_8	—	—	—	—	—	—	1.22	0.84
β_9	—	—	—	—	—	—	—	1.28

Figure 2.2 provides a complete illustration of the SAX pipeline. A z-normalized time series of n samples is divided into m equal-length segments; the mean of each segment yields the PAA coefficient; and each coefficient is then assigned a symbol according to its position relative to the Gaussian breakpoints. The result is a compact symbolic word that captures the coarse temporal shape of the original signal.

The use of equiprobable breakpoints has an important practical consequence: each symbol is assigned with approximately equal frequency across a normalized dataset, maximizing the information content of the encoding and avoiding symbol bias. Furthermore, because SAX satisfies both the *perfect mapping* and *tight bounding* properties [53], symbolic distances computed between SAX words provide meaningful approximations of the true Euclidean distances in the original space—an essential requirement for reliable classification and retrieval.

2.2.3 Bag-of-Patterns Representation

While a single SAX word encodes the shape of one time series window, practical classification requires a fixed-length, order-agnostic representation that summarizes the distributional properties of patterns across the entire time series. This is achieved through the BOP model [63].

BOP operates by sliding a fixed-length window of size w across the SAX representation S with a step of one position, extracting the symbolic substring at each position as a SAX word. The resulting collection of words is then summarized as a frequency histogram over the vocabulary of all possible a^w distinct words. The

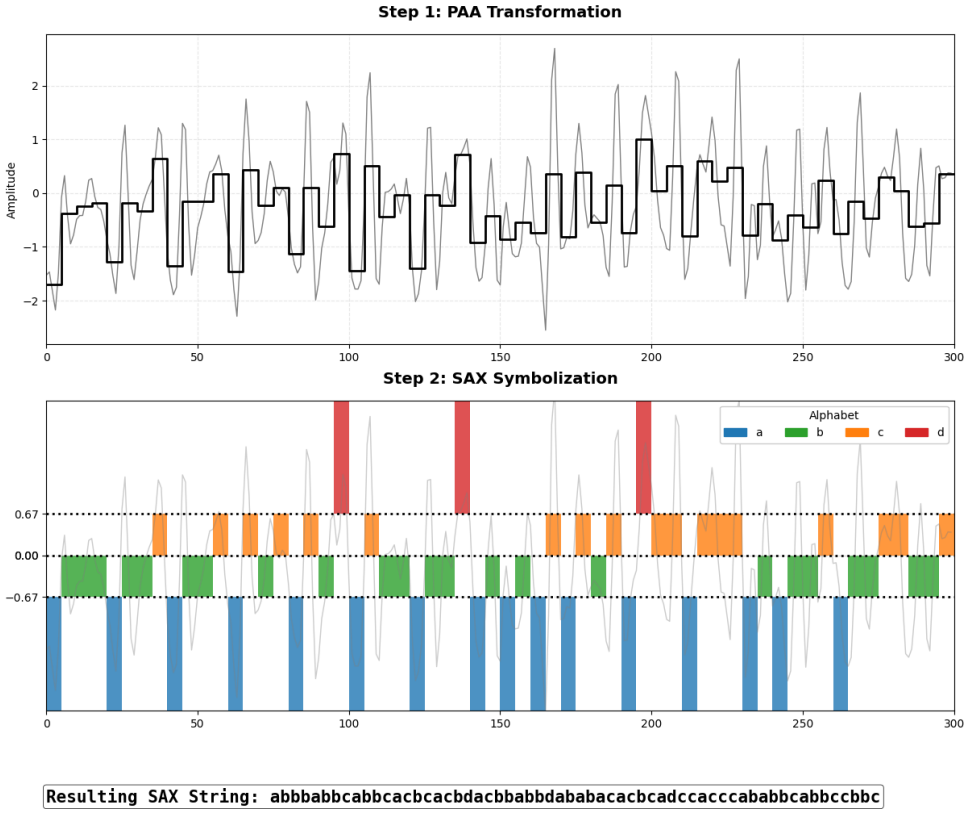


Figure 2.2: Overview of the SAX transformation pipeline. A z-normalized time series (gray curve) is divided into m equal-length segments; the mean of each segment yields the PAA representation (black step function). Each PAA coefficient is then mapped to a discrete symbol from a finite alphabet using Gaussian breakpoints, producing a compact symbolic word.

resulting collection of words is then summarized as a frequency histogram over the vocabulary of all possible a^w distinct words. Formally, given a symbolic sequence $S = \{s_1, s_2, \ldots, s_m\}$ of length m, BOP constructs a feature vector $\mathbf{h} \in \mathbb{Z}^{a^w}$ where each entry $\mathbf{h}_k$ counts the number of times the k-th word appears in the sliding window traversal:

$$h_k = \sum_{j=1}^{m-w+1} \mathbf{1}\left[W_j = \omega_k\right] \tag{2.2}$$

where $W_j = \{s_j, s_{j+1}, \ldots, s_{j+w-1}\}$ is the word extracted at position j, and ω_k is the k-th element of the word vocabulary.

The BOP histogram captures the relative frequency of local symbolic patterns throughout the time series, encoding structural regularities while deliberately discarding their absolute temporal positions. This position-invariance is a deliberate design choice: it makes BOP robust to phase shifts and local distortions that would otherwise cause identical patterns occurring at slightly different time points to appear unrelated. The resulting histogram is a sparse, high-dimensional vector that can be directly fed into standard vector-based classifiers.

Figure 2.3 illustrates the BOP construction process. Following the SAX sequence produced in the previous step, a sliding window of length $w = 2$ is applied with a step

of one position. Each extracted bigram is recorded, and the final histogram aggregates their counts across the full sequence.

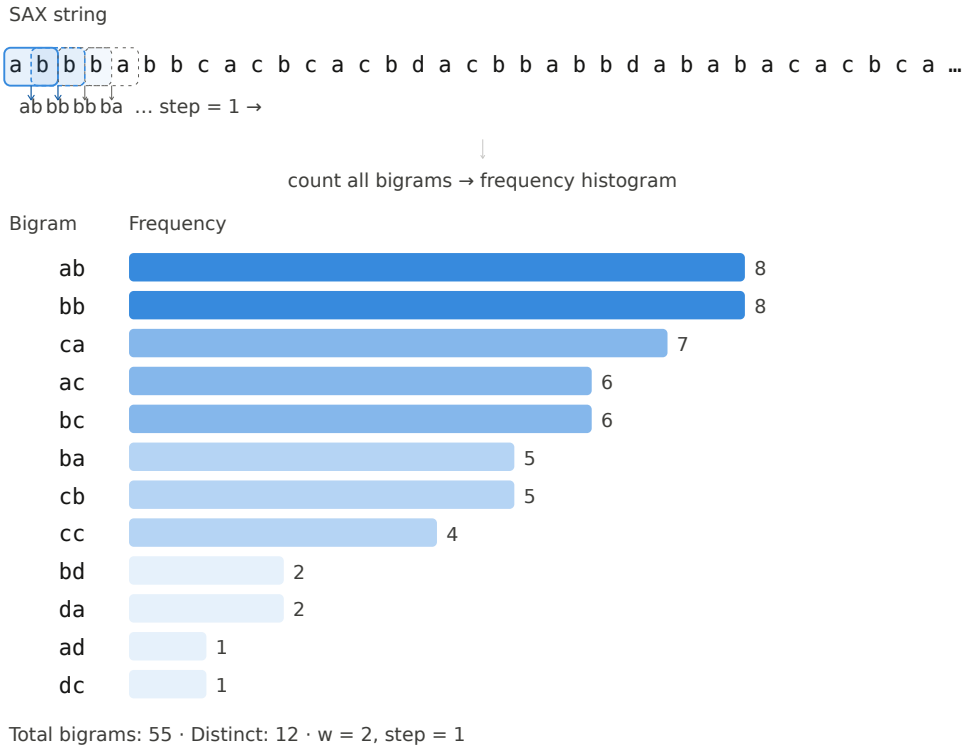


Figure 2.3: Construction of the Bag-of-Patterns representation. A sliding window of length $w = 2$ and step 1 is applied to the SAX string, extracting all overlapping bigrams. The first four windows are shown explicitly; the process continues identically across the full sequence. Bigram counts are aggregated into a frequency histogram (55 total bigrams, 12 distinct).

A practical consideration in BOP is the handling of numerosity reduction: consecutive identical words produced by the sliding window are often collapsed into a single count to avoid over-representing stationary segments [63]. This step prevents flat or slowly varying portions of the signal from dominating the histogram and inflating the counts of uninformative patterns. Together, PAA, SAX discretization, and BOP aggregation form a coherent symbolic pipeline that is interpretable at every stage: the segments are human-readable, the symbols are semantically grounded in the signal's amplitude, and the histogram reflects the actual distribution of temporal patterns.

2.2.4 Merits of the SAX Method

Introduced in 2003, SAX has maintained undiminished relevance in the time-series data mining community for over two decades [54, 56], a longevity that is itself a proof of its practical value. Its broad adoption across tasks as diverse as classification, clustering, anomaly detection, motif discovery, indexing, retrieval, forecasting, and change detection [57–60, 64] reflects a combination of computational, representational,

and practical merits that few competing methods simultaneously offer.

From a computational standpoint, SAX achieves substantial dimensionality reduction by converting a continuous time series of arbitrary length into a compact symbolic sequence, enabling significantly faster processing and reduced storage requirements relative to raw signal analysis. These properties scale favorably to large datasets, making SAX well suited for applications where computational resources or storage capacity are constrained. The PAA step additionally provides a degree of noise smoothing: by summarizing each segment through its mean value, fine-grained fluctuations and measurement noise are suppressed, allowing the dominant temporal patterns to emerge more clearly in the symbolic representation.

From a representational standpoint, the use of equiprobable Gaussian breakpoints ensures that symbols are assigned with approximately equal frequency across normalized data, maximizing the information content of the encoding. The lower bounding property — discussed formally in Section 2.2.2 — guarantees that symbolic distances provide a conservative approximation of true Euclidean distances, enabling efficient similarity search and nearest-neighbour retrieval without sacrificing correctness. Furthermore, the symbolic representation confers a degree of robustness to minor temporal variations and phase shifts: small perturbations in the signal that do not alter the segment mean will leave the SAX word unchanged, making the method tolerant to the kind of local distortions commonly encountered in real-world time series.

Finally, SAX offers considerable practical appeal. Its symbolic output is directly interpretable — each character in the SAX word corresponds to the amplitude level of a specific temporal segment — making representations accessible to domain experts without requiring familiarity with the underlying mathematics. The method is domain-agnostic and has been successfully deployed across fields including healthcare, finance, industrial monitoring, and environmental sensing, with minimal task-specific adaptation. Its implementation is straightforward, relying on standard statistical tables and elementary arithmetic, and its hyperparameters are few and well-understood. Together, these properties have established SAX as a reference point for symbolic time series representation and a natural first choice for a broad spectrum of data mining tasks [54].

2.3 Time Series Classification

TSC is the supervised learning task of assigning a complete temporal sequence to one of several predefined categories, leveraging the ordering of observations to identify characteristic shapes, trends, and recurring patterns. The objective is to infer the categorical label of an entire sequence, with applications spanning medical diagnosis, human activity recognition, financial analysis, and manufacturing. TSC has established itself as a fundamental task across numerous high-impact domains, including healthcare monitoring, financial forecasting, speech recognition, and industrial diagnostics [65–69]. In many such settings—particularly those involving complex systems or sensor-dense environments—data are inherently multivariate [70]. MTSC classification is considerably more demanding than its univariate counterpart, as it requires jointly modeling temporal dependencies within individual channels and cross-channel interdependencies [71]. The proliferation of multivariate data in domains such as Industry 4.0 further amplifies the demand for MTS classifiers that are simultaneously scalable, accurate, and interpretable [72].

Notwithstanding the substantial progress made in TSC, much of this advancement

has been concentrated in deep and hybrid architectures that favor predictive accuracy over interpretability and efficiency, while also being contingent on large volumes of training data. Such architectures rely on extensive labeled samples to effectively optimize their numerous parameters, rendering them data-intensive and ill-suited for settings where annotations are limited or costly. Recent models such as MF-Net [73], DKN [74], and Dyformer [75] exemplify this trajectory, combining components such as CNNs, transformers, and dynamic feature decomposition modules to push state-of-the-art performance. Although highly capable, these models impose considerable computational demands and raise significant concerns regarding transparency and practical deployment. Approaches like MTS2Graph [76] have sought to address interpretability through graph-based transformations, yet they similarly introduce architectural complexity and depend on elaborate post-hoc analysis.

Alongside deep learning, a parallel research direction has prioritized interpretability and computational lightness. Symbolic and dictionary-based methods have attracted renewed interest within this context. The WEASEL 2.0 framework [77], for instance, incorporates randomized dilated dictionary features to enrich symbolic representations while preserving scalability. Likewise, SCALE-BOSS [78] and its multiresolution variant SCALE-BOSS-MR [79] illustrate that carefully designed symbolic representations can remain competitive with neural approaches, particularly when transparency and low computational overhead are essential. BORF [80] further supports this perspective by embedding convolutional operations within symbolic encodings, producing fast and interpretable alternatives to opaque classifiers.

The viability of simpler methods has also been affirmed empirically. Dhariyal et al. [81] demonstrate that carefully tuned tabular models can match the accuracy of sophisticated architectures, underscoring the value of rigorous hyperparameter optimization. This finding fits within a broader methodological orientation toward transparent and principled model design. The Z-Time model [82] similarly shows that accuracy, interpretability, and computational efficiency can be balanced in MTS classification—a combination of properties increasingly sought in domains with strict auditability requirements.

A persistent obstacle to the wider adoption of MTS classification (MTSC) methods has been the limited availability of standardized benchmarking resources. The UCR archive has long served as the primary reference for UTSC evaluation [83], but analogous progress in multivariate settings was constrained until the 2018 introduction of the UEA archive, which provides 30 MTS benchmark datasets [71, 84]. Subsequent systematic reviews [40] and benchmarking studies [85] have helped consolidate this landscape, offering a clearer taxonomy of methods and confirming that symbolic approaches such as BOSS [86] and WEASEL [87] remain robust across diverse evaluation conditions.

More recent work has increasingly called for evaluation criteria extending beyond classification accuracy. Mahmud et al. [40] highlight the necessity of accounting for interpretability and resource efficiency, especially in industrial settings where decisions must be both timely and explainable. In a related vein, Baldán and Benítez [39] incorporate descriptive complexity features to improve interpretability in MTSC while maintaining competitive accuracy. These developments reflect a broader shift toward classification frameworks that are not only accurate but also transparent and computationally mindful.

2.4 Taxonomy of Time Series Classification Methods

TSC is a foundational machine learning task with applications spanning a wide range of domains. TSC methods can generally be characterized from two perspectives: the feature extraction techniques employed and the classification strategies applied. This work focuses on the former, as it determines how raw temporal signals are transformed into representations suitable for downstream classification. A prevailing tendency in the literature is the construction of increasingly elaborate and diverse feature spaces, often through the stacking of multiple signal transformations and the incorporation of randomization to attain state-of-the-art accuracy [80]. This pursuit of predictive power, however, frequently compromises interpretability—a trade-off that is drawing growing scrutiny in light of the expanding role of XAI [38]. XAI seeks to develop systems that not only perform well but also produce transparent and human-understandable decision processes, a requirement of particular importance in high-stakes domains such as healthcare, finance, and manufacturing [39]. Among interpretable TSC approaches, shapelet-based methods are among the most prominent, owing to their semantic grounding in time-domain substructures [80].

To contextualize the broader methodological landscape, Table 2.2 summarizes the typical trade-offs associated with the main algorithmic categories in TSC. While individual methods within each category naturally vary, this characterization reflects commonly observed patterns across accuracy, computational cost, and interpretability—three dimensions of central importance in real-world deployments where explainability, resource constraints, or auditability are non-negotiable. This overview motivates the deeper discussion that follows concerning where current approaches succeed and where they fall short. Consistent with recent systematic reviews, and tailored to the objectives of this work, we adopt a categorization of TSC algorithms that facilitates clearer comparison of model assumptions and representational choices.

Distance-based methods assess pairwise sequence similarity without constructing explicit features, which renders their decision process opaque and computationally costly at scale. Feature-based approaches summarize temporal dynamics through statistical or spectral descriptors, offering greater transparency since individual features can be directly inspected—though sometimes at the expense of sensitivity to subtle temporal patterns. Symbolic models discretize signals into tokens or words that encode frequency and pattern information, striking a favorable balance between interpretability and efficiency, as symbolic motifs can be traced back to meaningful temporal behaviors. Shapelet-based classifiers identify discriminative subsequences associated with particular classes, delivering very high interpretability since the learned patterns are directly observable, albeit at the cost of computationally intensive search procedures. Kernel-based models and deep learning architectures achieve high to state-of-the-art accuracy by inducing complex nonlinear representations, yet these transformations remain largely opaque and typically demand substantial computational resources. Ensemble and hybrid methods further elevate predictive performance through model aggregation, but at the cost of reduced interpretability, as the aggregation process obscures individual component contributions while increasing overall training overhead.

2.4.1 Univariate Time Series Classification

Building on the broader TSC landscape, UTSC has attracted the greatest share of research attention, giving rise to a rich and diverse taxonomy of methods.

Table 2.2: Summary of common methodological categories in time series classification, characterized by their typical performance in terms of classification accuracy, computational cost, and interpretability.

Category	**Typical Accuracy**	**Computational Cost**	**Interpretability**
Distance-Based	Moderate	Moderate to High	Low (no explanation of match logic)
Feature-Based	Moderate	Moderate	High (features are statistical and inspectable)
Dictionary-Based	Moderate to High	Low to Moderate	High (discrete, frequency-based, semantically traceable)
Shapelet-Based	Moderate to High	High	Very High (subsequences directly map to class patterns)
Kernel-Based	High	Low	Very Low (features are opaque)
Deep Learning	High to SOTA	Very High	Very Low (black-box architectures)
Ensemble/Hybrid	SOTA	Very High	Low (model aggregation obscures logic)

Among the earliest and most intuitive approaches are distance-based methods, which assess similarity by computing pairwise comparisons between sequences. The most widely adopted representative is Dynamic Time Warping (DTW) [88], which performs non-linear sequence alignment to accommodate temporal distortions. Related techniques such as LCSS [89], QDA [90], and decision-tree-augmented baselines like XGBoost [91] also fall within this category. DTW remains a reliable benchmark, particularly for data exhibiting regular or well-aligned patterns, but it carries substantial computational costs and offers limited model transparency. Despite these limitations, its historical prominence and demonstrated reliability have secured its continued role as a standard baseline in benchmarking studies [71].

A second category encompasses feature-based methods, which transform the original time series into vectors of statistical or domain-specific descriptors. TSFresh [92] is a prominent example, automating the extraction of hundreds of time-domain features. Such methods are generally transparent and integrate naturally into conventional machine learning pipelines. Their drawback, however, is that they tend to abstract away the sequential structure of the data, and may demand significant preprocessing or manual feature filtering to remain effective in multivariate or nonstationary settings.

Kernel and convolution-based methods have driven notable gains in both speed and accuracy. Algorithms such as ROCKET [93], MINIROCKET [94], MultiRocket [95], and Hydra [96] apply randomly generated convolutional kernels to produce expansive feature sets, which are then fed into linear classifiers. These models are computationally efficient and consistently competitive in benchmarks, frequently surpassing ensemble

methods by orders of magnitude in runtime. Their fundamental limitation, however, is that the resulting convolution-based features are inherently opaque, making them poorly suited for applications where transparency or model reasoning is required.

Deep learning methods—including FCN [97], ResNet [98], and InceptionTime [99]—remain dominant across many benchmark tasks, yet are marked by high training costs, substantial data requirements, and limited interpretability. Newer architectures such as LITETime [100] attempt to mitigate these shortcomings but continue to depend heavily on architecture-specific tuning. As Spinnato et al. [80] observe, standard deep models frequently underperform without dataset-specific adjustments, reinforcing their impracticality in lightweight or interpretable pipelines.

Ensemble and hybrid models—such as HIVE-COTE [101], HIVE-COTE 2.0 [102], TS-CHIEF [103], and TDE [104]—deliver strong classification accuracy but at a significant cost in terms of runtime and transparency. Although these approaches represent the current performance frontier in TSC [82], they are generally opaque and computationally intensive.

Dictionary-based and symbolic methods constitute a particularly compelling family of TSC approaches, especially in scenarios where interpretability and computational efficiency are priorities. These methods work by converting time series into sequences of discrete symbols, typically followed by frequency-based encoding using constructs such as histograms or BOPs. The foundational technique in this space is SAX [54], which discretizes time series through piecewise aggregation and quantization. SAX yields compact and interpretable representations, though its effectiveness is sensitive to parameter choices—including alphabet size, word length, and aggregation window—which are commonly selected through heuristic means.

Extending SAX, the BOP approach [63] treats symbolic words as features by constructing histograms over sliding windows. BOP is interpretable and computationally lightweight, but its abstraction of temporal relationships between patterns limits its expressive capacity. More expressive, though memory-intensive, variants include SAX-VSM [105], which models class-specific term weights using vector space representations. Slopewise Aggregate Approximation SAX (SAA-SAX) introduces slope-based segment encoding to more faithfully represent underlying trends, yielding improved classification performance [61, 106].

Several methods have sought greater robustness through alternative transformations and parameter ensembles. Symbolic Fourier Approximation (SFA) [107] maps windowed segments to frequency components prior to discretization, enhancing discriminability. This principle underpins BOSS [86], which constructs symbolic word distributions using SFA across varying window sizes. BOSSVS [108] extends this idea by incorporating vector-space weighting of symbolic patterns to further boost classification. Nonetheless, both BOSS and BOSSVS discard the original temporal ordering, sacrificing interpretability, and incur notable memory overhead through large symbolic dictionaries and multiple parameterized models.

WEASEL [87] and its successor WEASEL 2.0 [77] improve upon BOSS through statistical feature selection and dilated windows, achieving strong empirical results. This comes, however, at the cost of enlarged feature sets and increased runtime and memory demands—drawbacks that are further amplified in multivariate settings. MrSQM [109] offers a middle ground by using compressed discriminative subsequences to reduce model size while partially preserving interpretability.

More recent work has explored ensemble and multiresolution symbolic modeling. TDE [110] aggregates symbolic features extracted under diverse configurations

into an ensemble, improving robustness at the expense of computational cost and interpretability. SCALE-BOSS [78] and its multiresolution extension SCALE-BOSS-MR [79] advance this further by aggregating symbolic representations across multiple window sizes, dilations, and trend encodings. While these frameworks attain strong accuracy, they depend on heuristic feature engineering and ensemble diversity rather than principled parameter optimization, leading to high memory consumption and longer execution times.

By contrast, MR-SEQL [111] preserves interpretability by learning class-discriminative symbolic subsequences for use in a linear model. The approach is compact and transparent, but is constrained by its univariate focus and the absence of global parameter optimization. The recently introduced BORF [80] blends symbolic and convolutional ideas by applying SAX over dilated and strided windows, maintaining interpretability through simple decision tree classifiers with rule-based outputs. However, it applies a fixed configuration uniformly across datasets and performs no optimization of symbolic parameters.

Recent developments in symbolic TSC have also explored hybrid combinations of SAX and neural architectures. One such approach encodes SAX words as graph nodes and learns graph-based embeddings through Graph Neural Networks, achieving competitive performance on UCR benchmarks [112].

Further interpretability-oriented approaches include interval-based methods such as c-RISE [113], CIF [85], and QUANT [114], which characterize temporal behavior over local segments. Shapelet-based classifiers—including STC [115], RDST [116], and RSF [117]—identify discriminative subsequences directly from the data, offering high interpretability at the cost of computationally demanding search procedures.

2.4.2 Multivariate Time Series Classification

While UTSC has given rise to a well-developed ecosystem of benchmarked algorithms, MTSC poses a distinct set of challenges that remain comparatively underexplored. In practical settings—such as industrial diagnostics, physiological monitoring, and smart manufacturing, time series data are inherently multichannel, demanding that models account for both within-channel temporal dynamics and dependencies across channels. Two broad methodological directions have emerged in response to this complexity.

The first involves adapting existing UTSC methods to multivariate data by treating each channel independently, applying univariate classifiers per dimension, and combining predictions through ensembling or majority voting [71]. This strategy is simple and modular, but tends to overlook inter-channel correlations that can be decisive for accurate classification. The second involves developing methods natively designed for MTSC, explicitly modeling dependencies across dimensions. Such approaches vary considerably in how they represent multivariate interactions—from shared feature spaces to graph structures and convolutional attention mechanisms—and differ substantially in both interpretability and computational cost [81].

Early MTSC work extended distance-based classifiers, particularly DTW, to multivariate settings. $\mathrm{DTW_I}$ and $\mathrm{DTW_D}$ [118, 119] perform alignment either independently per channel or jointly across all channels, while MDDTW [120] proposes an ensemble-based variant aimed at balancing these two perspectives. Although effective at capturing alignment-based similarity, these models are computationally demanding and offer limited interpretability, particularly in high-dimensional settings. $\mathrm{DTW_D}$, while more faithful in modeling true signal dependencies, scales poorly and is sensitive to normalization choices [71]. Despite these limitations, it remains a strong baseline

in many benchmarks and serves as a reference point in our evaluations.

Some approaches instead rely on handcrafted or descriptive features derived from MTS. CMFMTS [39] is a notable example, computing complexity-based metrics across dimensions and aggregating them into a transparent feature set. The method is distinguished by its low computational footprint and high interpretability, requiring neither deep architectures nor extensive preprocessing. Its main limitation is a lack of automated feature selection or parameter tuning, which constrains its adaptability across diverse datasets.

Shapelet-based methods offer inherently interpretable models by locating short, discriminative subsequences strongly associated with particular classes. Extensions to MTS include Independent Shapelets [121], which learn per-dimension shapelets separately, as well as UFS [122] and gRSF [117], the latter generalizing random shapelet forests to multivariate settings. These methods support clear visual and semantic inspection of class-defining patterns, but involve computationally expensive search procedures—particularly when cross-channel interactions must be considered—and scale poorly with dataset size and dimensionality.

Symbolic and dictionary-based representations have likewise been adapted for MTSC. WEASEL+MUSE [123] applies symbolic word extraction and feature selection independently across dimensions before merging them into a unified feature space. While this achieves strong accuracy, it is memory-intensive and discards temporal structure, complicating interpretation [71, 124]. SMTS [125] takes a different approach, jointly modeling all channels through supervised tree-based symbol learning over raw time indices and first differences, yielding a scalable and interpretable multivariate representation. LPS [126] captures localized dependencies via tree-based autoregressive patterns, while mv-ARF [127] extends this with multitask autoregressive modeling, producing compact features that directly reflect inter-variable dynamics. MR-PETSC [128] explores interpretable pattern mining in time series, though its adaptation to multivariate settings remains insufficiently defined. BORF [80] applies SAX over dilated windows with decision tree classifiers to preserve interpretability through rule-based outputs, but relies on a fixed heuristic configuration without any optimization of symbolic parameters.

Z-Time [82] provides an interpretable interval-based solution through event abstraction and linear modeling. It operates efficiently and produces transparent decision rules, though it depends on handcrafted representations.

Deep neural networks have become the dominant direction in recent MTSC research, achieving state-of-the-art performance through their capacity to model complex temporal dependencies. However, these models require large datasets, significant computational resources, and remain effectively opaque in practice. MLSTM-FCN and ALSTM-FCN [129] combine convolutional and recurrent layers to jointly capture local patterns and long-range dependencies, though they suffer from poor runtime performance and low interpretability. TapNet [130] and RTFN [131] introduce attention mechanisms to improve feature selection and cross-channel alignment, enhancing accuracy while still requiring substantial tuning and computation, and without explicitly modeling inter-series relationships. Graph-based neural networks (GNNs) take a more structured approach by treating each univariate series as a node and encoding inter-variable dependencies through edges. MF-Net [73] combines temporal attention with graph-based reasoning to jointly capture spatial and temporal patterns, while MTHetGNN [132] and CauGNN [133] leverage heterogeneous and causal dependencies respectively within similar graph-centric frameworks. Homenda

et al. [124] convert MTS data into image-like representations for CNN processing, achieving strong results at the cost of obscured temporal structure. DKN [74] and Dyformer [75] integrate convolution, transformers, and frequency decomposition to model intricate dependencies, though both incur high computational cost and limited interpretability. MTS2Graph [76] combines CNNs and GNNs by constructing graphs from activation maps, improving reasoning ability while adding post-hoc analytical complexity.

Ensemble frameworks such as HIVE-COTE 2.0 [102] and LCEM [134] deliver state-of-the-art accuracy by combining large and diverse collections of base learners, but at the expense of slow runtimes and limited transparency. XEM [135] represents a partially interpretable alternative, though it remains constrained by memory demands and opaque final models.

In summary, MTSC approaches span a wide spectrum—from straightforward adaptations of UTSC methods to highly specialized deep and ensemble architectures. While deep and ensemble models dominate benchmark leaderboards, they frequently sacrifice interpretability and efficiency, two qualities of growing importance in domains such as smart manufacturing and healthcare [72]. Symbolic approaches remain among the few families of methods capable of balancing accuracy, interpretability, and computational practicality, particularly when paired with principled optimization strategies.

2.5 Synthesis

This chapter has traced two threads that converge on the remainder of this thesis. The first, developed in Section 2.1, established time series representation as a landscape of distinct encoding strategies and situated SAX as a specific point within that landscape: a data-adaptive, symbolic representation obtained through PAA and Gaussian discretization, satisfying the perfect-mapping and tight-bounding properties that make a symbolic encoding useful for retrieval and classification alike. Section 2.2 then detailed this representation's construction in full, including the BOP aggregation step that converts a single SAX string into a position-invariant feature vector.

The second thread, developed in Sections 2.3 and 2.4, shifted from representation to classification, and showed that the same trade-off recurs at the classifier level: Table 2.2 and the taxonomy of UTSC and MTSC methods that follows it confirm that no algorithmic family – distance-based, feature-based, dictionary-based, shapelet-based, kernel-based, deep-learning, or ensemble – achieves high accuracy, low computational cost, and high interpretability simultaneously. Symbolic and dictionary-based methods recur throughout this taxonomy as one of the few families capable of balancing all three dimensions reasonably well, but the same survey also surfaced the specific weaknesses that keep them from doing so more fully: parameter sensitivity in SAX's own construction, BOP's loss of sequential order once symbols are aggregated into a histogram, and a comparative scarcity of principled, automated tuning strategies relative to the heuristic or ensemble-based approaches that dominate elsewhere in the field.

These limitations are not incidental to this thesis; they are its starting point. Chapter 3 takes up the representation thread directly, conducting a systematic review of the SAX literature itself to catalogue how the method's specific weaknesses have been addressed to date, and organizing the result into a taxonomy that the remaining chapters of this thesis build on. The recurring gaps identified in this chapter's

classification taxonomy – particularly BOP's loss of sequential order and the field-wide reliance on heuristic parameter choice – resurface directly as motivating problems in later chapters: Chapter 5 addresses principled parameter selection through global optimization, and Chapter 7 returns to BOP's order-blindness through a graph-based representation built on top of the symbolic vocabulary introduced here. The chapters in between develop original extensions of SAX itself, addressing weaknesses Chapter 3's taxonomy will identify in detail.

This page has been left blank intentionally.

Chapter 3

A Structured Overview of SAX-Based Methodologies

As established in Section 2.1, the SAX method occupies a prominent position among time-series representation techniques. Its combination of dimensionality reduction, noise smoothing, computational efficiency, and support for lower bounding has made it a reference point for symbolic time series analysis, with successful applications spanning classification, clustering, anomaly detection, motif discovery, and similarity search across domains as diverse as healthcare, finance, and industrial monitoring [54, 57–59, 61]. The method's interpretability — each symbol directly corresponds to the amplitude level of a temporal segment — and its well-understood theoretical guarantees further consolidate its role as a first-choice tool for a broad spectrum of data mining tasks. Despite these numerous merits, however, SAX is not without limitations, and its widespread adoption has naturally attracted a substantial and growing body of research aimed at extending, refining, and adapting the original method. This chapter is based on the work published in [55] and presents a systematic examination of the SAX variations landscape. We first describe the systematic literature review through which the relevant SAX variations were identified and collected, and then present the recurring weaknesses of the original method that emerged from this review. These weaknesses form the basis of a proposed taxonomy that organizes the identified variations according to the specific limitation each addresses. Representative methods from each category are subsequently analyzed in terms of their design rationale, contributions, and limitations.

3.1 A Systematic Review of SAX Variations

To obtain a comprehensive and unbiased picture of the research landscape surrounding SAX-based methodologies, a systematic literature review was conducted following an established protocol. The review targeted all published works indexed in the Scopus database that describe variations of the original SAX method. The search terms *"SAX"* and *"Symbolic Aggregate Approximation"* were applied against the title, abstract, and keyword fields of stored records, covering the period from 2003 — the year of SAX's original publication [56] — through March 2023. The search was restricted to works belonging to the subject areas of computer science or engineering, and only English-language publications were considered. Deliberately avoiding the logical conjunction of the two search terms ensured that works citing only the abbreviated form were not inadvertently excluded.

This chapter is based on the following publication: [J1].

The initial search returned 817 documents. A thorough examination of each record revealed that the term "SAX" appears across multiple unrelated scientific domains, including networking, astronomy, radiology, chromatography, and oceanography; all such works were excluded. Among the remaining records pertinent to time series analysis, further exclusion criteria were applied. Works were excluded if they described variations of already-identified SAX variations rather than novel contributions; if they employed SAX solely as a baseline for comparison with other representation methods; if they referenced SAX variations for comparative purposes without providing a substantive description of those variations; or if the proposed modification concerned only the distance measure used to compare SAX representations, without altering the representation itself.

Following this filtering process, a general list of 245 original SAX-related documents was obtained, encompassing both variations and applications of the original method. This list was subsequently divided into two sublists: one containing identified SAX variations and one containing SAX applications. The present review is concerned with the former, which comprises 81 documents retrieved from the Scopus database. An additional variation was identified through the internal references of the retrieved works, yielding a final corpus of 82 SAX variations. The complete screening and selection process is summarized in Figure 3.1, presented as a PRISMA flow diagram [136].

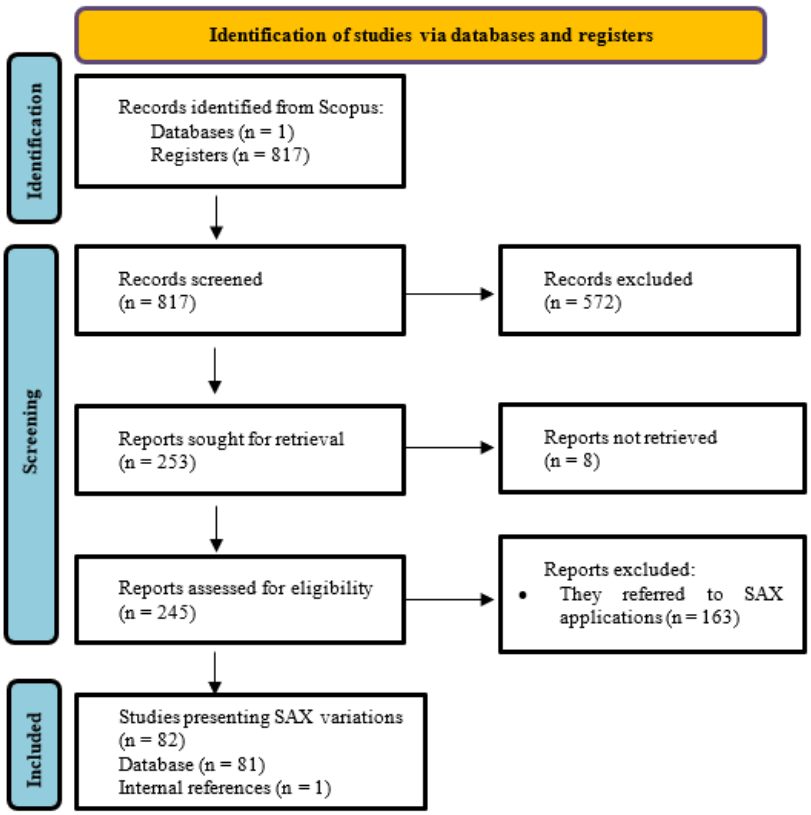


Figure 3.1: PRISMA flow diagram summarizing the systematic review process: database search, screening, eligibility assessment, and final inclusion of SAX variation documents.

Figure 3.2 illustrates the distribution of the 82 identified works over time, alongside

the cumulative citation counts received per year of publication. A notable observation is the marked discrepancy recorded in 2008, where the number of new publications was relatively low yet the corresponding citation count was disproportionately high, suggesting that a small number of highly influential works published in that year attracted substantial subsequent attention.

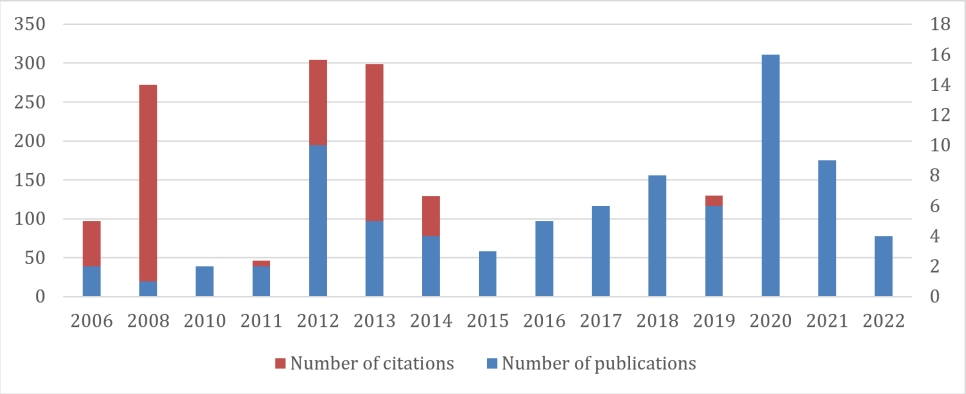


Figure 3.2: Distribution of publications per year related to SAX variations in the final review corpus, alongside the cumulative number of citations received per year of publication.

3.2 Limitations of the Original SAX Method

The systematic review presented in Section 3.1 reveals that the research efforts dedicated to extending and refining SAX converge around five recurring categories of weakness in the original method [55]. These categories emerged from a holistic examination of the full corpus of identified variations, each of which was found to address one or more of the same underlying limitations, and together they provide the basis for the taxonomy proposed in Section 3.3. These five categories are discussed in turn below.

Gaussian distribution assumption. The discretization step of SAX relies on breakpoints derived from the standard normal distribution, implicitly assuming that z-normalized PAA coefficients follow a Gaussian distribution. This assumption, however, does not hold universally. Real-world time series may exhibit skewed, heavy-tailed, or multimodal distributions, and the PAA step itself further distorts the data distribution by shrinking the standard deviation in a manner proportional to both the segment length and the degree of autocorrelation in the series [137]. When applied to non-Gaussian data, the equiprobable binning scheme produces unbalanced symbol assignments, degrading representation fidelity and compromising the validity of subsequent analysis [138–142]. To illustrate this weakness concretely, consider the example in Figure 3.3 [143]. The depicted time series follows a markedly skewed distribution, clearly deviating from the Gaussian assumption underlying the standard SAX discretization. In panel (a), breakpoints are placed according to the fixed Gaussian lookup table; because the true distribution is concentrated in a narrow value range, several bins capture disproportionately large or small fractions of the data, leading to heavily imbalanced symbol assignments. Panel (b) shows the same time series discretized using breakpoints derived from the empirical probability density function

of the data itself; the resulting bins are equiprobable under the true distribution, yielding a more faithful and balanced symbolic representation.

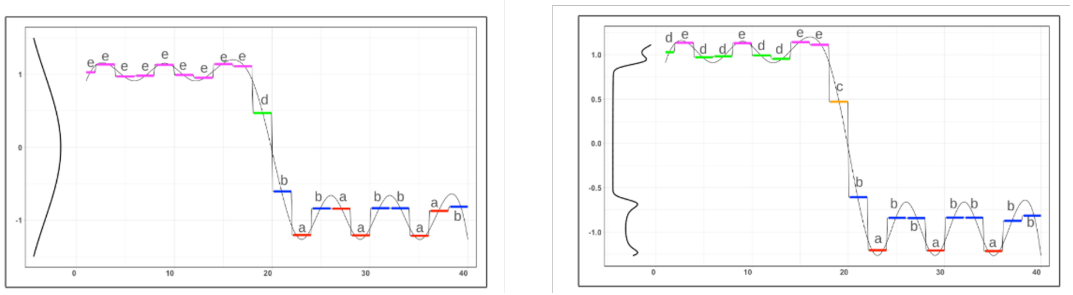
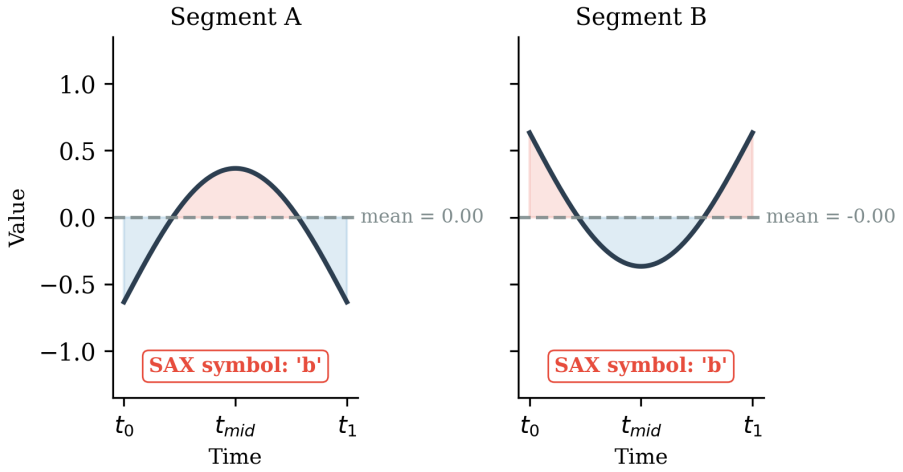


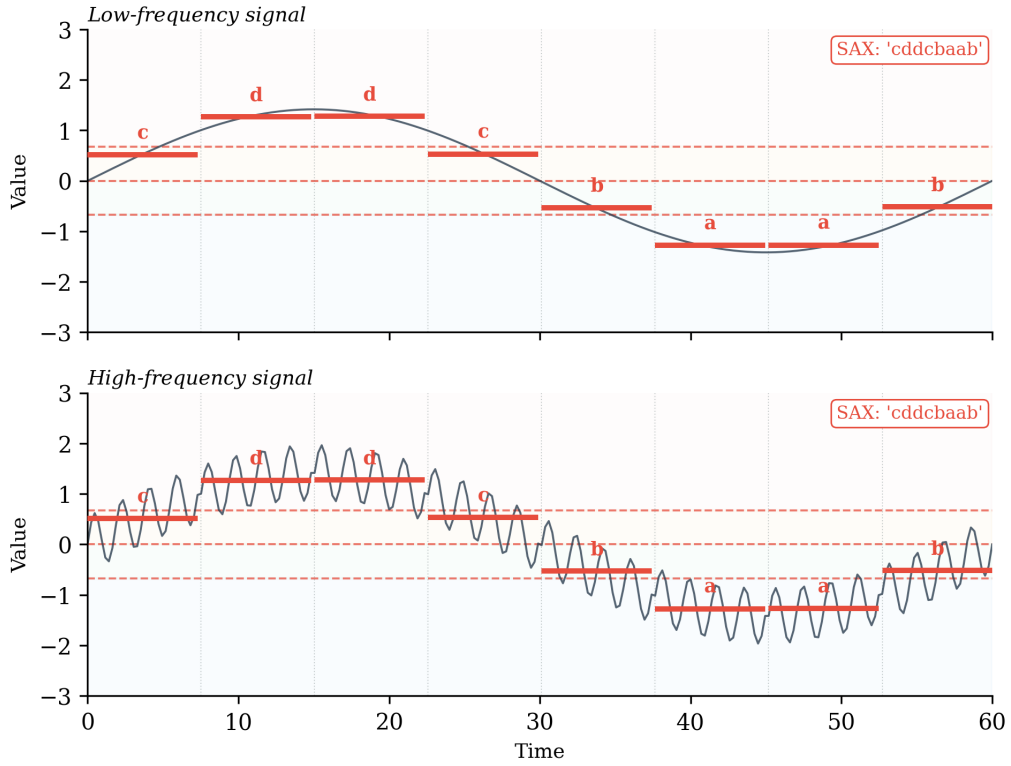
Figure 3.3: The effect of the Gaussian distribution assumption on SAX discretization. The panel on the left shows the symbolic representation produced when fixed Gaussian breakpoints are applied to a non-Gaussian time series: the bins capture unequal fractions of the true distribution (shown on the left), resulting in imbalanced symbol assignments. The panel on the right shows the same time series discretized using breakpoints derived from the empirical probability density function of the data; the bins are equiprobable under the true distribution, yielding a more balanced and faithful symbolic representation.

Lossy compression. SAX encodes each segment solely by its mean value, a scalar summary that discards all within-segment shape and trend information. As a consequence, segments with fundamentally different temporal profiles — for instance, a monotonically increasing segment and a monotonically decreasing one — may share identical mean values and thus receive the same symbol, rendering them indistinguishable in the symbolic representation. This information loss is particularly pronounced for high-frequency or volatile signals, where local extrema and intra-segment dynamics carry substantial discriminative content [144–148]. To sum up, the lossy compression problem manifests in two distinct ways, both illustrated in Figure 3.4. First, as shown in panel (a), segments with fundamentally different temporal profiles — an ascending arc and a descending arc — may share an identical mean value and therefore receive the same SAX symbol, rendering their opposing trends indistinguishable in the symbolic representation. Second, panel (b) demonstrates that signals with markedly different volatility profiles — one smooth and low-frequency, one highly oscillatory — can produce identical PAA coefficients and, consequently, the same SAX word, despite conveying entirely different dynamic behavior.

Parameter selection. The SAX transformation requires the manual specification of several interdependent parameters, summarized in Table 3.1: the sliding window length, the PAA segment size, the word length, and the alphabet size. The optimal configuration is highly data-dependent and cannot be determined analytically, yet parameter choice exerts a strong influence on representation quality and downstream task performance. As a general rule, time series with smooth, slowly varying patterns are well served by a short word length and large segment size, whereas rapidly changing signals may require a longer word and finer segmentation to preserve discriminative fluctuations [54]. In the absence of prior knowledge about the dataset, Lin et al. recommend a word length of three or four and an alphabet size of four as reasonable defaults [63]. However, these remain heuristics rather than principled criteria, and



(a) Trend omission: two segments with opposite temporal profiles share an identical mean value and are assigned the same SAX symbol.



(b) Volatility loss: a smooth low-frequency signal and a highly oscillatory signal produce identical PAA coefficients and the same SAX word, despite their fundamentally different dynamic behavior.

Figure 3.4: The lossy compression problem in SAX. The mean-based PAA aggregation discards both intra-segment trend (a) and volatility information (b), causing structurally distinct signals to collapse onto the same symbolic representation.

practitioners typically resort to exhaustive grid search or trial and error to identify a suitable configuration. Figure 3.5 illustrates how substantially the symbolic output varies across three different parameter configurations applied to the same signal,

highlighting the sensitivity of the representation to parameter choice and the difficulty of selecting an appropriate configuration without task-specific tuning.

Table 3.1: Parameters of the SAX method that must be predefined, alongside a brief description.

Parameter	Description
Window length	The number of consecutive data points included in each sliding window applied to the time series.
PAA segment size	The number of data points within each segment over which the mean value is computed to produce a PAA coefficient.
Word length	The number of symbols comprising a single SAX word, determined by the number of PAA segments into which the window is divided.
Alphabet size	The number of distinct symbols used to discretize the PAA coefficients. It controls the granularity of the representation and determines the placement of the breakpoint values.

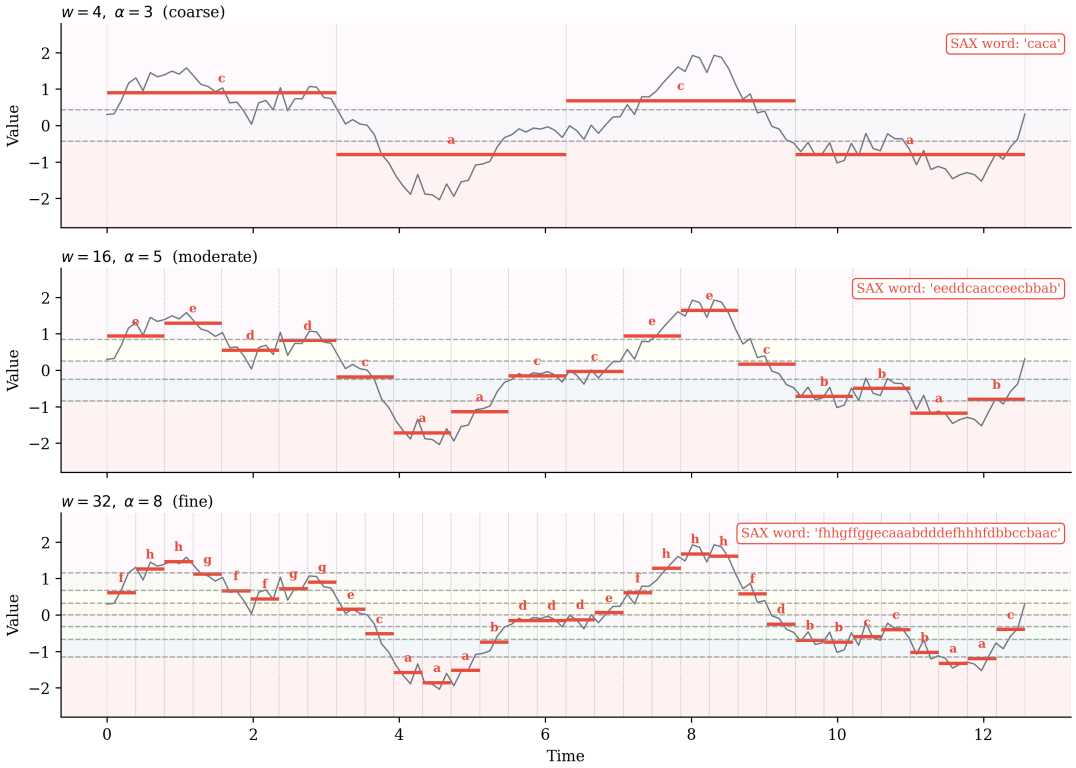


Figure 3.5: The effect of parameter configuration on the SAX representation of the same signal. Three configurations of increasing resolution are shown: coarse ($w = 4$, $\alpha = 3$), moderate ($w = 16$, $\alpha = 5$), and fine ($w = 32$, $\alpha = 8$). The PAA step function (red) and the resulting SAX word (top right of each panel) change substantially across configurations, illustrating the sensitivity of the symbolic representation to parameter choice.

Symbol mapping ambiguity. SAX assigns each PAA coefficient to a symbol based on which fixed interval it falls into. When a coefficient lies close to a breakpoint, small perturbations in the signal — whether due to noise or minor temporal variation — can cause it to cross the boundary and receive a different symbol, even though the underlying patterns are perceptually similar. Conversely, coefficients that are numerically distant but fall within the same bin receive identical symbols despite representing distinct signal behaviors. This rigidity of the fixed breakpoint scheme weakens the tightness of the lower bound and introduces instability in the symbolic representation [149].

Univariate design. SAX was originally formulated for univariate time series and does not natively account for the correlations and joint dynamics that arise in multivariate settings. While the method can be applied independently to each variable of a multivariate signal, this channel-wise treatment ignores inter-variable dependencies entirely, and the resulting per-channel representations cannot capture cross-dimensional patterns that are often the most informative features for tasks such as activity recognition, fault diagnosis, or physiological monitoring. This limitation is illustrated in Figure 3.6, where three mutually correlated channels of a multivariate signal are each represented independently via SAX. Although the channels exhibit clear inter-variable structure, the individual symbolic representations are derived in complete isolation, rendering any cross-channel relationships invisible to the resulting feature space.

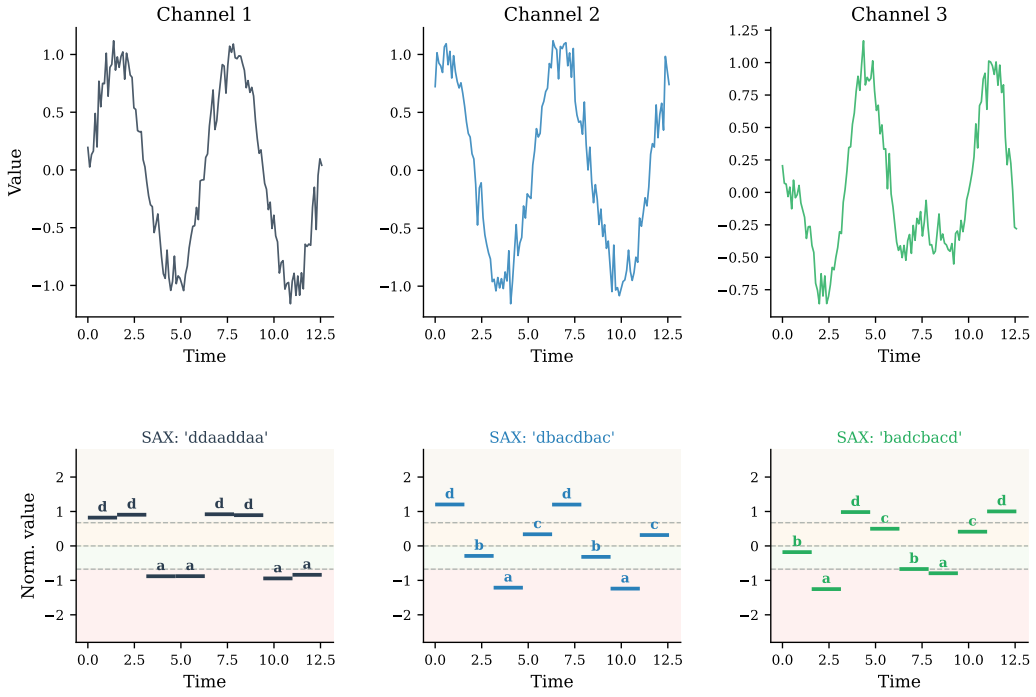


Figure 3.6: The univariate limitation of SAX applied to a multivariate signal. The top row shows three correlated channels of the same signal; the bottom row shows their independent SAX representations. Despite the clear inter-channel dependencies visible in the raw signal, the symbolic representations are derived in complete isolation, discarding all cross-dimensional structure.

3.3 A Proposed Taxonomy of SAX-Based Methodologies

The five weakness categories identified in Section 3.2 provide a natural and principled basis for organizing the SAX variations collected through the systematic review. While the primary motivation behind each variation is to address one or more of these recurring limitations, the corpus also contains works that extend SAX in other directions without targeting a specific weakness explicitly. These works are retained in the taxonomy under a separate non-categorized group, ensuring completeness of coverage.

The proposed taxonomy is illustrated in Figure 3.7, which organizes the identified variations into six groups: five corresponding to the weaknesses discussed in Section 3.2, and one collecting the non-categorized variations. This classification provides a structured map of the SAX variations landscape, enabling researchers to identify the most suitable method for a given limitation or application context, and to recognize where research gaps remain.

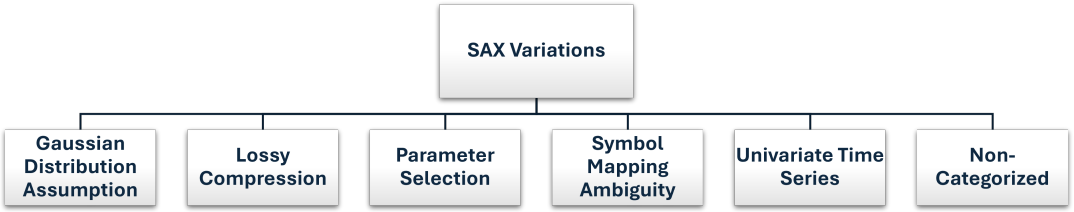


Figure 3.7: The proposed taxonomy of SAX-based methodologies. Variations are organized into six groups according to the primary weakness of the original SAX method that each addresses. A non-categorized group collects variations that extend SAX without targeting a specific limitation.

Figure 3.8 reports the distribution of the 82 identified variations across the six groups, expressed as a percentage of the total corpus. The group addressing lossy compression accounts for the largest share of publications by a considerable margin, reflecting the recognized severity and practical impact of the mean-based aggregation problem. The parameter selection group constitutes the second largest, consistent with the widely acknowledged difficulty of choosing appropriate SAX configurations across diverse datasets. At the other extreme, the symbol mapping ambiguity group contains only two works, suggesting that this weakness, while conceptually well-defined, has attracted comparatively little dedicated research attention and may represent an open direction for future work.

It is worth noting that certain variations address more than one weakness simultaneously — for instance, a method that introduces adaptive segmentation may simultaneously tackle both the parameter selection problem and the lossy compression problem. In such cases, the variation is assigned to the group corresponding to its primary stated contribution, as declared by the original authors.

3.3.1 Category-by-Category Analysis

Gaussian Distribution Assumption

Among the variations addressing the Gaussian distribution assumption, three works stand out in terms of citation impact: **aSAX**, **GASAX**, and **REWS**.

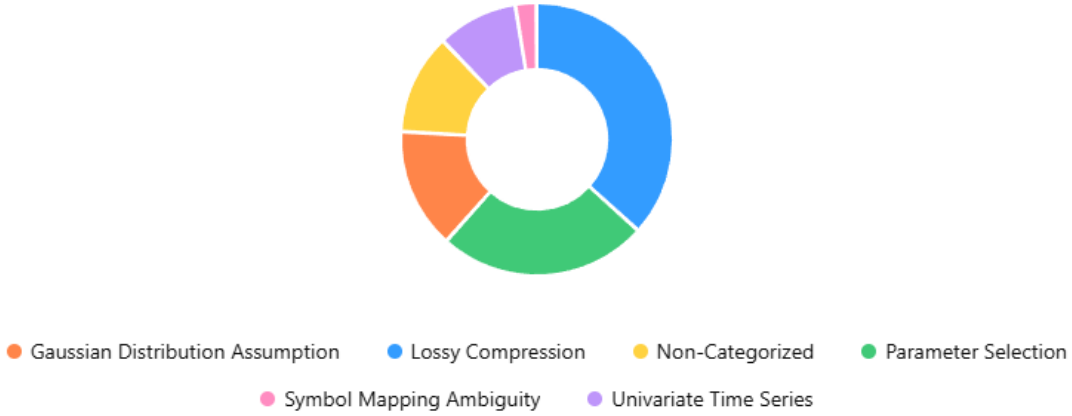


Figure 3.8: Distribution of the identified SAX variations across the six taxonomy groups, expressed as a percentage of the total corpus. The lossy compression group accounts for the largest share of publications, while symbol mapping ambiguity remains the least explored category.

aSAX (adaptive SAX) [150] replaces the fixed Gaussian breakpoints of the original SAX with adaptive breakpoints obtained through a one-dimensional clustering procedure, equivalent to the Lloyd algorithm [151], applied during a pre-processing phase. The alphabet size determines the number of clusters, with cluster centers initialized using the standard Gaussian breakpoints to guide convergence. Empirical evaluation shows that aSAX performs notably well on datasets that do not follow a Gaussian distribution, and it achieves a tighter lower bound and greater pruning power than the original SAX. As a pioneering contribution, aSAX remains a frequent point of reference for subsequent work addressing the Gaussian assumption problem. However, because it relies on piecewise constant approximations, aSAX is unable to represent extreme points of the time series [152], and the clustering procedure must be repeated for every PAA segment, substantially increasing processing and storage costs. This computational overhead limits its applicability in resource-constrained or streaming scenarios.

GASAX [153] takes a different approach, employing a genetic algorithm to evolve a set of breakpoints that minimize classification error, without assuming any particular underlying distribution. Because the breakpoints are evolved rather than looked up, GASAX requires no data normalization and can in principle accommodate any alphabet size, though the size itself must still be specified by the user. Experimental results show that GASAX outperforms the original SAX on standard benchmark datasets under matched parameter settings. The same author later proposed **DESAX** [154], which follows the same evolutionary principle but replaces the genetic algorithm with differential evolution, reporting further improvements in classification accuracy over GASAX. Both methods share the same fundamental limitation: the evolutionary search substantially increases computational cost relative to SAX, requires a dedicated training phase, and — being supervised, since classification error serves as the fitness function — depends on the availability of labeled data.

REWS (Relative Entropy Weibull-SAX) [155] is a data-driven framework for asset health monitoring that constructs a health indicator and performs health-stage segmentation for degradation modeling. At its core, REWS employs **Weibull-SAX**, a discretization scheme that replaces the standard Gaussian breakpoints with breakpoints

derived from the Weibull distribution — a distribution well suited to modeling the skewed and often censored failure-time data encountered in reliability engineering. Relative entropy is then used to quantify the divergence between an asset's current symbolic representation and its representation under normal operating conditions, providing a principled basis for distinguishing healthy from degrading states. While REWS demonstrates the value of domain-specific distributional assumptions and requires no expert knowledge to operate, it is tailored to the asset-degradation domain and still requires an optimization stage to determine its parameter values, limiting its direct transferability to other application areas.

Lossy Compression

Lossy compression constitutes by far the largest category in the proposed taxonomy, and the three most cited variations within it — **ESAX**, **1d-SAX**, and **SAX-TD** — map directly onto the two facets of the problem illustrated in Figure 3.4. ESAX primarily addresses the volatility-loss facet (Figure 3.4b), while 1d-SAX and SAX-TD both address the trend-loss facet (Figure 3.4a).

ESAX (Extended SAX) [144] tackles volatility loss directly by representing each segment with three symbols — corresponding to the minimum, maximum, and mean values — instead of the single mean-based symbol used in the original SAX. This straightforward extension preserves information about local extrema that would otherwise be discarded, which is particularly valuable for high-frequency or volatile signals such as financial time series. The method retains a proof of lower bounding, is intuitive, and is straightforward to implement. However, tripling the number of symbols per segment proportionally increases both the dimensionality of the representation and its computational cost, and — like the original SAX — ESAX requires all SAX parameters to be predefined and is not designed for streaming applications.

1d-SAX [156] addresses trend loss by encoding, for each segment, both the average value and the local slope obtained via linear regression into a single composite symbol. This symbol is drawn from a combined alphabet whose breakpoints separately discretize the average and the slope. As a result, the number of segments — and hence the length of the SAX word — remains unchanged relative to the original SAX, but the effective size of the alphabet grows multiplicatively with the number of breakpoints chosen for each of the two components. This introduces an additional configuration challenge: the allocation of breakpoints between the average and slope components is not determined analytically, and an inappropriate balance can either under-represent trend information or unnecessarily inflate the alphabet. Moreover, 1d-SAX retains the Gaussian assumption for both the average and slope distributions, inheriting the limitation discussed in Section 3.2.

SAX-TD (SAX Trend Distance) [157] takes a different approach to trend loss, augmenting each segment's mean-based symbol with a *trend variation* value, computed as the difference between the segment's start and end points relative to the point corresponding to the segment mean. These trend variations are appended to the symbolic representation as real-valued components, yielding a mixed symbolic–numeric representation. Despite this addition, SAX-TD achieves a tighter lower bound than the original SAX distance and lower classification error rates, at computational cost comparable to SAX itself, and its performance is reported to be largely insensitive to the choice of alphabet size. Its principal drawback is the mixed representation format, which complicates direct compatibility with methods designed for purely symbolic SAX output, and — as with all SAX-based methods — it still requires the underlying

SAX parameters to be defined in advance.

Parameter Selection

Among the variations addressing the parameter selection problem, three works stand out in terms of citation impact: **SAX-VSM**, **SensorSAX**, and the pair of adaptations proposed by **Nguyen et al.**

SAX-VSM (SAX-Vector Space Model) [105] combines SAX with the classic Vector Space Model, using the term frequency–inverse document frequency (tf-idf) weighting scheme to automatically discover and rank symbolic patterns according to their discriminative significance for each class, with cosine similarity serving as the comparison metric. Patterns that are characteristic of a single class receive higher weights, while patterns common across classes are down-weighted, directly improving classification accuracy over the original SAX. Critically, SAX-VSM addresses the parameter selection problem by employing the DIRECT global optimization algorithm to automatically determine the SAX parameters, removing the need for manual tuning entirely. This combination of automatic parameter selection and interpretable, class-discriminative pattern weighting likely accounts for the method's substantial and sustained citation impact. Its principal drawback is computational: the optimization and weighting procedures introduce considerable complexity, and a dedicated training phase is required before the method can be applied.

SensorSAX [158] instead targets the window length parameter, making it adaptive to the local volatility of the signal. The PAA segmentation stage is modified so that windows expand during periods of low activity — identified via low local standard deviation — and contract during periods of high activity, where finer temporal resolution is needed to preserve key features. The method takes as input a minimum and maximum window length together with a standard deviation threshold, and was shown to improve reconstruction error rates across datasets of varying variance. SensorSAX is intuitive and straightforward to implement, and is well suited to online processing. However, the variable window length entails additional computation relative to the original SAX, the method still requires several parameters to be defined in advance, and its evaluation has been limited to a specific sensor-data domain.

Nguyen et al. [111, 159] proposed two adaptations of the SEQL sequence learner — a method originally designed to identify discriminative subsequences within long discrete sequences over large alphabets using a greedy, all-subsequence search. The first adaptation, **SAX-VSEQL**, addresses the fixed SAX word length constraint by learning variable-length SAX words directly from the data. The second, **SAX-VFSEQL**, extends this further by learning *fuzzy* variable-length words, directly targeting the computational burden of parameter tuning. Both approaches combine symbolic representations obtained under different parameter settings and domains, using a greedy feature selection strategy to retain only the most discriminative ones — yielding classifiers that remain simple and interpretable while requiring no manual parameter tuning. This flexibility also makes the approach naturally extensible to multivariate time series. The principal limitation is computational cost: combining multiple symbolic representations across parameter settings is considerably more expensive than evaluating a single fixed configuration.

Symbol Mapping Ambiguity

The symbol mapping ambiguity category is, by a considerable margin, the least populated in the proposed taxonomy, comprising only two identified variations: **rSAX** and **ASAX**.

rSAX (Random Shifting SAX) [149] improves the tightness of the lower bound by generating τ symbolic representations of the same series, each obtained by randomly shifting the breakpoints by a small amount. Across these representations, near-boundary coefficients are more likely to be consistently mapped to the same symbol, softening the rigid boundary effect without enlarging the alphabet — at the cost of the need to decide how many shifted versions τ to generate, on top of the usual alphabet size and word length.

ASAX (Altered SAX) [160] instead removes the PAA step entirely, mapping each individual sample directly to its nearest discretization level and retaining a sample only when its assigned symbol changes. The result is a sequence of symbol–timestamp pairs that records transitions precisely as they occur, rather than smoothing them away through segment-wise averaging.

The limited size of this category suggests that symbol mapping ambiguity remains comparatively underexplored despite being a well-documented weakness of the original method.

Univariate Design

Three works stand out in this category: **MBOP**, **SAX-ARM**, and the extensions proposed by **Mohammad and Nishida (2014)**. Each addresses the cross-channel blindness illustrated in Figure 3.6 through a different mechanism: combining symbols across channels into joint words, restricting attention to anomalous symbols shared across channels, or preprocessing channels jointly before the SAX pipeline is applied.

MBOP (Multivariate Bag-of-Patterns) [161] extends the BOP model by forming *multivariate words*: at each time step, the SAX symbols from all variables are combined into a single joint symbol, and a frequency histogram of these joint words is then constructed as in standard BOP. Inverse document frequency and inverse frequency weighting are additionally used to emphasize unusual patterns and to accommodate series of differing lengths. This directly captures cross-channel co-occurrence, at the cost of the increased complexity and Gaussian assumption inherited from SAX.

SAX-ARM [162] targets the discovery of deviant event patterns across multivariate series via association rule mining. In place of z-normalization, it applies an Inverse Normal Transformation and restricts attention to outlier symbols only, which are collected across all variables at each time segment into a *symbol basket*. Association rules mined from these baskets reveal co-occurring anomalies across channels — a formulation well suited to manufacturing fault monitoring, though its applicability beyond event-pattern discovery remains unexplored.

Mohammad and Nishida (2014) [163] propose three extensions: SAX-PCA, which reduces dimensionality via principal component analysis prior to the standard SAX pipeline; SAX-REPEAT, which applies SAX independently per channel and combines the resulting symbols into an expanded alphabet, subsequently clustered back to the original alphabet size via k-means; and SAX-ZSCORE, which replaces standard normalization with a multidimensional z-score applied jointly across channels before PAA. The resulting representations are noise-resistant and usable standalone or in combination, though at considerable computational cost and with evaluation

limited to motion-demonstration tasks.

Non-Categorized Variations

The non-categorized group collects variations that extend SAX along directions not tied to a single specific weakness from Section 3.2. The two most cited works in the entire corpus belong to this group — **BOP** and **iSAX** — alongside the more recent contribution of **Rezvani et al. (2019)**.

BOP [63], already introduced in Section 2.2.3, is included here for its foundational role rather than as a response to a specific weakness: it provides the frequency-histogram framework underlying several variations discussed in the preceding categories, including SAX-VSM, MBOP, and TBOPE. Its position as the most cited work in the corpus reflects this central, enabling role within the broader SAX ecosystem.

iSAX (indexable SAX) [164] targets scalability rather than representational accuracy. It concatenates SAX symbols into a multi-resolution representation, analogous in spirit to wavelets, in contrast to the single-resolution representation of the original SAX. This multi-resolution structure enables hierarchical index trees with non-overlapping leaf regions and controlled fan-out, supporting extensible hashing and exact similarity search over terabyte-scale time series archives — a capability the original SAX does not provide.

Rezvani et al. (2019) [141] propose an alternative representation paradigm: following PAA, each time series frame is transformed into a unit vector via Lagrangian multipliers, yielding an eigenvector-space representation of local segments. Change points are then identified by applying entropy-based models to sequences of these vectors, detecting reversals between ascending and descending trends. The method was shown to outperform comparable approaches in clustering and change-point detection tasks, though it offers no adaptive mechanism for selecting the window length.

3.4 Discussion and Research Opportunities

The taxonomy presented in this chapter, together with the category-by-category analysis of representative methods, allows several broader observations to be drawn about the state of SAX-based research and the directions that remain comparatively unexplored.

The distribution of variations across the six categories (Figure 3.7) is itself informative. Lossy compression accounts for more than a third of all identified variations, considerably more than any other category. This concentration of research effort is consistent with the centrality of the problem: the mean-based PAA aggregation at the heart of SAX affects every downstream task, regardless of domain, and the resulting loss of trend and volatility information — illustrated in Figure 3.4 — is immediately apparent to anyone applying the method to real data. Parameter selection forms the second largest category, reflecting a more practical concern: even a theoretically sound representation is of limited use if its configuration cannot be determined reliably without extensive manual tuning. Together, these two categories account for over 60% of the corpus, suggesting that representational fidelity and practical deployability have been the dominant drivers of SAX-related research over the past two decades.

A recurring pattern across nearly all categories is a consistent trade-off: methods that improve representational accuracy, whether by adding trend symbols, adaptive breakpoints, or variable-length segments, almost invariably do so at the cost of

increased dimensionality, additional parameters, or higher computational complexity relative to the original SAX. Across the variations reviewed in this chapter, this trade-off is consistently reported descriptively — as separate gains and costs — rather than jointly, for instance as an accuracy gain normalized by the additional computational or representational overhead incurred. This suggests that an explicit cost-benefit framing remains absent from how SAX variations are typically evaluated, and could be a valuable addition to future comparative studies.

Symbol mapping ambiguity remains the most clearly underexplored category, comprising only two variations despite being identified as a distinct and well-defined weakness as early as 2013. A plausible explanation is that this weakness is partially — if indirectly — mitigated by methods developed for other categories: adaptive breakpoint schemes proposed to address the Gaussian assumption problem, for instance, may incidentally reduce the frequency of near-boundary symbol flips, even though this is not their stated objective. Nevertheless, no variation in the corpus addresses symbol mapping ambiguity as a primary objective in combination with the other four weaknesses, leaving open the question of whether a representation could be designed to jointly minimize boundary instability and the other identified limitations.

A further gap concerns evaluation consistency. Among the variations discussed in this chapter, even those addressing the same weakness have typically been evaluated on different numbers and types of datasets, reporting accuracy ranges that vary substantially from one method to the next. This heterogeneity makes direct comparison across variations difficult and complicates the identification of a single "best" method for a given weakness; instead, the appropriate choice remains heavily dependent on the characteristics of the target data and task.

Finally, it is worth noting that the most cited works in the corpus — BOP and iSAX — are also among the oldest, both predating most of the weakness-specific variations discussed in this chapter. This suggests a cumulative advantage effect typical of citation dynamics, whereby early foundational contributions continue to accrue citations as later work builds upon them. More recent variations, including several reviewed in this chapter, may therefore be underrepresented in terms of citation impact relative to their actual contribution, simply due to the limited time elapsed since publication. This observation reinforces the motivation for the present taxonomy: citation count alone is an insufficient guide to the relevance of a SAX variation, and a structured categorization by addressed weakness offers a complementary, and arguably more durable, basis for navigating this body of work.

This page has been left blank intentionally.

Chapter 4

Extending SAX: Multichannel Structure and Trend Recovery

Chapter 3 surveyed the extensive body of work that has extended the original SAX formulation, organizing these contributions according to the specific weakness of the base method that each was designed to mitigate. This chapter presents two original extensions of SAX, developed as a direct response to different limitations. Rather than pursuing a single unified solution, each variant targets one weakness on its own terms, allowing it to be motivated, developed, and evaluated independently.

Section 4.1 introduces the Multichannel Intelligent Icon (MII) [165], which extends SAX to multivariate time series by constructing a spatial correlation across channels at the symbolic level: symbols occurring at corresponding positions across dimensions are combined into multichannel words, and the frequency distribution of these words – the BOP representation – is used as an additional feature for classification. MII is evaluated on a Human Activity Recognition (HAR) task, where it improves both accuracy and sensitivity relative to the single-channel baseline.

Section 4.2 introduces the Slopewise Aggregate Approximation (SAA) [61, 106], which addresses the trend-loss problem by modifying the Piecewise Aggregate Approximation step itself: each segment is characterized by an angle-based descriptor of its local slope rather than, or alongside, its mean value. The resulting symbolic sequences are aggregated to the BOP construction and a nearest-neighbour classifier is deployed.

Taken together, these contributions are best understood not as a single combined method but as independent extensions of SAX, each closing one of the weaknesses identified in Chapter 3's taxonomy. Section 4.5 returns to this point and traces how each extension feeds forward into the work presented in later chapters.

4.1 Multichannel Intelligent Icon

As discussed in Section 3.3, a substantial body of work extends SAX to multivariate settings either by applying SAX independently to each channel and concatenating the per-channel outputs or by reducing dimensionality across channels prior to discretization, typically at the cost of increased representational complexity and a departure from SAX's per-channel symbolic interpretability. MII takes a third route: it leaves the per-channel SAX representation untouched and introduces an additional, lightweight feature that captures cross-channel structure directly in the symbolic domain, building on the existing concept of *Intelligent Icons* [166]. The method is presented and evaluated here in its original single-dataset HAR setting; Chapter 5 later revisits MII

This chapter is based on the following publications: [C1], [C2], [C3].

as one of several feature-extraction strategies subject to systematic, multi-dataset parameter optimization.

4.1.1 Spatial Correlation Across Channels and MII Construction

This section presents the MII feature [165], building on the SAX and BOP formulations already introduced in Section 2.2.

MII extends the symbolic representation framework to the multivariate setting by jointly encoding cross-channel co-occurrence patterns, rather than treating each channel's symbolic sequence in isolation. Given a p-dimensional time series, the SAX transformation described in Section 2.2.2 is applied independently to each channel, yielding p symbolic sequences $S^{(1)}, S^{(2)}, \ldots, S^{(p)}$, where each sequence $S^{(i)} = \{s_1^{(i)}, s_2^{(i)}, \ldots, s_m^{(i)}\}$, $i \in [1, p]$, has the same length m as a result of applying a shared PAA segmentation across all channels.

Rather than constructing a separate BOP histogram per channel as in Section 2.2.3, MII captures cross-channel dependencies by forming *multichannel words*: tuples of symbols drawn from the same relative position across all p sequences. Formally, for each position $j \in [1, m]$, a multichannel word is defined as:

$$\mathbf{W}_j = \left(s_j^{(1)}, s_j^{(2)}, \ldots, s_j^{(p)}\right), \quad s_j^{(i)} \in S^{(i)} \tag{4.1}$$

The ordering of symbols within $\mathbf{W}_j$ follows the fixed channel identities of the dataset (e.g., sensor or variable order), ensuring a consistent and reproducible encoding across samples. Each multichannel word therefore represents a joint symbolic state across all channels at a given temporal position, directly capturing co-occurring patterns that a per-channel BOP representation would treat as independent.

Traversing all positions $j = 1, \ldots, m$ yields a sequence of m multichannel words. The frequency of occurrence of each distinct word is then computed to build a frequency table, analogous to the BOP histogram but defined over the joint symbol space rather than a single channel. Since each channel contributes a symbol from an alphabet of size a, the total number of possible multichannel words is a^p, and the resulting feature vector $\mathbf{h} \in \mathbb{Z}^{a^p}$ has the k-th entry given by:

$$h_k = \sum_{j=1}^{m} \mathbf{1}\left[\mathbf{W}_j = \boldsymbol{\omega}_k\right] \tag{4.2}$$

where $\boldsymbol{\omega}_k$ is the k-th element of the a^p-sized multichannel word vocabulary. This frequency table constitutes the MII feature: a compact, sparse representation in which each entry quantifies how often a specific joint symbolic pattern occurs across channels. Figure 4.1 illustrates this construction, showing how per-channel SAX sequences are aligned to form multichannel words and how their frequencies are tabulated into the MII feature. When reshaped into a p-dimensional grid (or, for visualization purposes, displayed as a 2D heatmap), the resulting structure gives the method its name, since it allows the joint symbolic structure of the multivariate signal to be inspected visually as a compact "icon."

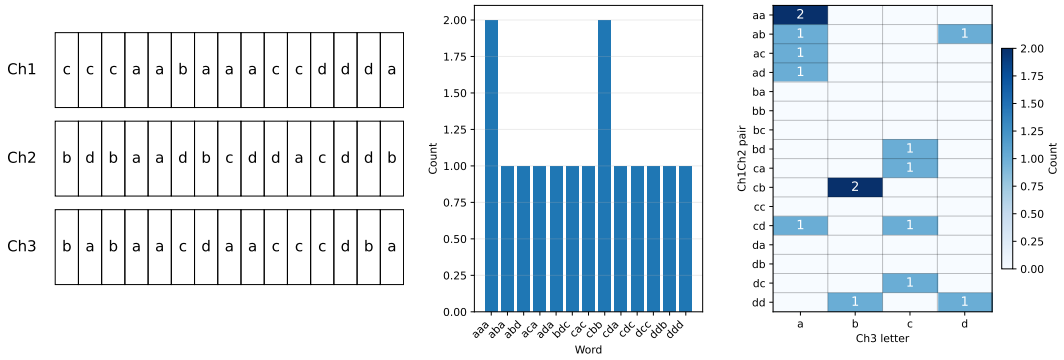


Figure 4.1: Construction of the MII feature. SAX representations of the p channels of a multivariate time series are aligned at corresponding positions to form multichannel words (Eq. 4.1). The frequencies of distinct multichannel words are binned into a histogram (Eq. 4.2), which can equivalently be reshaped into a compact 2D grid (heatmap) to form the Multichannel Intelligent Icon (MII).

The MII feature is interpretable by construction: each entry of $\mathbf{h}$ corresponds to a specific, human-readable combination of per-channel symbols, and its value can be traced back to the exact temporal positions at which that combination occurs, via the shared PAA segmentation.

Algorithm: Construction of Multichannel Intelligent Icons

Step 1. Apply the SAX transformation independently to each of the p dimensions of the multivariate signal, using a shared PAA segmentation across channels.

Step 2. Obtain p symbolic sequences $S^{(1)}, S^{(2)}, \ldots, S^{(p)}$, each of length m.

Step 3. For each position $j \in [1, m]$, form the multichannel word $\mathbf{W}_j$ by concatenating the symbols occurring at position j across all p sequences, following the fixed channel order.

Step 4. Compute the frequency of occurrence of each distinct multichannel word across all m positions, producing the MII feature vector of dimension a^p.

In our related proposed methods, the MII feature is concatenated with the per-channel BOP histograms, allowing the final feature vector to capture both within-channel symbolic dynamics and cross-channel symbolic co-occurrence.

4.2 Slopewise Aggregate Approximation

As discussed in Section 3.3.1, the most common strategy for preserving trend information in a SAX representation is to append one or more additional symbols per segment – describing, for instance, the slope of a fitted line or the maximum and minimum values. SAA follows the same general principle of pairing a slope-derived symbol with the standard mean-derived one, but arrives at it through a distinct architecture: SAA processes each window through two parallel branches. One branch applies standard PAA and discretization to obtain the conventional mean-based SAX string; the other applies SAA's angle-based descriptor (Section 4.2.1) to the same

segments and discretizes the resulting angle series independently. The two resulting symbolic strings are then converted into BOP representations and combined only at the feature extraction stage.

4.2.1 From PAA to SAA: Angle-Based Segment Representation

Recall that PAA divides a time series $X = \{x_1, \ldots, x_n\}$ into m equal-length segments of n/m points each, and represents each segment by the mean value of its points, yielding a reduced series $\bar{X} = \{\bar{x}_1, \ldots, \bar{x}_m\}$ of length m. This mean-value representation discards any information about how the signal evolves *within* a segment: two segments with identical means but opposite trends – one rising, one falling – are mapped to the same PAA coefficient and, subsequently, the same SAX symbol.

SAA computes, in parallel with this standard PAA branch, an angle-based descriptor of each segment's local trend, over the same segmentation of the same window. For segment i, let $(x_{i,1}, y_{i,1}), \ldots, (x_{i,n/m}, y_{i,n/m})$ denote its n/m data points, where x denotes time and y the signal value. Treating the first point of the segment as a fixed origin, a vector is drawn from $(x_{i,1}, y_{i,1})$ to every subsequent point $(x_{i,j}, y_{i,j})$, $j = 2, \ldots, n/m$, and an additional vector is drawn from $(x_{i,1}, y_{i,1})$ to the first point of the *next* segment, $(x_{i+1,1}, y_{i+1,1})$, so that the representation remains continuous across segment boundaries. Each such vector forms an angle with the horizontal (time) axis:

$$\theta_{i,j} = \tan^{-1}\left(\frac{y_{i,j+1} - y_{i,1}}{x_{i,j+1} - x_{i,1}}\right), \qquad j = 1, \ldots, n/m,$$

where for $j = n/m$ the point $(x_{i,j+1}, y_{i,j+1})$ is taken to be $(x_{i+1,1}, y_{i+1,1})$, the first point of the following segment. The n/m angles obtained for segment i are then averaged to give a single representative angle for the segment:

$$\bar{\theta}_i = \frac{1}{n/m} \sum_{j=1}^{n/m} \theta_{i,j}$$

Repeating this for all m segments yields an angle series $\bar{\Theta} = \{\bar{\theta}_1, \ldots, \bar{\theta}_m\}$ of the same length m as the PAA output $\bar{X}$. Geometrically, each segment of the original signal is replaced by a single vector, anchored at the segment's first point, whose orientation equals $\bar{\theta}_i$ – a one-vector approximation of the segment's local slope, smoothed over all the within-segment displacements rather than fit by least squares. $\bar{\Theta}$ is, critically, computed *alongside* $\bar{X}$ rather than in place of it: both series are produced from the same window and the same segmentation, then carried forward independently.

This construction is illustrated in Figure 4.2, which shows both the per-segment angle computation (a) and the resulting slope-vector approximation of the full signal (b).

4.2.2 Discretization and Feature Fusion

Because $\bar{\Theta}$ has exactly the same length m as the standard PAA output, it can be fed into the existing SAX discretization step – using the same alphabet size a and breakpoints derived under the Gaussian assumption – without any modification, producing a second symbolic string of length m, independent of the conventional SAX string derived from $\bar{X}$. Each string is then converted into a BOP representation of word length w, and the two BOP representations – one capturing the distribution of mean-value patterns, the other the distribution of slope patterns – are concatenated

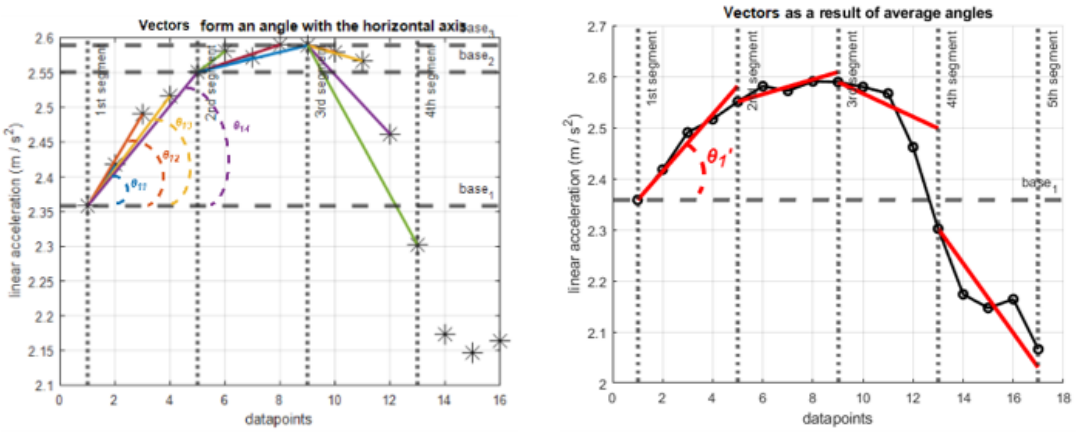


Figure 4.2: Construction of the Slopewise Aggregate Approximation (SAA). (a) Within-segment angle computation from a fixed anchor point. (b) The resulting piecewise slope-vector approximation of the signal, obtained by averaging the angles in (a) for every segment and replacing each segment with a single representative vector.

into a single feature table at the feature extraction stage. The overall pipeline can be summarized as

$$X \longrightarrow \begin{cases} \bar{X} \longrightarrow S_{\text{mean}} \longrightarrow \text{BOP}_{\text{mean}} \\ \bar{\Theta} \longrightarrow S_{\text{slope}} \longrightarrow \text{BOP}_{\text{slope}} \end{cases} \longrightarrow [\text{BOP}_{\text{mean}}, \text{BOP}_{\text{slope}}],$$

where the bracketed term denotes feature-level concatenation rather than symbol-level fusion.

4.3 Experimental Evaluation

MII (Section 4.1) and SAA (Section 4.2) were both evaluated on the same dataset, signal preprocessing pipeline, and parameter configuration, alongside a classical feature-extraction baseline [106], which allows the two proposed methods to be assessed not only against each other under identical conditions, but also against both a non-symbolic baseline and the shared BOP representation each one builds on. This section presents that shared evaluation once, rather than duplicating the dataset and preprocessing description across both proposed methods.

4.3.1 Dataset

Both methods were evaluated on the RealWorld (HAR) dataset [167], a publicly available benchmark for body-worn activity recognition originally designed to study how sensor placement affects classification performance: the source recordings span seven on-body positions (chest, forearm, head, shin, thigh, upper arm, and waist) and six sensing modalities (acceleration, GPS, gyroscope, light, magnetic field, and sound level), of which this evaluation draws on only the triaxial accelerometer and gyroscope channels. The dataset contains recordings from fifteen subjects (eight male, seven female; age 31.9 ± 12.4, height 173.1 ± 6.9 cm, weight 74.1 ± 13.8 kg) performing eight everyday activities: climbing stairs down and up, jumping, lying,

sitting, standing, running/jogging, and walking, sampled at 50 Hz. Each subject performed each activity for approximately ten minutes, except jumping, which was limited to roughly 1.7 minutes due to physical exertion; the contribution of data across genders is approximately balanced, with male and female subjects contributing a similar total recording duration. The same is not true across activities, however: because jumping was capped at a fraction of the duration used for every other class, the dataset is class-imbalanced by construction, with jumping windows substantially underrepresented relative to the other seven activities. Only the accelerometer and gyroscope channels were used, giving six raw signals (x, y, z for each sensor) per subject per activity.

4.3.2 Preprocessing and Segmentation

The accelerometer and gyroscope streams were first synchronized, since sensor sampling instants drift apart over the course of a recording. The initial and final 2% of each signal were then discarded to remove the transitional period before a subject settles into an activity's steady pace. Each of the six resulting signals was denoised using a fifth-order median filter followed by a fifth-order low-pass Butterworth filter with a 20 Hz cutoff – a cutoff sufficient for capturing human body motion, since 99% of its signal energy lies below 15 Hz – and z-score normalized to zero mean and unit variance, removing the offset and amplitude distortions that would otherwise bias distance-based comparison of the resulting symbolic sequences.

The filtered, normalized signals were segmented into sliding windows of $T = 2.56\,\text{s}$, corresponding to $50\,\text{Hz} \times 2.56\,\text{s} = 128$ samples per window, with 50% overlap between consecutive windows. Table 4.1 summarizes the parameter values used throughout, which were fixed across SAX, MII, and SAA-SAX to ensure a fair comparison; in particular, the SAA evaluation deliberately reused the exact parameter values established for the MII study rather than re-tuning them for its own benefit.

Table 4.1: Parameter values used in the joint evaluation of SAX, MII, and SAA-SAX.

Parameter	**Symbol**	**Value**
Window duration	T	2.56 s
Data points per window	n	128
Segments per window	m	32
Alphabet size	a	4
Number of channels (MII word size)	p	3
Window overlap	—	50%

4.3.3 Feature Representations and Classifier

Four feature representations were compared, all derived from the same underlying windows:

- **Feature Extraction**: a classical time- and frequency-domain feature set [106], comprising fifteen statistical and spectral descriptors per window (minimum,

maximum, mean, bandpower, zero-crossing rate, variance, kurtosis, skewness, root-mean-square, median frequency, entropy, Euclidean norm, mean absolute value, sum, and total power), computed independently for each of the six raw signals and concatenated into a single feature vector. This representation serves as a non-symbolic baseline, included to test whether the symbolic representations evaluated here capture discriminative structure that conventional handcrafted features miss.

- **BOP**: the six single-channel BOP representations (one per accelerometer axis, one per gyroscope axis), each computed from the standard PAA → SAX pipeline and concatenated into a single feature vector.
- **BOP+MII**: the six single-channel BOP representations described above, concatenated with the two Multichannel Intelligent Icons (one per sensor), as introduced in Section 4.1.1.
- **SAA-SAX**: the BOP representation derived from the SAA-based angle sequence $\bar{\Theta}$ (Section 4.2.2), following the same discretization and word-length settings as the BOP baseline.

For each representation, the resulting feature table was used to train a 1-Nearest Neighbour classifier. For every activity class, 80% of the windows were randomly assigned to the training set and the remaining 20% to the test set; this random split, training, and evaluation cycle was repeated ten times for all four representations, including Feature Extraction, and the mean accuracy and per-class true-positive rate (TPR) across the ten repetitions are reported below. The original publication of the feature-extraction baseline [106] reports only the mean over these ten repetitions, not the standard deviation; its STD is accordingly shown as not reported (–) in Tables 4.2 and 4.3.

4.3.4 Results

Table 4.2 reports overall classification accuracy for the four representations.

Table 4.2: Mean classification accuracy and standard deviation (STD) over ten repetitions, for Feature Extraction, BOP, BOP+MII, and SAA-SAX. STD for Feature Extraction is not reported (–) in the original publication.

Method	Mean Accuracy (%)	STD (%)
Feature Extraction	81.32	–
BOP	90.13	0.20
BOP+MII	92.39	0.24
SAA-SAX	**96.00**	**0.19**

The gap between the classical feature-extraction baseline and every symbolic representation is the largest in the table: BOP alone already improves on it by 8.81 points, and SAA-SAX widens this to 14.68 points. SAA-SAX additionally improves overall accuracy by 5.87 points over BOP and 3.61 points over BOP+MII, while also achieving the lowest standard deviation among the three symbolic representations.

Table 4.3 breaks this down by activity, reporting the mean TPR and its standard deviation for each method.

Table 4.3: Per-activity true-positive rate (TPR %) ± standard deviation, for Feature Extraction, BOP, BOP+MII, and SAA-SAX, over ten repetitions. STD for Feature Extraction is not reported (–) in the original publication.

Activity	**Feature Extraction**	**BOP**	**BOP+MII**	**SAA-SAX**
Climbing down	61.53	92.51 ± 0.88	95.70 ± 0.63	**97.28 ± 0.26**
Climbing up	77.88	92.47 ± 1.04	94.70 ± 0.98	**96.11 ± 0.94**
Jumping	65.63	95.89 ± 1.18	**96.89 ± 0.91**	96.76 ± 1.11
Lying	91.58	89.92 ± 0.74	92.10 ± 0.37	**96.81 ± 0.48**
Running	91.52	97.19 ± 0.44	97.69 ± 0.46	**98.66 ± 0.20**
Sitting	82.42	80.20 ± 0.75	83.87 ± 0.59	**92.66 ± 0.81**
Standing	80.65	81.62 ± 1.03	85.13 ± 0.91	**92.20 ± 0.51**
Walking	81.28	96.11 ± 0.50	97.22 ± 0.33	**98.15 ± 0.43**
Mean	79.06	90.74 ± 0.82	92.91 ± 0.65	**96.08 ± 0.59**

SAA-SAX achieves the highest TPR for seven of the eight activities, with jumping the sole exception, where BOP+MII is marginally ahead (96.89% vs. 96.76%) – a difference smaller than either method's standard deviation and therefore not meaningful. Progressing from BOP to BOP+MII to SAA-SAX, both proposed methods improve on the symbolic baseline for every activity, and SAA-SAX improves further on BOP+MII for all activities except jumping. The feature-extraction baseline falls short of every symbolic representation on every activity except lying, where it slightly exceeds plain BOP (91.58% vs. 89.92%) – the one case where a handcrafted statistical summary edges out a representation that discards absolute magnitude information, consistent with lying being distinguished primarily by a stable gravitational offset rather than by any repeating local pattern a symbolic word could capture. Its most pronounced weakness is on the two stair-climbing classes, where it trails every symbolic method by 15–31 points (Climbing down: 61.53% vs. 92.51–97.28%; Climbing up: 77.88% vs. 92.47–96.11%), indicating that the gross statistical summaries it relies on cannot capture the structured, repeating motion these activities require.

The largest gains of SAA-SAX over both symbolic baselines occur for sitting and standing (improvements of roughly 9–12 points over BOP+MII, and 11–12 points over BOP), and for lying (an improvement of about 4.7 points over BOP+MII). These three are precisely the activities most often flagged in the HAR literature as difficult to separate from one another, since they involve little motion and are primarily distinguished by posture-dependent gravitational components on the accelerometer axes rather than by dynamic movement patterns. BOP+MII's gains over plain BOP, by contrast, were concentrated in activities with a distinctive joint movement signature *across* axes (e.g., jumping, climbing), consistent with its cross-channel design. SAA's gains, deriving from *within-axis* trend information, remain informative even for postures with minimal movement, since the orientation of each axis relative to gravity still produces a consistent local slope.

4.4 Discussion

The joint evaluation results in Section 4.3.4 point to a consistent pattern: each proposed extension improves accuracy through a mechanism specific to its design, rather than through a generic strengthening of the symbolic representation. BOP+MII's gains over plain BOP concentrate in activities with a distinctive joint movement signature across axes – jumping and the stair-climbing classes – consistent with its cross-channel construction. SAA's gains instead concentrate in the activities BOP and BOP+MII struggle with most, sitting, standing, and lying, where little gross motion is present and class identity depends on a stable within-axis orientation relative to gravity rather than on coordinated movement across channels. This division is informative beyond the specific numbers: it suggests the two extensions are answering genuinely different weaknesses identified in Chapter 3's taxonomy (multivariate support and trend loss, respectively), rather than two routes to the same improvement – though, whether combining them would yield a further gain remains untested here, since MII and SAA were evaluated as independent extensions of BOP rather than jointly.

This evaluation is also narrower in scope than later chapters in this thesis. Both methods are assessed on a single dataset with a single, fixed parameter configuration (Table 4.1), evaluated with a simple 1-Nearest Neighbour classifier rather than the Random Forest used from Chapter 5 onward, and the SAA evaluation deliberately reuses MII's parameter values rather than being tuned independently. None of these choices were made to maximize either method's reported accuracy; they were made to keep the comparison controlled and attributable to the representational change under test.

4.5 Synthesis

These results establish MII as a self-contained, low-cost extension of the SAX representation to the multivariate case: it adds a single α^p-dimensional feature derived entirely from quantities already computed during the standard SAX pipeline, requires no additional parameters beyond those already needed for per-channel SAX, and yields a consistent improvement in both accuracy and stability over the single-channel baseline across all eight activities considered. In doing so, it closes the multivariate-support gap identified in Chapter 3 without departing from the SAX alphabet or introducing a separate cross-channel preprocessing stage.

By computing trend information through a parallel, independently discretized branch rather than a symbol fused directly into the mean-value representation, SAA closes the trend-preservation gap identified in Chapter 3 while keeping each of its two constituent representations structurally identical in form to standard SAX, and allowing the two to be combined or analyzed independently at the feature level.

This page has been left blank intentionally.

Chapter 5

Optimized SAX for Multivariate Time Series Classification

Chapter 3's taxonomy identified parameter selection as the second-largest category of SAX weaknesses, only behind lossy compression, with the two together accounting for over 60% of the reviewed corpus (Section 3.4). Among the methods surveyed there, SAX-VSM (Section 3.3.1) already showed that the DIRECT global optimization algorithm can remove the need for manual SAX tuning, but its use of DIRECT is entangled with class-discriminative term weighting and confined to the univariate case. Chapter 4 then introduced the MII (Section 4.1) as a lightweight extension of SAX to multivariate time series, but evaluated it on a single HAR dataset with parameters fixed by hand rather than tuned. This chapter closes both gaps together: it develops a deterministic, dataset-adaptive framework that applies DIRECT directly to the SAX transformation parameters for MTSC, and revisits MII as one of several feature-extraction strategies subjected to this tuning across a substantially larger and more heterogeneous benchmark than its original single-dataset evaluation. The work presented in this chapter is based on [168].

Chapter 3's discussion of SAX's univariate design (Section 3.3.1) illustrates one strand of this space through MBOP, SAX-ARM, and the SAX-PCA/SAX-REPEAT /SAX-ZSCORE family of Mohammad and Nishida – each extending SAX beyond its single-channel origins. The broader SAX literature includes several further methods designed specifically for MTSC that illustrate just how varied the mechanisms for incorporating cross-channel information can be. MultiSAX [169] layers inter-variable, spatial, and temporal correlations onto the symbolic representation to support concept-level abstraction in sensor data, while MSAX [170] normalizes the multivariate series via its covariance matrix prior to discretization, preserving cross-channel structure at the cost of added complexity. WSAX [171] combines both perspectives in a two-phase pipeline – wavelet-based correlation clustering followed by SAX-based shape modeling – to separate series that share similar statistics but differ in morphology. More recent contributions model statistical dependencies among variables directly: Jha and Tripathi [172] fit joint distribution functions across channels before applying SAX, SAXRegEx [173] forms intertwined symbolic sequences that support regular-expression-based querying across asynchronous channels, and MultiCast [174] uses dimensional multiplexing and SAX quantization to convert MTS into compact symbolic forms suitable for large language models. Common to all of these methods – and to the variants discussed in Chapter 3 – is that none addresses principled, automated selection of the SAX transformation parameters themselves; each focuses on how

This chapter is based on the following publication: [J2].

cross-channel information is incorporated, not on how the underlying parameters should be chosen for a given dataset. That gap is the one this chapter targets.

The same gap persists one level up, among MTSC methods that are not SAX-based at all. As discussed in Section 2.4.2, even the strongest dictionary-based MTSC methods – WEASEL 2.0, SCALE-BOSS-MR, and BORF – rely on heuristic parameter choices, grid search, or a single fixed configuration applied uniformly across datasets, sacrificing exactly the reproducibility and dataset-adaptivity that a principled tuning procedure would provide. Motivated by this gap, and by the broader finding that well-tuned lightweight models can rival far more complex architectures [81], this chapter proposes a deterministic framework that applies DIRECT to globally optimize SAX's transformation parameters – the PAA segment-size fraction, the alphabet size, and the word length – either as a single configuration shared across all channels of a multivariate series or as independent configurations per channel, and combines this tuning with two feature-extraction choices: per-channel BOP histograms alone, or BOP histograms augmented with the MII. The resulting four configurations, Methods A through D, allow the effects of tuning granularity and multichannel feature inclusion to be examined independently.

The remainder of the chapter is organized as follows. Section 5.1 details the DIRECT-based tuning procedure and the four proposed methods. Section 5.2 describes the datasets, evaluation protocol, and baselines used to assess them. Section 5.3 reports accuracy and runtime results across 24 datasets from the UEA MTSC archive, together with an interpretability case study. Section 5.4 interprets these findings, and the chapter closes by relating them back to the thesis's broader narrative.

5.1 Proposed Methodology

Figure 5.1 summarizes the overall pipeline developed in this chapter: for each dataset, DIRECT performs a dataset-level search over the SAX parameters; the selected configuration is then fixed and used to extract features for both training and test-time inference.

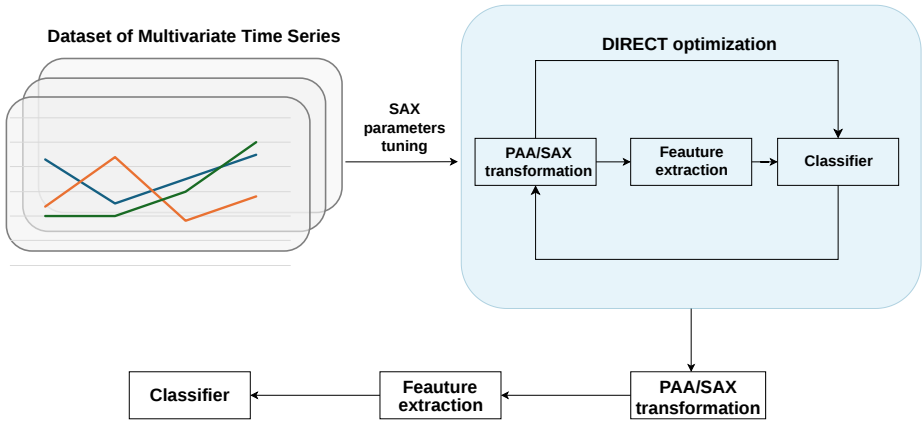


Figure 5.1: Overview of the proposed pipeline. For each multivariate time series dataset, DIRECT performs a dataset-level search to select the SAX transformation parameters. The selected configuration is then fixed and used for PAA/SAX-based feature extraction, after which a classifier is trained and applied at test time.

5.1.1 DIRECT-Based Parameter Optimization

SAX's three transformation parameters – the PAA segment size, the alphabet size, and the word length (Sections 2.2.1, 2.2.2, 2.2.3) – jointly control the trade-off between simplification and representational detail. Choosing them well is what allows the resulting symbolic sequence to retain the class-discriminative structure of the original signal; choosing them poorly is, as Chapter 3 documented, the second most common failure mode across the SAX literature. To select these parameters automatically and adapt them to each dataset, this chapter employs the DIRECT (DIviding RECTangles) algorithm [175], a deterministic, derivative-free global optimization method already shown to be effective for tuning SAX in the univariate case by SAX-VSM (Section 3.3.1). DIRECT requires only a bounded search space and no gradient information, which suits it to optimizing a non-smooth, black-box objective such as classification accuracy, and makes it a more principled alternative to manual tuning or exhaustive grid search – particularly given how much parameter sensitivity varies across datasets of different length, sampling rate, and dimensionality. Building on this precedent, the present chapter extends DIRECT's use beyond SAX-VSM's univariate, weighting-entangled setting to the multichannel case, tuning the PAA segment size, the alphabet size, and the word length either as a single configuration shared across all channels or as independent configurations per channel, depending on the method (Section 5.1.2).

For PAA in particular, the number of segments m is not tuned as an absolute count but as a segment-size fraction $\rho \in (0, 1]$ of the series length n, so that a single tuned value adapts automatically to datasets and channels of very different lengths rather than requiring a separate search range for each one.

Figure 5.2 illustrates how DIRECT operates over a three-dimensional search domain: starting from the whole domain as a single hyperrectangle, the algorithm repeatedly trisects selected rectangles into equal thirds along their longest dimension, evaluates the objective at the center of each resulting rectangle, and selects for further subdivision those rectangles that are simultaneously promising – having a low function value at their center – and still large enough to contain unexplored volume. This dual criterion balances global exploration of the parameter space against local refinement around the best regions found so far, letting the search concentrate computation on promising areas without the combinatorial cost of an exhaustive grid.

Wrapping this search procedure around the symbolic pipeline – evaluating, for each candidate parameter set, the resulting representation's cross-validated classification accuracy – yields a fully automated, dataset-adaptive tuning step that requires no manual intervention while remaining far cheaper than exhaustive search. The concrete search ranges, the regularized objective used to break ties between equally accurate configurations, and the optimization budget are specified in Section 5.2.

5.1.2 Four Proposed Feature-Extraction Strategies

Building on this tuning procedure, four feature-extraction strategies are proposed and compared. Each pairs SAX-based discretization with BOP histogram construction (Section 2.2.3) and differs along two independent axes: (i) whether the SAX parameters selected by DIRECT are shared across all channels of a multivariate series or tuned independently per channel, and (ii) whether the per-channel BOP histograms are used alone or augmented with the MII (Section 4.1). The resulting four configurations – Methods A through D – are detailed below; each is evaluated with the same downstream classifier, a Random Forest with 100 trees, so that differences in performance can be

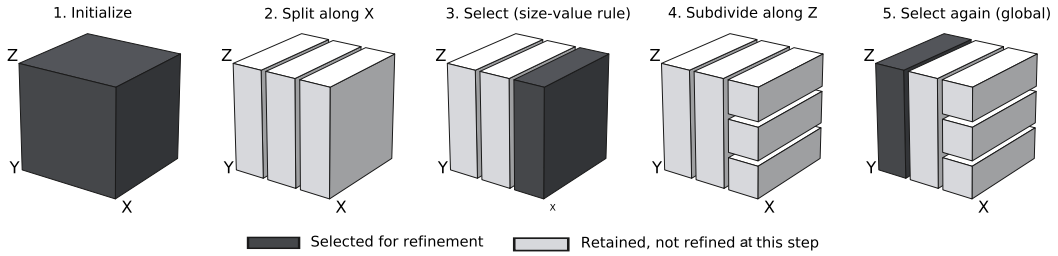


Figure 5.2: Step-by-step illustration of the DIRECT algorithm over a 3D optimization domain. (1) The whole domain starts as a single hyper-rectangle. (2) It is split into thirds along its longest dimension (X). (3) Based on the size-versus-value selection rule, one rectangle is chosen for refinement (dark); the other two are retained, not discarded. (4) The selected rectangle is subdivided along its own longest dimension while the unselected rectangles from step 3 remain untouched. (5) At the next iteration, a large, still-unrefined rectangle (left) is selected again, even though smaller, more recently refined rectangles exist elsewhere: global exploration continues alongside local refinement, rather than concentrating permanently on whichever region looked best first.

attributed to the feature-extraction strategy rather than to the classifier.

Method A: Shared Tuning with Per-Channel BOP

In Method A, each channel of the multivariate series is transformed independently via SAX, and a BOP histogram is constructed per channel; the resulting per-channel histograms are concatenated into the final feature vector. A single configuration – the PAA segment-size fraction ρ, the alphabet size a, and the word length w – is tuned once per dataset via DIRECT (Section 5.1.1) and applied uniformly across all channels. Figure 5.3 illustrates the pipeline.

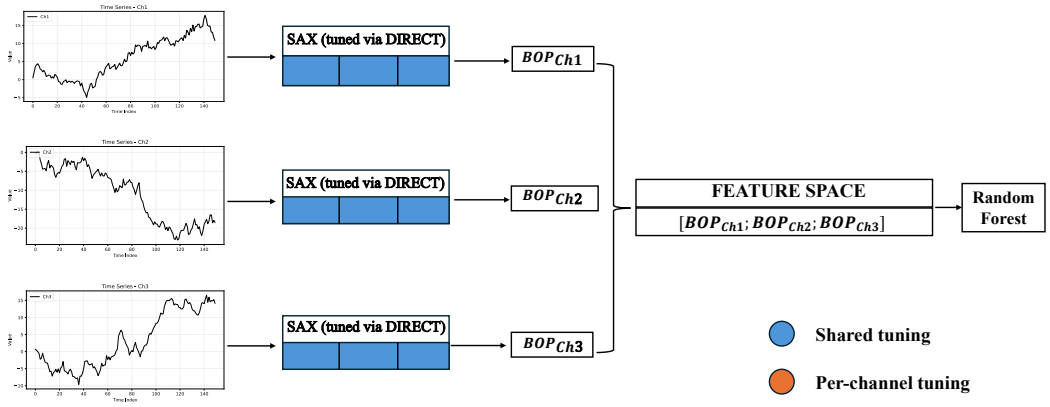


Figure 5.3: Implementation of Method A. Each channel is transformed using SAX under a single set of parameters, tuned once on the training split via DIRECT and then applied uniformly across all channels.

Method B: Shared Tuning with Per-Channel BOP and MII

Method B extends Method A by augmenting the feature space with the MII (Section 4.1): the same shared, DIRECT-tuned configuration is used to construct both the per-channel BOP histograms and the multichannel words underlying the MII, and the two are concatenated into a single feature vector. Because it reuses Method A's tuned parameters rather than running a separate search, Method B isolates the effect of adding cross-channel symbolic information on top of an already-optimized per-channel representation. Figure 5.4 illustrates the pipeline.

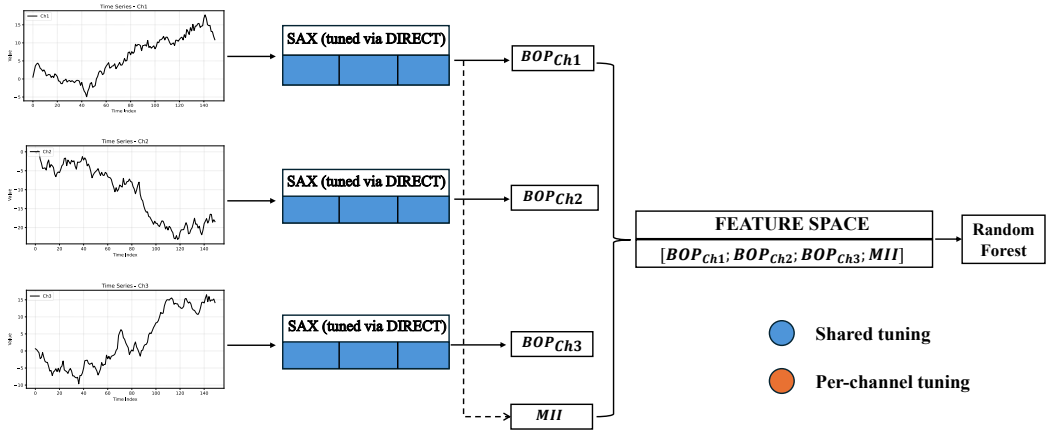


Figure 5.4: Implementation of Method B. As in Method A, each channel is transformed using a single shared, DIRECT-tuned configuration; the MII feature is additionally concatenated to the per-channel BOP histograms.

Method C: Per-Channel Tuning with Per-Channel BOP

Method C instead lets the SAX configuration vary across channels: each channel is independently SAX-transformed and converted into a BOP histogram as in Method A, but ρ, a, and w are each tuned separately, per channel, via DIRECT. The resulting per-channel histograms – built under possibly different parameter choices – are concatenated as before, producing a representation tailored to each dimension's own characteristics, at the cost of a substantially larger search space than Method A. Figure 5.5 illustrates the pipeline.

Method D: Per-Channel Tuning with Per-Channel BOP and MII

Method D combines Method C's per-channel tuning with the multichannel feature of Method B. Each channel's alphabet size and word length are tuned independently, but the PAA segment-size fraction ρ is kept shared across channels so that the temporal alignment required to construct multichannel words is preserved. The resulting feature vector concatenates the per-channel BOP histograms – each under its own tuned alphabet and word length – with the MII built from the shared PAA segmentation, combining the flexibility of per-channel tuning with the cross-channel structure captured by MII. Figure 5.6 illustrates the pipeline.

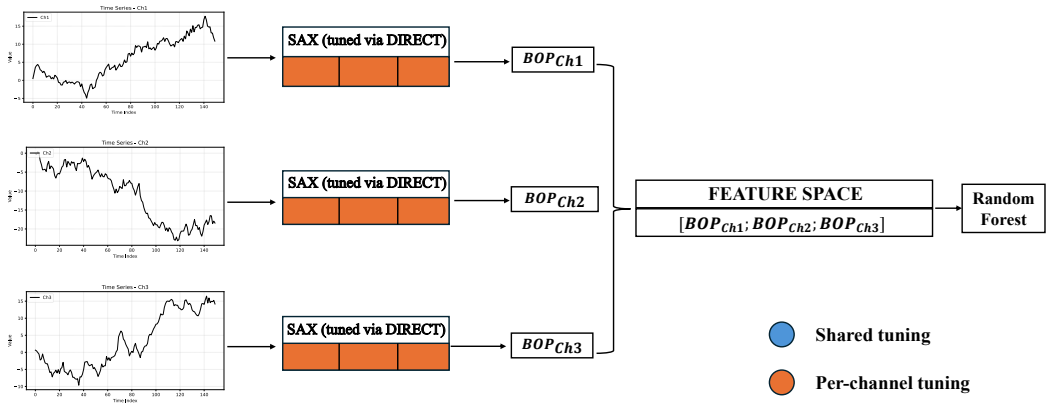


Figure 5.5: Implementation of Method C. Each channel is transformed using its own SAX configuration, tuned independently via DIRECT.

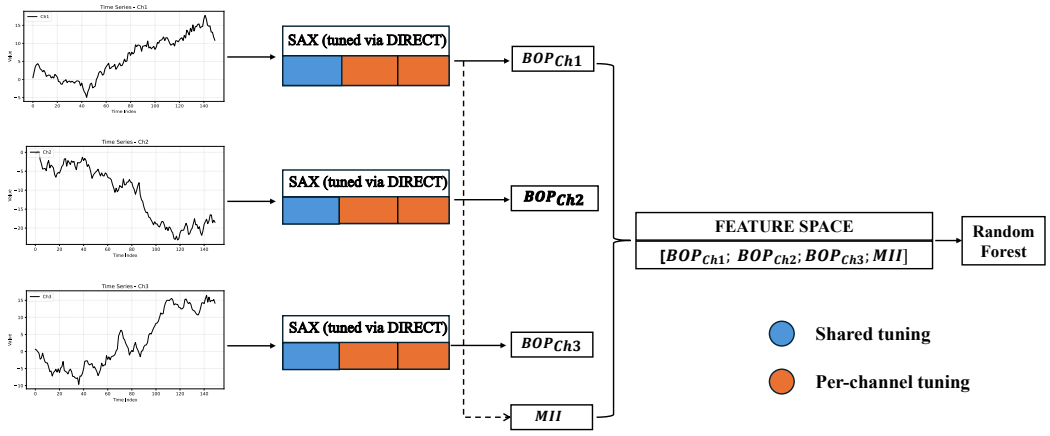


Figure 5.6: Implementation of Method D. Each channel's alphabet size and word length are tuned independently via DIRECT, while the PAA segment-size fraction is kept shared to preserve cross-channel alignment; the MII is concatenated to the resulting feature set.

Each of these four configurations is evaluated under the same Random Forest classifier, isolating the impact of tuning granularity and multichannel feature inclusion on classification performance – examined empirically in Section 5.3.

5.2 Experiments and Evaluation Method

This section describes the experimental setup used to evaluate Methods A–D: the benchmark datasets (Section 5.2.1) and the evaluation protocol covering metrics, classifier, parameter-tuning strategy, and baseline comparisons (Section 5.2.2).

5.2.1 Datasets

Methods A–D are evaluated on 24 datasets (Table 5.1) drawn from the UEA MTSC archive (Section 2.3), which provides predefined train/test splits to support standardized and reproducible comparison. Of the archive's 30 datasets, six were excluded for practical reasons detailed in the table notes; the remaining 24 span a broad range of problem characteristics – number of instances, series length, dimensionality, and number of classes – across a variety of application domains, providing a diverse testbed for both predictive performance and runtime behavior.

Table 5.1: Summary of the 24 multivariate time series datasets used for evaluation, drawn from the UEA Multivariate Time Series Classification archive.

Dataset	Train	Test	Dimensions	Length	Classes
ArticularyWordRecognition	275	300	9	144	25
AtrialFibrillation	15	15	2	640	3
BasicMotions	40	40	6	100	4
CharacterTrajectories	1422	1436	3	182	20
Cricket	108	72	6	1197	12
EigenWorms	128	131	6	17984	5
Epilepsy	137	138	3	206	4
ERing	30	30	4	65	6
EthanolConcentration	261	263	3	1751	4
FingerMovements	316	100	28	50	2
HandMovementDirection	320	147	10	400	4
Handwriting	150	850	3	152	26
Heartbeat	204	205	61	405	2
JapaneseVowels	270	370	12	29	9
Libras	180	180	2	45	15
LSST	2459	2466	6	36	14
MotorImagery	278	100	64	3000	2
NATOPS	180	180	24	51	6
RacketSports	151	152	6	30	4
SelfRegulationSCP1	268	293	6	896	2
SelfRegulationSCP2	200	180	7	1152	2
SpokenArabicDigits	6599	2199	13	93	10
StandWalkJump	12	15	4	2500	3
UWaveGestureLibrary	120	320	3	315	8

Note: Six UEA archive datasets were excluded from the evaluation. DuckDuckGeese, FaceDetection, InsectWingbeat, and PEMS-SF were excluded due to excessive computational requirements; PenDigits was omitted because its series length was insufficient for the SAX transformation; and Phoneme was excluded due to a label inconsistency in the distributed archive (only one class present).

5.2.2 Evaluation Protocol

Evaluation Metrics and Classifier

Classification accuracy on the held-out test split is the primary evaluation metric throughout; wall-clock runtime is additionally reported, detailed in Section 5.3, to characterize the computational cost of each method. All four proposed methods use a Random Forest classifier [176] with 100 trees; node splitting uses the Gini impurity criterion with best-split selection, and trees are trained with bootstrap sampling.

Parameter Tuning Strategy

For every dataset, DIRECT (Section 5.1.1) tunes one set of SAX parameters using only the training split; the selected configuration is then fixed for all subsequent evaluation. Since DIRECT is a minimization algorithm while the underlying goal is to maximize classification accuracy, the parameters are tuned via 5-fold cross-validation on the training split, with the objective defined as the negative of the mean cross-validated accuracy. The search space explored by DIRECT is given in Table 5.2.

Table 5.2: Optimization search space for each SAX parameter tuned via DIRECT.

Parameter	Notation	Range	Justification
Alphabet size	a	2 to 8	Prior work on SAX [54] shows that sizes 5–8 balance discretization granularity against computational efficiency; the range is extended down to 2 to additionally explore coarser granularities.
Word length	w	2 to 6	Empirical studies across diverse time series [54, 87] show that word lengths in this range capture temporal patterns effectively while remaining computationally tractable.
PAA segment-size fraction	ρ	Dataset-specific: fraction of the series length	Tuning a fraction rather than an absolute segment count lets a single configuration adapt to channels and datasets of widely different lengths.

In preliminary experiments, multiple parameter combinations were frequently found to achieve identical cross-validated accuracy. To discourage unnecessarily large or complex configurations among these ties – which inflate runtime and feature dimensionality without improving accuracy – the optimization objective is augmented with a mild regularization term. Given accuracy $\mathrm{acc} \in [0, 1]$ and the tuned parameter set $\{\rho, a, w\}$, the loss minimized by DIRECT is

$$L = -\,\mathrm{acc} + \varepsilon\,(\rho + a + w) \tag{5.1}$$

where ε controls the strength of the penalty. Across all experiments, $\varepsilon = 10^{-3}$.

DIRECT is allowed at most 500 objective evaluations per dataset and terminates early after 50 consecutive evaluations with no improvement, where an improvement is an absolute increase of at least 10^{-3} in validation accuracy. In practice, DIRECT often reaches a performance plateau well before this patience limit is reached; the limit is nonetheless set conservatively, together with a lenient 10^{-3} improvement threshold, to reduce the risk of stopping prematurely due to the inherent stochasticity of the cross-validated accuracy estimate and to allow sufficient exploration of the search space before termination.

Evaluation Strategy

Methods A–D are compared against six baselines spanning the major MTSC families already surveyed in Section 2.4.2: the elastic distance-based classifier 1NN–DTW_D, the ROCKET-family method MultiROCKET, the deep learning model InceptionTime, and the dictionary-based classifiers WEASEL+MUSE, BORF, and SAX-VSM (Section 3.3.1). All baselines are evaluated under the same protocol and hardware as the proposed methods. The key components of the experimental setup are summarized in Table 5.3.

Table 5.3: Summary of the experimental setup: evaluation metrics, classifier, parameter tuning, optimization budget and stopping criteria, validation and testing protocol, baseline comparisons, runtime measurement, and statistical analysis.

Component	**Specification**
Metrics	Accuracy (test); Runtime (wall-clock)
Classifier	Random Forest (100 trees)
Optimization method	DIRECT global optimization
Tuned parameters	PAA segment-size fraction ρ; alphabet size a; word length w
Parameter ranges	$a \in [2, 8]$; $w \in [2, 6]$; ρ: dataset-specific
Optimization objective	Minimize $L = -\text{acc} + \varepsilon(\rho + a + w)$
Budget and stopping criteria	Max. evaluations = 500; early-stopping patience = 50 consecutive non-improving evaluations; minimum improvement $\Delta \geq 10^{-3}$ (absolute accuracy)
Validation	5-fold cross-validation on the training split (used for parameter selection)
Final evaluation protocol	Refit on the full training split using the selected parameters; evaluate once on the held-out test split; report test accuracy
Baseline methods	1NN–DTW_D; MultiROCKET; InceptionTime; WEASEL+MUSE; BORF; SAX-VSM
Runtime measurement	Wall-clock runtime under the same hardware/software environment for all methods
Statistical analysis	Average ranks across datasets; Friedman test with Nemenyi post-hoc; critical difference (CD) diagram at $\alpha = 0.05$

5.3 Results

This section reports the experimental outcomes for Methods A–D under the protocol of Section 5.2: classification accuracy on the UEA benchmarks (Section 5.3.1), runtime (Section 5.3.2), and an interpretability case study (Section 5.3.3).

5.3.1 Accuracy on the UEA Benchmarks

Table 5.4 reports per-dataset test accuracy for Methods A–D and the six baselines, together with mean accuracy, win/tie/loss counts, and average rank. Methods B and D failed to complete on several high-dimensional datasets due to memory exhaustion; for the rank-based summaries, these non-completing runs are conservatively assigned the worst rank on the corresponding dataset, so that any rank advantage retained by B or D reflects genuine predictive performance rather than lenient handling of missing data.

Globally, MultiROCKET achieves both the highest mean accuracy (78.11%) and the leading average rank (2.33), followed by WEASEL+MUSE (3.23). BORF is the strongest symbolic baseline, ranking third overall (4.42, 72.57% mean accuracy), while SAX-VSM ranks last among the baselines (7.71, 45.87%), though it remains competitive on a narrow set of physiological datasets such as Heartbeat and FingerMovements. Among the proposed variants, Method B is the most robust (average rank 6.13), closely followed by Method A (6.19).

A closer, per-dataset look reveals specific strengths in physiological signal domains. Method A reaches 53.33% on AtrialFibrillation, well ahead of every baseline, including InceptionTime (33.30%), BORF (20.00%), and SAX-VSM (40.00%). Method B reaches a benchmark-best 63.48% on SelfRegulationSCP2, more than 7 points above the nearest baseline. BORF, for its part, takes the top rank on ArticularyWordRecognition (99.67%) and places in the top three on nine of the 24 datasets, confirming its strength as a general-purpose symbolic baseline. These results suggest that while the proposed methods do not match the overall generalization of ROCKET-family approaches, they remain a competitive, interpretable alternative on datasets where high-capacity models struggle to extract the relevant temporal structure.

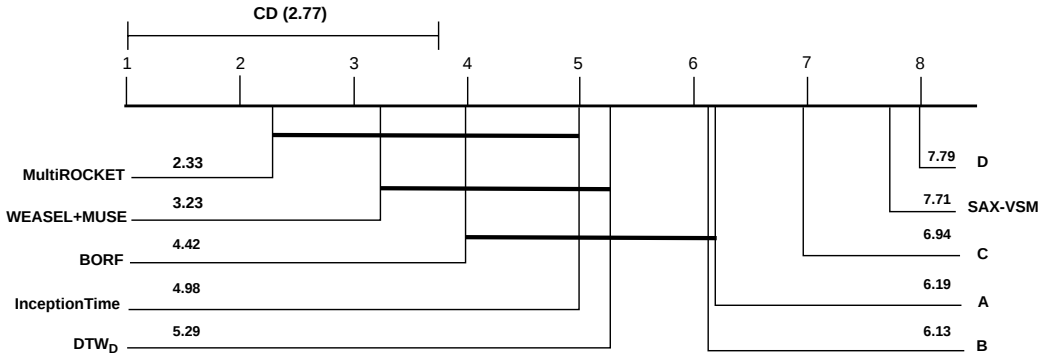


Figure 5.7: Critical Difference (CD) diagram comparing all ten methods on the full 24-dataset evaluation. Markers show each method's average rank (lower is better); methods joined by a horizontal bar form a non-significant clique under the Friedman test with Nemenyi post-hoc at $\alpha = 0.05$. Non-completing runs are assigned the worst rank on the corresponding dataset before averaging.

Table 5.4: Classification accuracy (%) on the standard UEA MTSC train/test splits for Methods A–D against six baselines (1NN–DTW_D, MultiROCKET, InceptionTime, WEASEL+MUSE, BORF, SAX-VSM). Best result per dataset in bold; "–" denotes non-completing runs (assigned worst rank in rank-based summaries). Summary rows report mean accuracy, Win/Tie/Lose, and average rank; CD is the critical difference of average ranks at $\alpha = 0.05$ (Friedman test with Nemenyi post-hoc, Demšar procedure).

	Proposed Methods				**Comparative Methods**					
Dataset	**A**	**B**	**C**	**D**	**DTW_D**	**MultiROCKET**	**Inception Time**	**WEASEL +MUSE**	**BORF**	**SAX-VSM**
ArticularyWordRecognition	95.00%	93.33%	86.00%	-	98.67%	99.33%	98.67%	99.33%	**99.67%**	41.67%
AtrialFibrillation	**53.33%**	33.33%	46.67%	26.67%	13.33%	6.67%	33.30%	13.33%	20.00%	40.00%
BasicMotions	92.50%	92.50%	**100.00%**	95.00%	**100.00%**	**100.00%**	52.50%	**100.00%**	**100.00%**	82.50%
CharacterTrajectories	86.70%	89.90%	80.92%	94.15%	97.70%	99.37%	**99.72%**	98.96%	96.67%	21.45%
Cricket	97.22%	98.61%	91.67%	95.83%	**100.00%**	98.61%	98.61%	**100.00%**	98.61%	16.67%
EigenWorms	87.79%	88.55%	88.55%	87.79%	59.54%	**96.95%**	53.44%	95.42%	91.60%	25.19%
Epilepsy	96.38%	97.10%	95.65%	96.38%	92.75%	**100.00%**	97.83%	**100.00%**	**100.00%**	94.20%
EthanolConcentration	31.18%	28.52%	23.19%	26.62%	29.66%	**55.51%**	28.90%	54.37%	53.61%	33.46%
ERing	65.56%	75.93%	63.70%	72.96%	94.44%	**99.63%**	84.81%	95.56%	90.74%	72.59%
FingerMovements	52.00%	-	56.00%	-	48.00%	**63.00%**	54.00%	54.00%	54.00%	61.00%
HandMovementDirection	24.32%	28.38%	32.43%	-	27.03%	33.78%	**40.54%**	31.08%	25.68%	20.27%
Handwriting	17.41%	19.18%	14.94%	16.82%	49.65%	52.59%	**56.47%**	34.94%	51.18%	30.35%
Heartbeat	72.20%	-	72.20%	-	66.34%	**76.59%**	44.80%	73.66%	72.68%	74.15%
JapaneseVowels	56.49%	59.19%	33.78%	-	95.95%	**98.38%**	72.97%	93.24%	75.68%	25.68%
Libras	66.11%	66.67%	40.00%	68.89%	87.78%	**93.33%**	6.67%	92.22%	83.33%	32.22%
LSST	57.06%	57.06%	49.88%	46.92%	56.85%	**63.75%**	25.79%	60.91%	54.62%	15.61%
MotorImagery	**55.00%**	-	51.00%	-	43.00%	49.00%	**55.00%**	53.00%	50.00%	**55.00%**
NATOPS	82.78%	-	75.00%	-	87.22%	90.56%	**94.44%**	93.33%	91.11%	73.89%
RacketSports	68.42%	71.05%	57.24%	69.74%	84.21%	**90.13%**	83.55%	86.84%	86.84%	58.55%
SelfRegulationSCP1	70.99%	72.70%	70.99%	74.40%	78.50%	**94.88%**	78.16%	78.16%	72.35%	45.39%
SelfRegulationSCP2	52.22%	**63.48%**	51.67%	50.56%	50.56%	53.33%	52.78%	56.11%	42.22%	50.00%
SpokenArabicDigits	88.54%	88.40%	86.58%	-	97.64%	98.91%	**99.23%**	98.86%	98.18%	71.53%
StandWalkJump	46.67%	53.33%	40.00%	46.67%	33.33%	**66.67%**	33.33%	40.00%	40.00%	33.33%
UWaveGestureLibrary	79.37%	81.25%	72.81%	48.13%	91.56%	**93.75%**	87.81%	90.62%	92.81%	26.25%
AVERAGE	66.47%	67.92%	61.70%	63.60%	70.15%	**78.11%**	63.89%	74.75%	72.57%	45.87%
Win/Tie/Lose	1/1/22	1/0/23	0/1/23	0/0/24	0/2/22	**12/2/10**	5/1/18	0/3/21	1/2/21	1/0/23
Average Rank	6.19	6.13	6.94	7.79	5.29	**2.33**	4.98	3.23	4.42	7.71
CD						2.77				

Figure 5.7 summarizes these ranks via a Friedman test with Nemenyi post-hoc analysis at $alpha = 0.05$ (Critical Difference — CD = 2.77), following the Demšar [177] procedure. MultiROCKET's rank advantage is statistically significant over DTW_D, all four proposed variants, and SAX-VSM, but its gap to WEASEL+MUSE, BORF, and InceptionTime does not exceed the CD threshold and is therefore not significant. Among the proposed methods, Method B offers the best average rank, but its failures on several high-dimensional datasets limit its general applicability; Methods A and C, by contrast, complete all 24 datasets without resource-related failure. Of the two, Method A combines this stability with stronger predictive power, beating Method C on both mean accuracy (66.47% vs. 61.70%) and average rank (6.19 vs. 6.94).

Because the Friedman/Nemenyi analysis assumes a complete block design, the missing entries from Methods B and D's incomplete runs warrant a secondary, complete-case verification restricted to the 16 datasets on which all ten methods completed successfully. Table 5.5 and Figure 5.8 report this recomputed comparison.

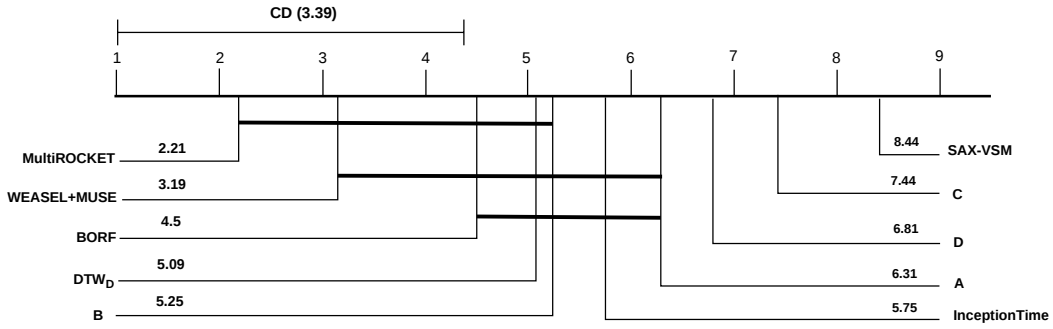


Figure 5.8: Critical Difference (CD) diagram on the common subset of 16 datasets where all methods completed successfully. Markers show each method's average rank across these datasets; methods joined by a horizontal bar form a non-significant clique under the Friedman test with Nemenyi post-hoc at $\alpha = 0.05$.

Removing the failure-rank penalty shifts the picture somewhat. MultiROCKET remains the strongest performer (79.07% mean accuracy, rank 2.21), followed by WEASEL+MUSE (3.19) and now BORF in third place (4.50, 73.41%), displacing InceptionTime from the upper tier. DTW_D (5.09) and Method B (5.25) follow closely, with InceptionTime sixth (5.75). Method B is the most competitive of the proposed variants here, and removing the penalty reveals its true standing relative to InceptionTime: where the full 24-dataset evaluation ranked it behind InceptionTime (6.13 vs. 4.98) purely because of its incomplete runs, on the common subset it overtakes InceptionTime in rank (5.25 vs. 5.75) and exceeds it by more than 7 points in mean accuracy (68.07% vs. 60.85%) – confirming that InceptionTime's earlier advantage was an artifact of Method B's missing entries rather than a genuine performance gap. SAX-VSM ranks last on this subset (8.44, 42.36%), with its only competitive result being AtrialFibrillation (40.00%), still well below Method A's 53.33% there. The domain-specific strengths persist on this subset too: Method A retains its lead on AtrialFibrillation, Method B its best-in-class result on SelfRegulationSCP2, and Method C reaches a perfect 100% on BasicMotions, matching the strongest baselines. With CD = 3.39, Method B's rank of 5.25 is statistically indistinguishable from MultiROCKET, WEASEL+MUSE, BORF, and DTW_D on this subset. Method A's stability is notable here too: even restricted to the subset where the more memory-intensive Method B succeeds, Method A still achieves 66.81% mean accuracy,

Table 5.5: Classification accuracy (%) on the common subset of 16 UEA datasets where all ten methods completed successfully. The best result per dataset is bold. Summary rows report mean accuracy, Win/Tie/Lose counts, average rank, and the critical difference (CD) of average ranks (Friedman test with Nemenyi post-hoc) at $\alpha = 0.05$.

	Proposed Methods				**Comparative Methods**					
Dataset	**A**	**B**	**C**	**D**	**DTW$_D$**	**MultiROCKET**	**Inception Time**	**WEASEL +MUSE**	**BORF**	**SAX-VSM**
AtrialFibrillation	**53.33%**	33.33%	46.67%	26.67%	13.33%	6.67%	33.30%	13.33%	20.00%	40.00%
BasicMotions	92.50%	92.50%	**100.00%**	95.00%	**100.00%**	**100.00%**	52.50%	**100.00%**	**100.00%**	82.50%
CharacterTrajectories	86.70%	89.90%	80.92%	94.15%	97.70%	99.37%	**99.72%**	98.96%	96.67%	21.45%
Cricket	97.22%	98.61%	91.67%	95.83%	**100.00%**	98.61%	98.61%	**100.00%**	98.61%	16.67%
EigenWorms	87.79%	88.55%	88.55%	87.79%	59.54%	**96.95%**	53.44%	95.42%	91.60%	25.19%
Epilepsy	96.38%	97.10%	95.65%	96.38%	92.75%	**100.00%**	97.83%	**100.00%**	**100.00%**	94.20%
EthanolConcentration	31.18%	28.52%	23.19%	26.62%	29.66%	**55.51%**	28.90%	54.37%	53.61%	33.46%
ERing	65.56%	75.93%	63.70%	72.96%	94.44%	**99.63%**	84.81%	95.56%	90.74%	72.59%
Handwriting	17.41%	19.18%	14.94%	16.82%	49.65%	52.59%	**56.47%**	34.94%	51.18%	30.35%
Libras	66.11%	66.67%	40.00%	68.89%	87.78%	**93.33%**	6.67%	92.22%	83.33%	32.22%
LSST	57.06%	57.06%	49.88%	46.92%	56.85%	**63.75%**	25.79%	60.91%	54.62%	15.61%
RacketSports	68.42%	71.05%	57.24%	69.74%	84.21%	**90.13%**	83.55%	86.84%	86.84%	58.55%
SelfRegulationSCP1	70.99%	72.70%	70.99%	74.40%	78.50%	**94.88%**	78.16%	78.16%	72.35%	45.39%
SelfRegulationSCP2	52.22%	**63.48%**	51.67%	50.56%	50.56%	53.33%	52.78%	56.11%	42.22%	50.00%
StandWalkJump	46.67%	53.33%	40.00%	46.67%	33.33%	**66.67%**	33.33%	40.00%	40.00%	33.33%
UWaveGestureLibrary	79.37%	81.25%	72.81%	48.13%	91.56%	**93.75%**	87.81%	90.62%	92.81%	26.25%
AVERAGE	66.81%	68.07%	61.74%	63.60%	69.99%	**79.07%**	60.85%	74.84%	73.41%	42.36%
Win/Tie/Lose	1/0/15	1/0/15	0/1/15	0/0/16	0/2/14	**9/2/5**	2/0/14	0/3/13	0/2/14	0/0/16
Average Rank	6.31	5.25	7.44	6.81	5.09	**2.21**	5.75	3.19	4.50	8.44
CD						3.39				

confirming it as a genuinely competitive configuration rather than merely a stable fallback.

Although Methods C and D are nominally more flexible than A and B, since they tune SAX parameters independently per channel, this flexibility does not translate into better generalization in this benchmark. Independent per-channel tuning substantially enlarges the search space; under a fixed DIRECT evaluation budget and early-stopping policy, this makes optimization harder and increases the variance of the resulting model selection. Method C attains a lower mean validation accuracy than Method A (65.74% vs. 69.28%), so the larger per-channel search did not even find a better training-time configuration, while Method D shows a sizeable validation–test gap (70.23% vs. 63.60%) consistent with higher selection variance when more parameters are tuned. Method A – and Method B, which inherits A's tuned parameters – instead enforces a single shared configuration across channels, which acts as an implicit regularizer: every channel is discretized at the same temporal and amplitude resolution, producing more homogeneous, aligned BOP histograms that the Random Forest classifier can exploit for cross-channel regularities under a consistent symbolic partition. In this benchmark, the resulting regularization and alignment benefit outweighs the expressive advantage of per-channel tuning.

5.3.2 Runtime Analysis

Computational efficiency is the second axis of evaluation, particularly relevant for real-time or resource-constrained deployment. Runtime is reported on the common 16-dataset subset where every method completed successfully: unlike the accuracy analysis, where a conservative worst-rank penalty is a meaningful substitute for a non-completing run, runtime has no analogous well-defined value to impute, so including incomplete runs would bias any summary.

In this framework, DIRECT's computational cost is evaluation-driven: the overhead of partitioning the search domain is negligible next to the cost of evaluating a single candidate parameter set. Each evaluation performs (i) SAX feature extraction on the training split and (ii) k-fold cross-validation of the classifier, so the overall tuning cost scales approximately as

$$O\big(N_{\text{eval}} \cdot k \cdot \big(C_{\text{feat}} + C_{\text{clf}}\big)\big)\,, \tag{5.2}$$

where N_{eval} is the number of DIRECT evaluations, k the number of cross-validation folds, C_{feat} the cost of SAX/PAA plus histogram construction, and C_{clf} the per-fold classifier training/evaluation cost. For N series of length T with p channels, feature extraction is roughly linear in input size, $C_{\text{feat}} = O(N\,p\,T)$, with a small additional overhead for histogram updates and indexing; Random Forest cost depends on implementation-specific stopping criteria and tree depth, so only a coarse characterization is offered here – it grows with sample count, feature dimension, and the number of trees evaluated per fold.

Table 5.6 reports DIRECT tuning time and the number of objective evaluations required per dataset, under the budget and stopping criteria of Section 5.2.2. Method B is not tuned independently – it reuses Method A's selected configuration – so its tuning time and evaluation counts are identical to Method A's.

The 1NN–DTW$_{\text{D}}$ baseline uses a Sakoe–Chiba warping window of 10% of series length, a default widely adopted as a strong accuracy–runtime trade-off [178, 179]. Because DTW runtime is highly sensitive to the allowed warping range, a small

Table 5.6: DIRECT tuning time (seconds) and number of objective evaluations on the common 16-dataset subset. Method B reuses Method A's tuned configuration, so its tuning time and evaluation count match Method A's.

	DIRECT tuning time (s)				# evaluations			
Dataset	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**
AtrialFibrillation	19.61	19.61	13.26	14.41	53	53	77	87
BasicMotions	29.96	29.96	34.31	69.30	127	127	127	208
CharacterTrajectories	66.68	66.68	72.99	68.52	58	58	100	59
Cricket	30.96	30.96	49.58	27.60	61	61	132	51
EigenWorms	46.96	46.96	88.47	54.08	55	55	124	59
Epilepsy	31.32	31.32	26.32	27.79	103	103	77	78
EthanolConcentration	41.01	41.01	34.88	47.06	67	67	73	95
ERing	22.40	22.40	19.43	19.89	64	64	88	79
Handwriting	23.94	23.94	51.26	21.96	63	63	145	51
Libras	34.23	34.23	30.14	31.30	111	111	141	103
LSST	47.14	47.14	6.54	1308.24	17	17	3	173
RacketSports	42.55	42.55	38.05	102.27	73	73	96	89
SelfRegulationSCP1	36.04	36.04	62.05	247.34	66	66	121	182
SelfRegulationSCP2	50.74	50.74	38.87	382.99	112	112	81	86
StandWalkJump	19.38	19.38	10.80	16.72	122	122	53	82
UWaveGestureLibrary	27.40	27.40	28.96	36.30	114	114	123	181
SUM	570.32	570.32	605.91	2475.77	1266	1266	1561	1663
AVERAGE	35.65	35.65	37.87	154.74	79.13	79.13	97.56	103.94

sensitivity check repeats the baseline with 5%/10%/20% windows on six representative datasets, reporting runtime and accuracy alongside their deltas relative to the 10% default (Table 5.7). This documents the accuracy–speed trade-off across window choices and indicates that the conclusions drawn here are not sensitive to the specific choice of a 10% band.

All benchmark runs were executed on a single workstation (Dell Precision 3680) running Windows 11 Pro (64-bit), with an Intel Core i9-14900K CPU (24 cores, 32 logical processors, ~3.2 GHz) and 128 GB of RAM, ensuring all timing comparisons are made under identical hardware and software conditions. For Methods A–D, Table 5.8 reports both (i) total end-to-end runtime, including DIRECT tuning, and (ii) comparable runtime – feature extraction, training, and inference under the already-selected parameters – which isolates the one-off tuning cost from the per-run deployment cost and enables a fair comparison against baselines that perform no dataset-level tuning under this protocol.

On this subset, all four proposed methods are substantially faster than the baselines, with Methods A–C particularly strong: Method A is fastest on average (1.51 s comparable runtime), followed by B (1.58 s) and C (2.46 s); Method D is slower (3.41 s) but still well ahead of every baseline. Among baselines, MultiROCKET is fastest overall (4.70 s), SAX-VSM is the second-fastest baseline (10.47 s, fastest on CharacterTrajectories), and BORF follows (21.94 s) as the most efficient symbolic baseline;

Table 5.7: Sensitivity of 1NN–DTW$_D$ to the Sakoe–Chiba window width. Runtime (s) and accuracy (%) are reported for 5%, 10% (default), and 20%, together with differences relative to the 10% setting. ΔAcc is in percentage points (pp); ΔRT is in seconds.

Dataset	Win.	Runtime (s)	Acc. (%)	ΔAcc vs 10% (pp)		ΔRT vs 10% (s)	
				5–10	20–10	5–10	20–10
ArticularyWordRecognition	5%	42.23	98.00	-0.67	+0.33	-0.22	+0.72
	10%	42.45	98.67				
	20%	43.17	99.00				
CharacterTrajectories	5%	231.02	98.12	+0.42	-0.35	-4.00	+3.33
	10%	235.02	97.70				
	20%	238.35	97.35				
EthanolConcentration	5%	711.53	31.18	+1.52	+0.76	-26.40	+14.06
	10%	737.93	29.66				
	20%	751.99	30.42				
Libras	5%	11.11	83.33	-4.45	+2.78	-0.13	+0.01
	10%	11.24	87.78				
	20%	11.25	90.56				
RacketSports	5%	11.91	84.87	+0.66	+0.66	-0.08	+0.04
	10%	11.99	84.21				
	20%	12.03	84.87				
Handwriting	5%	27.78	48.47	-1.18	+1.06	-0.21	+0.60
	10%	27.99	49.65				
	20%	28.59	50.71				

WEASEL+MUSE (86.97 s), InceptionTime (649.69 s), and DTW$_D$ (2653.46 s) are considerably slower. Method A attains the lowest comparable runtime on 7 of 16 datasets and is never the slowest method on any of them; MultiROCKET wins on the remaining 8, Method C once, and Methods B and D, along with every baseline besides MultiROCKET and Method A, never finish fastest. Even including DIRECT's one-off tuning cost in the total runtime, Methods A–D remain far below the slower baselines: average totals of 37.16, 37.23, 40.33, and 158.15 s for A–D, against 2653.46, 649.69, and 86.97 s for DTW$_D$, InceptionTime, and WEASEL+MUSE, respectively.

The corresponding CD diagram (Figure 5.9, CD = 3.39) confirms this ordering. Method A attains the best average rank among all ten methods (2.09), with B (2.84) and C (3.94) also occupying the fast end of the spectrum; MultiROCKET is competitive with this group at 2.75. Method D has the weakest rank among the proposed methods (4.66), reflecting its higher per-channel tuning cost, but still clearly outperforms DTW$_D$ (9.13), WEASEL+MUSE (6.81), BORF (7.63), and InceptionTime (9.81). The CD value indicates that the gap between the fast cluster (A–D, MultiROCKET, SAX-VSM) and the slow cluster (WEASEL+MUSE, BORF, DTW$_D$, InceptionTime) is large enough to be statistically significant at $\alpha = 0.05$.

Figure 5.10 plots average accuracy against total runtime to illustrate this trade-

Table 5.8: Runtime comparison on the common 16-dataset subset. Total time for Methods A–D includes DIRECT tuning. Comparable runtime reports the test-phase time (feature extraction + training + inference) for Methods A–D using the selected parameters, and end-to-end runtime for the baselines. Bold marks the lowest runtime per dataset within the comparable-runtime block. Win/Tie/Lose, average ranks, and CD are computed on the comparable-runtime block via Friedman/Nemenyi at $\alpha = 0.05$. All times are in seconds.

	Total time incl. tuning (s)				**Comparable runtime (s)**									
Dataset	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**DTW**	**MultiROCKET**	**Inception Time**	**WEASEL +MUSE**	**BORF**	**SAX-VSM**
AtrialFibrillation	20.51	20.51	15.14	16.59	0.90	0.90	1.88	2.18	14.52	**0.42**	66.20	2.07	10.32	2.05
BasicMotions	31.38	31.39	36.43	71.36	1.42	1.43	2.12	2.06	19.37	**0.37**	49.60	2.24	9.93	2.06
CharacterTrajectories	69.14	69.21	75.72	71.91	**2.46**	2.53	2.73	3.39	235.02	4.90	320.51	34.16	18.62	4.18
Cricket	32.26	32.66	51.85	29.88	**1.30**	1.70	2.27	2.28	110.67	2.87	314.30	49.93	14.19	3.80
EigenWorms	48.70	48.96	91.32	57.12	**1.74**	2.00	2.85	3.04	39511.94	43.67	6682.94	909.97	68.91	105.26
Epilepsy	33.45	33.47	28.44	29.88	2.13	2.15	2.12	2.09	27.21	**0.76**	90.98	5.77	23.42	2.38
EthanolConcentration	43.49	43.56	37.20	49.45	2.48	2.55	**2.32**	2.39	737.93	5.38	1085.85	92.95	17.52	10.75
ERing	23.34	23.36	21.76	22.11	0.94	0.96	2.33	2.22	20.01	**0.49**	47.66	2.40	10.35	2.05
Handwriting	26.04	26.29	53.79	24.71	2.10	2.35	2.53	2.75	27.99	**1.40**	84.68	9.55	12.47	2.81
Libras	36.39	36.39	32.27	33.46	2.16	2.16	2.13	2.16	22.66	**0.72**	63.91	1.50	13.95	1.95
LSST	48.85	49.07	11.25	1317.42	1.71	1.93	4.71	9.18	706.64	**0.26**	272.75	38.64	66.34	3.17
RacketSports	43.88	43.45	40.27	108.00	1.33	0.90	2.22	5.73	11.99	**0.64**	57.61	1.80	14.65	2.08
SelfRegulationSCP1	36.90	36.91	64.49	253.37	**0.86**	0.87	2.44	6.03	528.03	4.85	518.33	112.70	25.84	9.71
SelfRegulationSCP2	51.64	51.83	41.40	387.87	**0.90**	1.09	2.53	4.88	425.24	6.29	529.41	107.26	23.05	10.24
StandWalkJump	20.21	20.22	12.85	18.70	**0.83**	0.84	2.05	1.98	20.15	1.14	107.29	9.82	10.67	2.40
UWaveGestureLibrary	28.37	28.39	31.08	38.50	**0.97**	0.99	2.12	2.20	36.00	1.02	103.00	10.79	10.81	2.57
AVERAGE	37.16	37.23	40.33	158.15	1.51	1.58	2.46	3.41	2653.46	4.70	649.69	86.97	21.94	10.46
Win/Tie/Lose			–		7/0/9	0/0/16	1/0/15	0/0/16	0/0/16	8/0/8	0/0/16	0/0/16	0/0/16	0/0/16
Average Rank			–		2.09	2.84	3.94	4.66	9.13	2.75	9.81	6.81	7.63	5.34
CD			–							3.39				

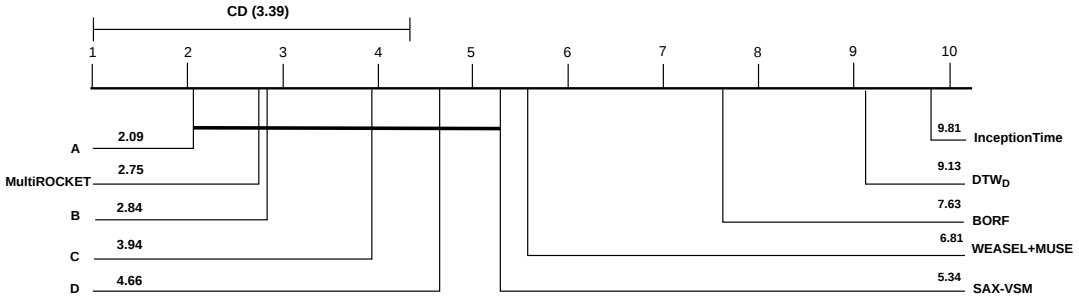


Figure 5.9: Critical Difference (CD) diagram comparing runtime on the common 16-dataset subset. Ranks are computed from the comparable runtime measure (feature extraction + training + inference for Methods A–D; end-to-end runtime for the baselines), with lower average rank indicating faster execution.

off directly. MultiROCKET sits closest to the ideal corner, combining the highest accuracy with a comparatively low runtime; BORF offers a strong mid-range option, with high accuracy at moderate cost, while WEASEL+MUSE reaches similarly high accuracy only at substantially higher runtime. Among the proposed variants, A and B form the most attractive speed–accuracy compromise – the fastest methods overall while retaining moderate-to-high accuracy – whereas C and D are slower without a compensating accuracy gain. SAX-VSM is the lowest-performing method plotted, failing to reach 45% accuracy despite a runtime comparable to the much more accurate symbolic variants, while $\mathrm{DTW_D}$ and InceptionTime are runtime-heavy for the accuracy they deliver: $\mathrm{DTW_D}$ is by far the slowest method plotted yet only mid-range in accuracy, and InceptionTime is similarly expensive while trailing most other methods in accuracy.

Since runtime might plausibly scale with series length, this relationship is examined directly in Figure 5.11. Methods A–D occupy the lowest band across almost the entire length range, requiring only a few seconds even as sequence length grows, which suggests comparatively stable scaling on this subset; SAX-VSM exhibits a similarly efficient profile, sitting just above the proposed methods. MultiROCKET is also generally low, often overlapping with A–D, though with a slightly wider spread on some datasets. InceptionTime and $\mathrm{DTW_D}$, by contrast, show a much steeper increase in runtime with length and the largest variability, dominating the upper region of the plot – particularly for the longest sequences – which is consistent with poorer computational scalability. WEASEL+MUSE sits between these extremes, typically slower than A–D and MultiROCKET and growing clearly with length, though not reaching the very large runtimes seen for $\mathrm{DTW_D}$ and InceptionTime; BORF occupies a comparable mid-tier band, more efficient than WEASEL+MUSE but costlier than the fastest symbolic methods. Overall, the proposed symbolic variants – together with MultiROCKET – offer the most favorable runtime scaling as sequence length increases, while $\mathrm{DTW_D}$ and InceptionTime become prohibitively expensive on long sequences.

5.3.3 Interpretability Case Study

Method B serves as the representative case study for interpretability, since it converts each channel into a SAX sequence and represents every multivariate series via (i) a BOP histogram of per-channel SAX words and (ii) the MII, yielding an explicit,

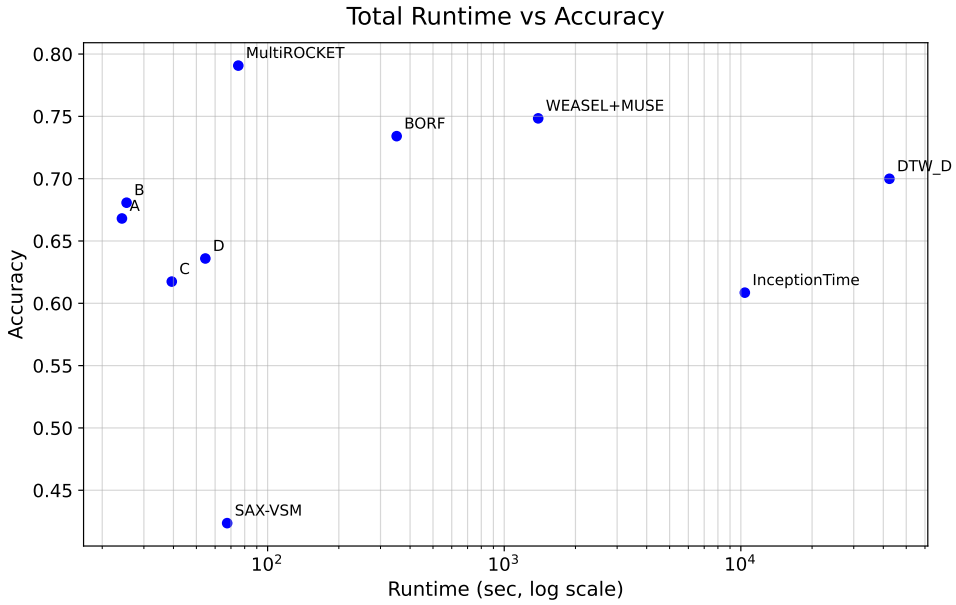


Figure 5.10: Trade-off between total runtime and accuracy on the common subset. Each point is one method; the x-axis shows total training and inference time (seconds, log scale) summed over the subset's datasets, and the y-axis shows average test accuracy. DIRECT tuning time is excluded, since parameters are tuned once per dataset and reused across all resamples.

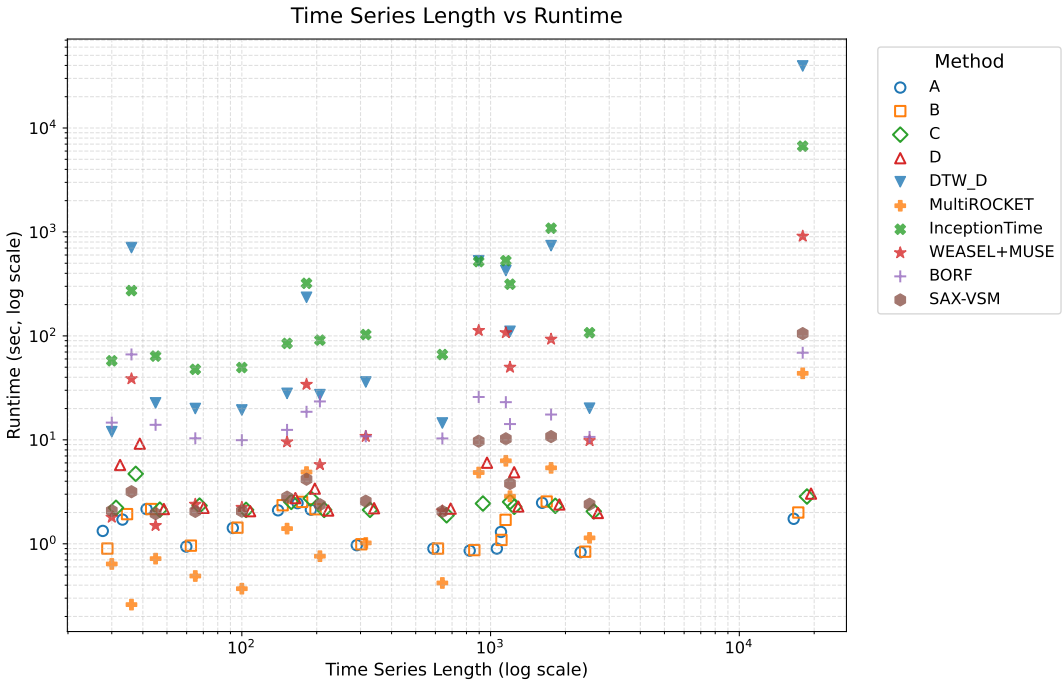


Figure 5.11: Time series length versus runtime. Each point corresponds to one dataset–method pair; the x-axis is the dataset's sequence length and the y-value is total training and inference time for that dataset. DIRECT tuning time is excluded.

sparse feature space in which every multichannel word has a direct semantic meaning – a symbolic pattern across channels – and can be traced back to the time interval(s) where it occurs.

To identify what drives a given classification, class-discriminative multichannel words are scored in a one-vs-rest setting: for each class, every multichannel word is scored on the training split via a chi-square criterion measuring how strongly its occurrence counts are associated with that class versus all others, and the top-ranked words per class are localized in the original signals by scanning the corresponding SAX sequences for occurrences and mapping their symbolic positions back to raw time indices through the SAX segmentation.

This procedure is demonstrated on the BasicMotions training split, using Method B's tuned configuration for that dataset ($\rho = 0.02444$, $a = 2$, $w = 5$). Figure 5.12 traces a single discriminative multichannel word back to concrete, repeated, time-localized events in the raw signal. For the walking class, the most discriminative multichannel word is "bbabab"; the figure shows one representative sample (sample 26) with six shaded vertical bands overlaid across all six channels, corresponding exactly to the word's occurrences in the SAX representation. The word appears at SAX positions $[8, 9, 15, 16, 23, 30]$, which map back to raw-sample spans $[(16, 18), (18, 20), (30, 32), (32, 34), (46, 48), (60, 62)]$: rather than activating indiscriminately, the model repeatedly fires on specific short intervals distributed throughout the sequence.

This is not a post-hoc saliency map: each highlighted region is exactly the time segment that produced the selected symbolic word. The interpretability is inherent to the representation itself – each SAX symbol summarizes a fixed-length segment of the original signal via PAA and discretization (Sections 2.2.1, 2.2.2), and words are built directly from these symbols – so any selected word can be deterministically traced back to the raw-signal segment(s) that generated it, with no separate explanation model required.

5.4 Discussion

The evaluation in Section 5.3 shows that the proposed symbolic methods – Method B in particular – can reach accuracy competitive with several established baselines, which is notable given the method's conceptual simplicity, interpretability, and computational cost. At the same time, Table 5.4 makes clear that this competitiveness is far from uniform: how well the symbolic representation performs depends heavily on the dataset and the domain it comes from.

A clearer picture emerges by grouping datasets into performance clusters and relating each cluster to the kind of temporal structure it appears to require. Four such clusters are visible: a small group on which the proposed representation is exceptionally effective; a broader group where it is very good to excellent; a group where it remains broadly competitive with the strongest baselines; and a group of clear failure cases. This grouping is informative precisely because the downstream classifier is held fixed throughout – what varies is the match between a dataset's discriminative structure and what SAX is able to preserve.

The first cluster – AtrialFibrillation, where Methods A and B are particularly strong, and SelfRegulationSCP2, where B stands out while A/C/D perform similarly – is dominated by short, recurring motifs and coordinated multichannel patterns that tuned SAX discretization and multichannel words capture directly. In AtrialFibrillation, class evidence is often expressed through rhythm irregularities and waveform morphology

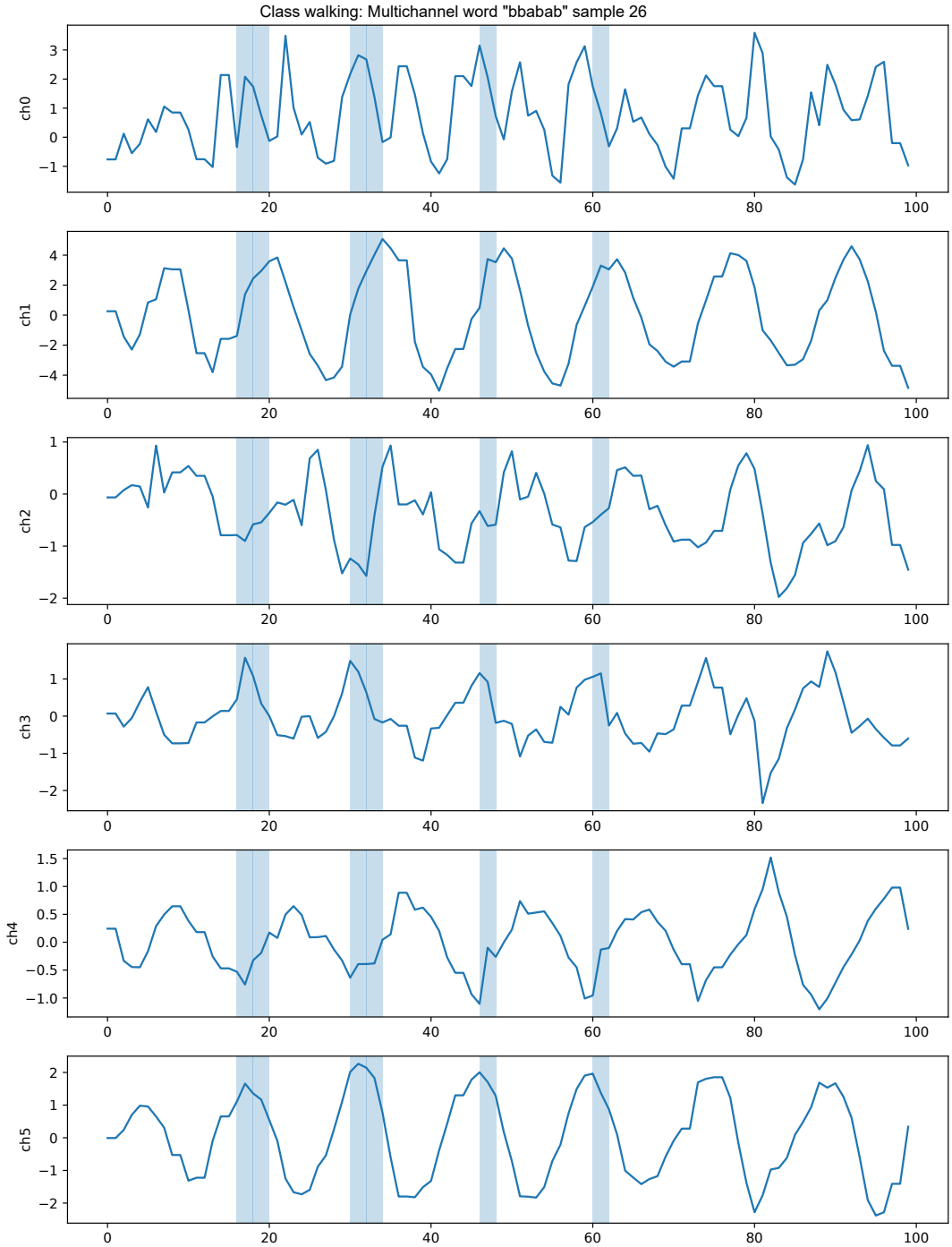


Figure 5.12: Most discriminative multichannel word for the walking class in BasicMotions. The multichannel word “bbabab” highlights repeated cross-channel configurations consistent with periodic motion patterns.

fragments that reduce naturally to recurring symbolic patterns; that A and B excel here suggests that capturing stable local motifs is enough to outperform methods built on more generic feature mappings. In SelfRegulationSCP2, Method B’s advantage over the other symbolic variants is consistent with discrimination depending more on cross-channel coordination than on per-channel patterns considered in isolation.

A second cluster – ArticularyWordRecognition, BasicMotions, Cricket, and Epilepsy – shows very good to excellent performance, and these datasets are structurally well suited to a motif-based symbolic view. BasicMotions and Cricket exhibit near-periodic activity cycles with repeated motion primitives, which symbolic words represent compactly while remaining tolerant of small variations, and multichannel encoding helps further when sensors move together. Epilepsy similarly contains distinctive transient events and recurrent waveform shapes. That strong accuracy is achievable here with a deliberately simple classifier reinforces the broader pattern: success in this regime comes from extracting the right repeatable local structures, not from a sophisticated decision boundary.

A third, "mixed-regime" cluster – FingerMovements, Heartbeat, LSST, MotorImagery, and StandWalkJump – is where the proposed methods are typically competitive with the best baseline or close to it. These datasets contain some motif-like evidence that symbolic discretization captures well, but also variability – subject effects, amplitude or phase differences, longer-range context – that the stronger baselines are better positioned to exploit.

The most diagnostic cluster for understanding the method's limits is the fourth: ERing, Handwriting, JapaneseVowels, Libras, RacketSports, and UWaveGestureLibrary, where the gap to the best baseline is substantial – roughly 15 to 40 percentage points. A gap of this size signals more than a small loss of capacity; it points to a representational mismatch, where the symbolic encoding fails to preserve the cues that actually separate classes in these domains. Several of these datasets depend on fine-grained continuous structure – subtle trajectory-shape differences in Handwriting and Libras, or spectral and phonetic characteristics in JapaneseVowels – or on complex temporal evolution that is not well expressed as a small set of repeated local motifs. Here, PAA smoothing and coarse discretization can collapse distinctions that methods such as MultiROCKET or deep models retain, and multichannel dependencies may be too diffuse or context-dependent for compact multichannel words to capture. This also explains a statistical artifact visible in the aggregate results: because the shortfall is large on this cluster, these few datasets disproportionately pull down the mean accuracy even though the method is strong or competitive across most of the others.

Taken together, this clustering supports a practically useful conclusion. The proposed framework is most effective when class identity is expressed through recurring, localized motifs – and, for Method B, through repeatable cross-channel coordination; it remains competitive on mixed-regime problems; and it can fail sharply where discrimination depends on subtle, highly variable, or texture-like continuous structure that discretization does not preserve well. This clarifies when the approach should be preferred, as an efficient and interpretable alternative, and when richer feature mappings or higher-capacity models should be expected to do better.

Section 5.3.1 already traced why shared-parameter tuning (A and B) generalizes more robustly than per-channel tuning (C and D) in this benchmark. The practical recommendation that follows is to prefer Method B when cross-channel interactions are informative – since it captures them through the MII at no extra tuning cost beyond Method A's – and to fall back to Method A as a lighter alternative when this additional multichannel feature is unnecessary or when memory is constrained. Method D remains worth considering in settings with genuinely heterogeneous channels that benefit from some per-channel adaptation, provided the shared PAA segmentation needed for the icon-based representation is preserved; Method C, by contrast, is best read as an ablation rather than a deployable configuration, since it combines the

largest search space with the weakest accuracy among the four.

A central strength of the proposed framework is its interpretability: classification operates on sparse, human-readable symbolic features that correspond to localized temporal motifs, and the MII further exposes cross-channel symbolic interactions on top of these. Computationally, the symbolic pipeline scales favorably with sequence length and keeps deployment-time runtime low (Section 5.3.2), which together make it attractive for time-sensitive or resource-constrained settings.

These properties suit deployment scenarios that need fast inference, interpretability, and reliable accuracy under a constrained compute budget at the same time. On-device physiological or activity monitoring – wearable ECG/EEG, or multi-sensor movement tracking – is a representative example: decisions must be produced near real time and backed by evidence a human can audit. In such settings, DTW-based approaches can become prohibitively expensive as sequence length grows, and deep models are often impractical given their heavier compute requirements and limited transparency, whereas Method B offers accurate predictions together with localized, multichannel symbolic evidence that can be directly inspected and validated.

Two limitations follow from the same analysis. The first is representational mismatch: the proposed methods perform well when classes are characterized by recurring local motifs and cross-channel coordination, but can degrade sharply on datasets dominated by subtle, highly variable, or more continuous dynamics – handwriting- and speech-like problems being the clearest examples – where PAA smoothing and discretization remove exactly the fine-grained cues that stronger baselines exploit. The second is scope of applicability: although runtime scales favorably with sequence length, modeling cross-channel interactions grows more costly as the number of channels increases, which confines the approach to low-to-moderate dimensional MTS; at the other extreme, very short sequences leave too little temporal support for meaningful segmentation and quantization, making the resulting symbolic vocabulary unstable and the aggregated features largely uninformative. Together, these observations delimit the method's intended operating regime: multivariate time series with enough temporal resolution to support meaningful local pattern extraction, and with a channel dimensionality that allows cross-channel structure to be modeled without the feature set growing to a prohibitive size.

5.5 Synthesis

This chapter set out to close this gap: a principled, automated tuning procedure for SAX that extends beyond the univariate, weighting-entangled setting into MTSC proper. Applying DIRECT directly to the PAA segment-size fraction, alphabet size, and word length – shared across channels or tuned per channel, with or without the MII – and validating the result across 24 datasets from the UEA archive rather than a single heuristically-tuned case, confirms that globally optimized symbolic representations remain a viable, dataset-adaptive alternative to costlier models, at least within the operating regime characterized in Section 5.4.

This evaluation also completes the trajectory promised in Chapter 4: MII moves from a single-dataset proof-of-concept, evaluated with parameters fixed by hand, to one ingredient of Method B, evaluated under DIRECT-tuned, dataset-adaptive configurations across a far larger and more heterogeneous benchmark. The outcome is a more complete picture than either chapter could offer alone. Method B's cross-channel feature pays off precisely where Chapter 4's single HAR dataset could not reveal,

namely whenever class identity depends on coordinated multichannel dynamics, but the broader benchmark also exposes where it does not: high-dimensional channel counts that inflate the multichannel vocabulary, and datasets whose discriminative structure is too fine-grained or continuous for PAA-based discretization to preserve. Generalizing MII's evaluation in this way was only possible because Section 5.1.1 replaced its originally hand-fixed parameters with a tuned, dataset-adaptive configuration; the two contributions of this chapter – the tuning procedure and the multi-dataset evaluation – are therefore inseparable in practice, not just in framing.

Read together with the per-channel-versus-shared-tuning analysis in Section 5.3.1, the practical recommendation that emerges is to prefer shared-parameter tuning over per-channel tuning by default: Method B when cross-channel coordination is likely to be informative, since it inherits Method A's tuned configuration at no extra search cost, and Method A itself as a lighter, more memory-stable fallback otherwise. This recommendation is not offered as a universal replacement for deep or ensemble MTSC models, which remain the stronger choice when the discriminative signal is diffuse, continuous, or context-dependent in the way Section 5.4's fourth cluster illustrates. Rather, it positions the proposed framework as a complementary, auditable alternative for settings – such as on-device physiological or activity monitoring – where compute and tuning budgets are limited and decisions must remain traceable to specific, human-readable evidence.

The operating-regime limits identified in Section 5.4 point to three directions for extending this framework rather than abandoning it. The first concerns scalability: as channel count grows, the multichannel token space underlying the MII grows with it, so controlling that growth – through interaction pruning, channel grouping, or hashing/sketching of the token space – would let cross-channel features remain tractable without the feature set becoming prohibitively large. The second concerns transferability: since DIRECT currently tunes each dataset independently from scratch, studying whether tuned configurations transfer across datasets of a similar nature could identify regimes where a single configuration generalizes, and motivate meta-feature-guided warm starts that reduce tuning cost without sacrificing the dataset-adaptivity demonstrated in this chapter. The third concerns channel heterogeneity: learning data-driven importance weights for individual channels, and propagating them into the higher-order MII features, would let multichannel tokens be emphasized only when they involve genuinely informative channels – reducing the influence of noisy or weak channels on the final representation while preserving the interpretability that motivated this framework in the first place.

This page has been left blank intentionally.

Chapter 6

Symbolic Tokenization for LLM-Based Time Series Classification

Chapter 3's taxonomy identified trend loss as one of the most heavily populated weakness categories in the SAX literature (Section 3.4). Chapter 4 already answered this gap once, with SAA (Section 4.2): a parallel, independently discretized angle-based branch that captures segment slope alongside the standard mean-based SAX symbol. This chapter answers the same gap a second time [180], through a structurally different mechanism. Rather than computing trend in a separate branch, the representation developed here folds trend directly into the case of the existing SAX symbol – mapping each segment's slope to an uppercase or lowercase variant of the same character – so that a single symbolic string carries both amplitude and direction without doubling the representation's dimensionality. SAA and this trend-aware encoding therefore share a motivation while committing to opposite design choices: SAA keeps trend information in a separately discretized stream that can be fused or analyzed independently of the mean-based symbol; the case-modulation introduced here instead fuses trend into the very same symbol used for amplitude, trading the ability to inspect trend in isolation for a more compact, single-string representation. That trade-off matters once the representation has to serve as a token sequence for a language model, where every additional symbol lengthens the prompt, and where a structurally separate trend stream would in any case have to be serialized back into the same text.

This chapter also returns to human activity recognition (HAR), the domain in which Chapter 4 evaluated MII and SAA, but pursues a different question: not how to extend SAX's accuracy through richer engineered features, but whether a symbolic representation can serve directly as the input vocabulary of an LLM, letting the LLM's pre-trained sequential-reasoning capacity stand in for the hand-built classifiers used elsewhere in this thesis. The motivating observation is structural: SAX already converts a continuous signal into an ordered sequence of discrete tokens, precisely the input format LLMs are built to process, whereas LLMs have no native mechanism for ingesting raw continuous sensor readings. Bridging the two is therefore, in principle, a matter of representation design rather than architecture: SAX already outputs the kind of ordered discrete token sequence an LLM expects, so no structural adaptation is required – only a symbolic representation expressive enough that the tokens retain what made the original signal classifiable.

As reviewed in Section 2.3, SAX-based classifiers such as BOP and its descendants

This chapter is based on the following publication: [J3].

remain interpretable and computationally light, but typically rely on heuristically chosen parameters and comparatively simple downstream classifiers, which struggle to capture the complex sequential dependencies present in modern, fine-grained activity data. LLMs are a natural complement, since they excel precisely at modeling dependencies over long discrete token sequences, but they are fundamentally mismatched to raw, continuous signals: they carry no inductive bias for physical dynamics, and existing attempts to bridge this gap – through text-reprogramming or multimodal fusion – tend to discard essential physical semantics or rely on ad hoc encodings that obscure the resulting decision logic, exactly the property that motivated symbolic representations in this thesis in the first place. The remainder of this chapter develops a way to close this gap that stays faithful to that motivation: a symbolic representation expressive enough to serve as an LLM's native vocabulary, augmented just enough to recover the physical information symbolic discretization would otherwise discard.

This chapter proposes SAX_HAR-LLM, a dual-stream framework that pairs the trend-aware SAX encoding introduced above with lightweight kinematics-informed descriptors, fuses both into a structured textual prompt, and fine-tunes a causal LLM to complete that prompt with the corresponding activity label. Section 6.1 situates this design within the literature on using LLMs for time series, not yet covered elsewhere in this thesis. Section 6.2 details the proposed encoding, prompt construction, fine-tuning procedure, and interpretability analysis. Section 6.3 describes the datasets, baselines, and evaluation protocol; Section 6.4 reports classification, statistical, ablation, attention, and computational-cost results on two HAR benchmarks; and Section 6.5 closes the chapter by relating these findings back to the thesis's broader account of SAX-based representation design.

6.1 Background

Section 2.4 already surveyed the broader landscape of time series classification, including the SAX-based dictionary methods this chapter builds on. The remainder of this section instead covers a literature not yet introduced in this thesis: how symbolic sequences, and SAX in particular, relate to sequence-based language models, and how LLMs have so far been applied to time series data.

6.1.1 Symbolic Sequences as Inputs to Sequence-Based Models

In natural language processing, discrete symbols are typically mapped to distributed embeddings to support compositional reasoning, a step that enables flexible generalization but obscures the identity of the original symbol [181]. Beyond natural language, symbolic sequences are interpreted through structured operators in several neuro-symbolic settings: GNN-QE executes first-order logic queries expressed as symbolic sequences [182], neuro-symbolic temporal reasoners verify execution traces against linear temporal logic specifications [183], and planning-based models treat sequences as action traces with explicit preconditions and effects [184]. The common thread across these examples is that transforming numeric or structured data into symbolic tokens makes it directly usable by sequence-reasoning models built for text – a property that extends naturally to symbolic time series representations, which are structurally analogous to text and align with prior work on interpretable, rule-based reasoning over discrete human-activity descriptions [185].

Purely symbolic representations, however, have a cost: discretization and nor-

malization can remove essential physical information – absolute magnitude, gravity, orientation – introducing ambiguity once the representation is taken out of its original numeric context [185, 186]. Neural-only models avoid this cost but obscure symbolic structure and decision logic in the process, and although recent neuro-symbolic work aims to balance the two, symbolic time series representations are still rarely treated as first-class token sequences for modern sequence models [186, 187]. Two recent examples illustrate what treating them as first-class inputs can look like outside HAR specifically: Onchis and Hogea [188] augment a one-dimensional Transformer with logical constraints for fault diagnosis, improving F1-score over a transformer alone while keeping classifications interpretable, and Zhang et al. [189] discretize vibration signals into quantized frequency tokens and route them through a frozen LLM via retrieval-augmented instruction alignment, achieving strong generalization across heterogeneous machinery. Both examples motivate the central question this chapter pursues for HAR specifically: whether a symbolic representation can be made informative enough, on its own, to serve as a first-class LLM input rather than an auxiliary signal.

6.1.2 Large Language Models for Sequential Modeling

LLMs have demonstrated emergent capabilities for sequential and symbolic reasoning that go well beyond generating grammatically fluent text [190,191]. A prominent driver of this behavior is Chain-of-Thought (CoT) prompting, which elicits intermediate reasoning steps from the model, effectively decomposing a hard problem into an ordered chain of symbolic operations [191,192]. The resulting step-by-step structure has proven productive across a wide variety of tasks requiring sustained consistency over long token sequences, including arithmetic, logic puzzles, and program execution [193].

Prompting alone, however, is not the only route to stronger reasoning. Instruction tuning, knowledge distillation, and neuro-symbolic integration each offer complementary mechanisms. On the distillation side, temporal relational reasoning – event sequencing, causal ordering, narrative coherence – can be transferred from a large teacher to a smaller student model with measurable fidelity [194]. On the neuro-symbolic side, frameworks such as Logic-LM sidestep the problem of unreliable in-context inference altogether by translating natural language into a formal symbolic representation – first-order logic, explicit constraints – and delegating execution to a deterministic solver, substantially reducing hallucination and improving logical faithfulness [193]. ProverGen pursues the same separation for mathematical proof generation, gaining robustness on problems outside the training distribution [192]. What unites these approaches is a shared architectural intuition: LLMs process structured symbolic token sequences most reliably when temporal or logical dependencies are stated explicitly in the input rather than left to be inferred – precisely the property that a symbolic time series encoding is designed to provide.

That said, a direct connection between LLMs and raw continuous signals remains elusive. Individual numeric values carry no semantic content that a language model's pre-trained vocabulary can latch onto [195], and unstructured numeric sequences lack the discrete tokenizable form that the LLM's input pipeline assumes [196]. Converting a continuous signal into a fixed symbolic vocabulary bridges this gap by presenting the model with input that is structurally compatible with its pre-trained pattern-matching capacity [197]. Theoretical support for this move comes from analysis showing that Transformers can carry out relational reasoning over inputs encoded as symbolic strings, essentially processing them as instances of learnable template

structures [198]. Empirical support comes from decomposition-based methods such as TEMPO, which show that making temporal structure explicit – separating a series into trend and seasonality components before passing it to the LLM – measurably improves distributional alignment between the model and the input signal [199].

6.1.3 Paradigms for Applying LLMs to Time Series

Work adapting LLMs specifically for time series analysis has converged on four broad methodological paradigms [200]. The first, *prompting*, treats numerical data as raw text or digit sequences and frames the task as next-token prediction, leveraging the LLM's native sequential modeling without any architectural change [201, 202]; PromptCast [203] converts numerical data into sentence-to-sentence templates this way, and LLMTime [202] uses digit-level tokenization to keep temporal values represented consistently. The second, *time series quantization*, discretizes continuous signals into symbolic tokens the LLM can process natively – the paradigm this chapter's own SAX-based encoding belongs to. LLM-ABBA [204] is the closest published example, using adaptive Brownian-bridge-based symbolic aggregation to match state-of-the-art benchmarks, while Numerical Greedy Tokenization instead constrains the input to a predefined set of numerical symbols, aiming to unlock latent reasoning while avoiding interference from natural-language tokens [197]. Both routes share the same underlying move: translating the intrinsic syntax of a time series into a token sequence the LLM can read directly [205].

The third paradigm, *aligning*, instead uses a neural encoder to map continuous temporal embeddings into the LLM's existing semantic space – exemplified by reprogramming frameworks such as Time-LLM [206] and TEST [207], which align signal patches with textual prototypes [208], and by fine-tuning strategies such as GPT4TS, which adapts pre-trained attention patterns directly to temporal data [209]. The fourth, *vision as a bridge*, instead routes time series through a visual or spatial intermediate representation; IMUGPT, for example, synthesizes 3D human-motion sequences from textual descriptions to derive virtual IMU sensor signals via the principles of motion kinematics [210].

A further, more granular trend layers semantic enrichment on top of these four paradigms: recent frameworks augment numerical patches with explicit textual descriptions and domain-specific statistics for each variable [195], or combine hierarchical text summarization with bidirectional co-attention so that textual semantics and temporal dynamics reinforce each other without either structure being lost [211].

6.1.4 Current Challenges and the Proposed Approach

Whether LLMs genuinely help with time series, however, remains an open question. LLMs show strong zero-shot generalization in data-scarce settings, but several ablation studies find that a simple linear model or basic attention layer can match or exceed their performance at a fraction of the computational cost [201, 212]. Reprogramming-style methods face a sharper criticism still: recent analysis suggests they can produce pseudo-alignment, where the model transfers only the centroid of the time series distribution into the language space without genuinely aligning the two manifolds, so that the LLM's reasoning capacity is never actually exercised [213]. More fundamentally, continuous signals and discrete text tokens remain a structural mismatch [214], and tokenization schemes built for language are inefficient at representing the high-precision numerical dependencies a time series can carry [215]. Recent work on unified temporal

tokenization proposes a hybrid semantic-numeric encoding that preserves cyclical temporal structure directly in token form [216], reinforcing that domain-specific tokenization design, not a generic text vocabulary, is what ultimately makes a time-aware LLM work.

Table 6.1 consolidates the key limitations raised across this section and Section 2.4, motivating the design pursued in the remainder of this chapter. The approach adopted here sits directly in the time-series quantization paradigm: SAX discretizes a continuous signal into symbolic tokens, establishing a direct structural alignment with the text-based interface LLMs already expect, while the case-modulation trend encoding – mirroring SAA's motivation, Section 4.2 – and the kinematics-informed descriptors introduced in Section 6.2 compensate for the physical information that discretization and normalization would otherwise discard. Because SAX already produces ordered symbolic sequences – precisely the kind of input over which language models excel – the LLM is not applied to time series data arbitrarily; symbolic time series representations and natural language occupy the same discrete, sequential space by construction, and the framework developed below simply makes that correspondence explicit.

Table 6.1: Methodological families relevant to time series classification and LLM-based sequence modeling, with the key strength and principal limitation of each as identified in Section 2.4 and the literature reviewed above.

Methodological Family	**Key Strength**	**Principal Limitation**
Distance-based TSC (e.g., DTW)	No feature engineering; strong alignment	High computational cost; poor scalability; limited interpretability
Feature-based TSC (e.g., TSFresh)	Interpretable features	Loss of temporal ordering; weak long-range dependency modeling
Convolutional/kernel methods (ROCKET family)	High accuracy; efficient training	Large opaque feature spaces; limited explainability
Deep learning ensembles	State-of-the-art accuracy	Computationally expensive; black-box behavior
Symbolic TS (SAX, SFA, BOSS)	Interpretability; efficiency; lower-bounding	Loss of physical magnitude; ambiguity after normalization
LLM-based TS (numeric or text-reprogrammed)	Zero-shot capability; long-range dependency modeling	Continuous–discrete mismatch; inefficient tokenization; pseudo-alignment

6.2 Methodology

The proposed framework, SAX_HAR-LLM, treats activity recognition as a supervised prompt-completion task: a pre-trained causal language model is fine-tuned on structured prompts that pair symbolic and physical representations of each inertial sensor window with its activity label. The dual-stream design addresses the next limitation: a purely symbolic representation discards the absolute physical information that normalization removes, while physical descriptors alone cannot capture the temporal dynamics that symbolic abstraction encodes. Neither stream is sufficient on its own; the two are complementary by construction.

Figure 6.1 illustrates the full pipeline. After signal conditioning and preprocessing (Section 6.2.2), each sensor window enters two parallel processing paths. The symbolic stream applies global z-normalization to the full signal before windowing, then encodes each window as a trend-aware SAX string (Section 6.2.3). The kinematics-informed stream operates on the conditioned but non-normalized signal, extracting window-level statistical descriptors that preserve the absolute physical magnitude normalization would otherwise erase (Section 6.2.3). The outputs of both streams are combined at the prompt construction stage (Section 6.2.4), and the resulting structured prompt is passed to the fine-tuned LLM for classification (Section 6.2.5).

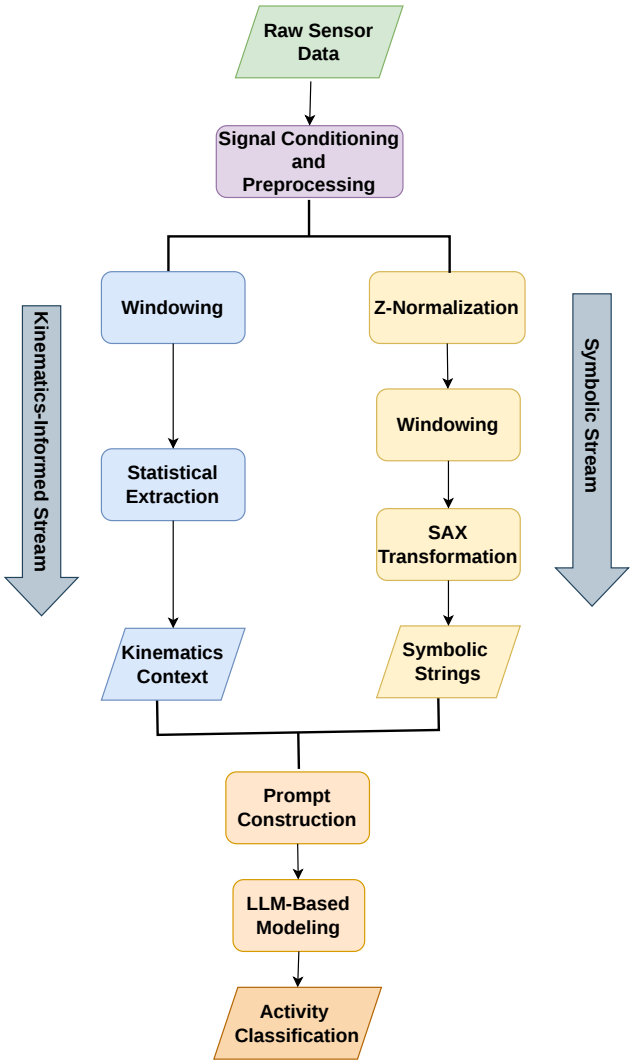


Figure 6.1: The dual-stream pipeline of the proposed methodology.

6.2.1 Problem Definition

The classification setting is MTS HAR. A multivariate time series of length T with D sensor channels is denoted

$$\boldsymbol{X} = \{x_t \in \mathbb{R}^D\}_{t=1}^{T} \tag{6.1}$$

where each observation x_t corresponds to synchronized readings from multiple inertial sensors (e.g., tri-axial accelerometer and gyroscope). A discrete activity label $y \in \mathcal{Y}$ is associated with each series, $\mathcal{Y}$ being a finite set of activity classes, and the goal is to learn a mapping $f : \mathcal{X} \to \mathcal{Y}$. Rather than classifying directly from the continuous signal, the approach taken here derives two representations from $\boldsymbol{X}$: an ordered symbolic sequence encoding temporal structure, and a set of physical descriptors preserving the absolute magnitude information that symbolic discretization removes. Both are used jointly as input to the sequence-level classifier.

6.2.2 Signal Conditioning and Preprocessing

Raw inertial signals carry noise and sampling irregularities that interfere with representation learning; preprocessing is therefore tailored to each dataset's native characteristics (see Section 6.3.1). WISDM signals are resampled to a uniform 20 Hz, while MotionSense observations are kept at their native 50 Hz. Both datasets are then denoised with a median filter (window size 3) followed by a second-order Butterworth low-pass filter at 5 Hz – a cutoff standard in HAR for separating human motion from high-frequency noise [217,218], and consistent with the filtering approach already used in Section 4.3.2.

Dataset-specific temporal cropping removes device-handling artifacts: the first 15 seconds of each WISDM recording are discarded to exclude start-up transients, while MotionSense trials are symmetrically cropped by 3 seconds at each end to remove pocketing and unpocketing effects.

Orientation variability across subjects and sessions is handled via the rWISDM alignment protocol [219]: a gravitational reference vector – estimated from the accelerometer mean for WISDM, or read from MotionSense's pre-computed device-motion attributes – is used to correct axis polarity and permute axes so that the dominant gravity component consistently aligns with the vertical Y-axis.

Finally, per-channel z-normalization is applied over the full signal to reduce inter-subject magnitude variability [220]. For a sensor channel $X = \{x_1, \ldots, x_n\}$, the normalized value is

$$z_i = \frac{x_i - \mu}{\sigma}, \tag{6.2}$$

with μ and σ the channel mean and standard deviation. A non-normalized copy of the conditioned signal is retained in parallel for the kinematics-informed stream (Section 6.2.3); the two aligned versions together form the input to the dual-stream pipeline.

6.2.3 Dual-Stream Representation Construction

Following conditioning, each activity instance is represented through a construction that separates symbolic temporal abstraction from physical magnitude preservation. Both streams are derived from the same windowed signal and processed in parallel, to be fused only at prompt construction.

Symbolic Stream: SAX Encoding

The symbolic stream encodes the temporal structure of each sensor signal in a discrete, structured form suitable for sequence-level modeling. Global z-normalization (Eq. (6.2)) is applied per channel prior to segmentation; the resulting signal is then segmented into overlapping sliding windows of fixed duration, and each window is transformed into a symbolic string.

Within each window, PAA (Section 2.2.1) reduces the n z-normalized sample points into m segment means $\bar{X} = \{\bar{x}_1, \ldots, \bar{x}_m\}$, exactly as in Eq. (2.1), applied to the normalized values z_j. Each coefficient is then mapped to a symbol s_i from an alphabet of size a using the Gaussian-breakpoint discretization already established in Section 2.2.2, yielding a standard SAX string $S = \{s_1, \ldots, s_m\}$ per channel per window.

Trend-Aware Symbolic Quantization. Standard SAX discretizes only the amplitude of each segment: the segment-wise averaging at the heart of PAA discards everything about how the signal evolves *within* a segment, the same lossy-compression weakness documented at length in Chapter 3 (Section 3.2) and already answered once in this thesis by SAA (Section 4.2). In human activity recognition, however, the *direction* of change within a segment is frequently as informative as its magnitude – a sharp upward slope and a gradual leveling can correspond to entirely different kinematic phases despite sharing a similar mean.

SAA closes this gap by computing a second, independently discretized angle series $\bar{\Theta}$ alongside the standard PAA output $\bar{X}$ (Section 4.2.1), carrying trend information in a structurally separate symbolic string that can be combined with the mean-based one only at the feature level. This chapter instead folds trend directly into the existing SAX symbol, rather than carrying it in a parallel string. For each segment i, the slope γ_i of the ordinary-least-squares line fitted to the n/m raw z_j sample points within that segment is computed via linear regression [221]. To suppress minor sensor noise rather than treat every small fluctuation as a meaningful trend, a threshold $\varepsilon = 0.05$ is used – corresponding to an angle of roughly 3° from the horizontal, i.e. a deliberately gentle requirement for a segment to be called anything other than flat. Formally, ε acts as a tolerance band around zero slope:

$$\text{trend}(\gamma_i) = \begin{cases} \text{up}, & \gamma_i > \varepsilon \\ \text{down}, & \gamma_i < -\varepsilon \\ \text{flat}, & |\gamma_i| \leq \varepsilon \end{cases} \tag{6.3}$$

The case of the symbol s_i already obtained from amplitude discretization is then modulated according to this trend, producing the trend-aware symbol $\tilde{s}_i$:

$$\tilde{s}_i = \begin{cases} \text{Uppercase}(s_i) & \text{if } \gamma_i > \varepsilon \quad \text{(up trend)} \\ \text{Lowercase}(s_i) & \text{otherwise} \quad \text{(flat or down)} \end{cases} \tag{6.4}$$

Where SAA carries trend in a separate string of the same length m as the mean-based one – doubling the number of symbolic positions but keeping each string's own alphabet intact and individually inspectable – case modulation instead doubles the effective alphabet size of a *single* string, from a to $2a$, without lengthening the sequence at all. This is precisely the trade-off anticipated in this chapter's opening: a more compact, single-string representation, at the cost of no longer being able to read

off trend independently of amplitude from the string alone. For a token sequence that must ultimately be serialized into an LLM prompt, where every additional symbol lengthens the context the model has to attend over, that compactness is the more relevant property – and it lets the model differentiate "high-energy rising" from "high-energy falling" states purely from symbolic context, without any increase in sequence length. Figure 6.2 illustrates the trend-aware SAX-encoding.

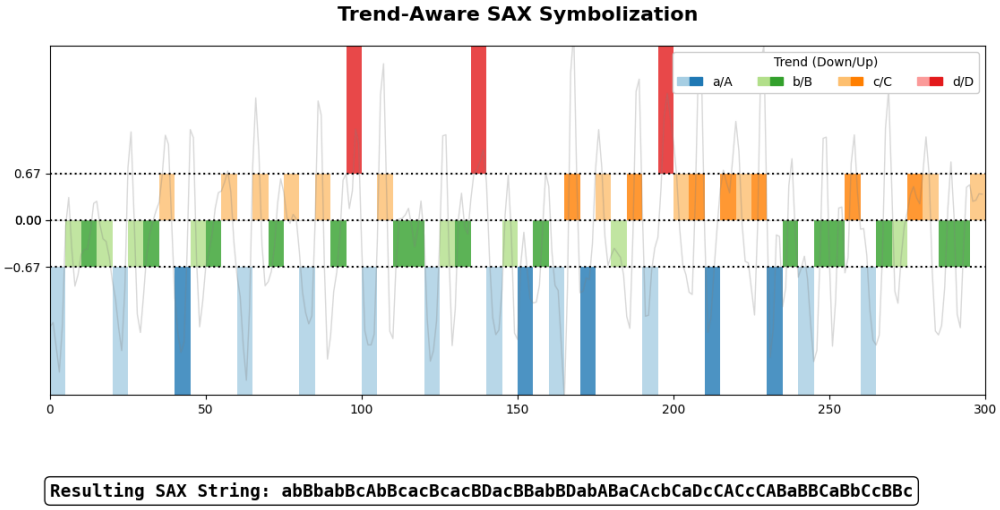


Figure 6.2: Trend-aware SAX transformation. Segments with a positive slope ($\gamma_i > \varepsilon$, increasing trend) are mapped to uppercase symbols and shown in darker tones; segments with a negative or near-zero slope are mapped to lowercase symbols and shown in lighter tones. The resulting string interleaves case and character to encode amplitude and direction jointly, without lengthening the sequence relative to standard SAX.

Kinematics-Informed Stream

In parallel with the symbolic stream, a kinematics-informed stream supplies complementary low-level descriptors directly from the raw sensor data. As noted in Section 6.2.2, the data have already undergone orientation normalization under the rWISDM protocol, aligning the accelerometer and gyroscope axes to a consistent physical reference frame in which the Y-axis captures the vertical gravitational component while the Z-axis signature remains available for posture differentiation.

To recover the physical context that symbolic discretization discards, three statistical descriptors [222] are extracted per analysis window directly from the accelerometer signal: from the gravity-aligned Y-axis, the mean acceleration μ_y and standard deviation σ_y; and from the Z-axis, the mean acceleration μ_z, capturing device inclination. The choice is physically grounded. μ_y quantifies the dominant gravitational component, establishing the device's vertical orientation; σ_y is a robust proxy for motion intensity, with high variability distinguishing dynamic activities such as jogging from static states [222]; and μ_z resolves the ambiguity between standing and sitting – both static and low-σ_y, but sitting induces a distinct gravitational shift onto the Z-axis from the thigh's horizontal orientation, a distinction the symbolic stream alone cannot make.

Critically, these statistics are computed per window on the raw, orientation-

normalized signal, *before* the global standardization applied for the symbolic stream: the symbolic stream uses data standardized over the full series, while the kinematics-informed descriptors are derived from the non-normalized values, so that absolute gravitational offset and motion magnitude survive into the descriptors that explicitly compensate for what symbolic discretization removes.

6.2.4 Prompt Construction

To let the LLM operate as a structured sequence model for activity recognition, each sensor window is reformulated as a structured textual prompt [223] that integrates global physical context with local temporal dynamics, rather than as a raw numerical tensor – encouraging the model to process physical constraints hierarchically before interpreting the symbolic motion pattern itself.

Each prompt encodes one fixed-length sensor window as a structured text sequence with three components: a brief task instruction, the window-level physical descriptors from the kinematics-informed stream, and the trend-aware SAX strings for each accelerometer and gyroscope axis (Section 6.2.3). Components are labeled explicitly by modality and axis, so the model can integrate numerical physical context and symbolic temporal patterns within a single coherent input. The prompt terminates with a class delimiter; during fine-tuning the activity label is provided only as the target completion, keeping the conditioning input and the supervision signal cleanly separated. An example is shown below.

```
{"prompt":  "Classify the activity based on the sensor data.
Stats:  Gravity-Y=9.54, Gravity-Z=0.39, Intensity=2.99
Accel-X: ABdbABdcBBdcABdcABdcABdcABdcABdcABdcABdcABDbBCDcaCdcBBDaABDbaCDa
Accel-Y: CBCbCBCbCBCcCBCcCBCcBBCbBbCcBcCcBbCbBACbCbCbBbCbBBCbCcCaCbDa
Accel-Z: DbBbdACbDBCbcABbcBcbDBcbDCcbDbBcDDcbDAcbDABbDCcaDAbbcACbdDbb
Gyro-X: cCbAcCcAcCcAcCcAcCcAcCcAcCcAcCcAcCcAcCcAcDbAcDbAcCbAcDbAcDbBcDaB
Gyro-Y: bABDcaADbaBCcACCcACDcACDcABCcABCcABDcbBCcABDcaBDcABDcABDbABD
Gyro-Z: caBdcABdbaBdcaBdbaBdcABdcaBdcaBdcABdcAbdbABdcABdbABdbABdcACd
### Class:",
"completion":  " Walking"}
```

6.2.5 LLM Fine-Tuning and Inference

The following subsections detail how activity recognition is formulated as a prompt-completion task and how the model is trained and deployed.

LLM Formulation and Prompt Design

A Mistral-7B causal language model [224] is fine-tuned via supervised instruction tuning on the structured prompt–completion pairs of Section 6.2.4. Mistral-7B was selected for its architectural efficiency and long-context modeling: it employs Sliding Window Attention to handle sequences of arbitrary length at reduced inference cost, and Grouped-Query Attention for faster inference, and has been shown to outperform larger counterparts such as Llama 2 13B across evaluated benchmarks [224]. Fine-tuning uses parameter-efficient adaptation to preserve the base model's general linguistic and sequential modeling capacity while specializing it for activity recognition. Each training example concatenates a prompt terminating in the class delimiter with a completion containing the ground-truth label, and the model is optimized to maximize

the likelihood of the completion tokens conditioned on the prompt – casting activity recognition as conditional next-token prediction [202], with no task-specific classifier head required.

We stress that the Mistral-7B tokenizer treats uppercase and lowercase symbols as entirely independent tokens with separate embeddings, so the intended relationship between a rising symbol and its flat-or-falling counterpart – say `A` and `a` – is not structurally encoded at the tokenization level. The case distinction is applied consistently across all training examples, however, so the model can acquire the association statistically during fine-tuning even without an explicit structural grounding.

Fine-Tuning Procedure and Inference

Figure 6.3 summarizes the fine-tuning framework: structured prompts derived from the dual-stream representation condition a frozen, 4-bit quantized LLM, while only the lightweight LoRA adapter parameters are updated during training.

Concretely, fine-tuning uses QLoRA [225], combining 4-bit NormalFloat (NF4) quantization with double quantization and bfloat16 computation to minimize memory overhead while preserving numerical fidelity. LoRA modules [226] are injected into all attention and feed-forward projection layers with rank $r = 32$, scaling factor $\alpha = 16$, and dropout 0.1; biases remain frozen throughout. Training is executed via the TRL SFTTrainer [227] using the paged AdamW optimizer [225] with learning rate 2×10^{-4}, weight decay 0.001, warm-up ratio 0.03, and gradient clipping 0.3.

The number of training epochs is determined per dataset by monitoring validation accuracy on a held-out 20% of training windows over up to 15 epochs. The model is then retrained on the complete training set using the selected epoch count. The full hyperparameter configuration is summarized in Table 6.2; all experiments ran on a workstation with an AMD Ryzen Threadripper PRO 7975WX (32 cores, 64 threads), 512 GB RAM, and an NVIDIA RTX 6000 Ada (48 GB VRAM).

At inference time, a prompt is constructed for each test window in the same structured format used during training – trend-aware SAX strings, kinematics-informed descriptors, and the terminating class delimiter – but without an activity label. The quantized base model is loaded with the fine-tuned LoRA adapter weights and generates a completion using greedy decoding (`do_sample = False`) with at most 10 new tokens, sufficient to cover every activity name in both datasets. The prediction is extracted from the first generated line following the delimiter, guarding against hallucinated continuations and establishing a direct one-to-one mapping between prompt and predicted label.

6.2.6 Attention-Based Interpretability Analysis

The prompt-completion formulation allows attention patterns to be inspected at inference time without modifying the model or adding an external explanation mechanism, providing a window into which symbolic motifs and physical descriptors most influence the predicted label. The analysis is descriptive and exploratory throughout: it characterizes where the model concentrates predictive weight, not what causally determines its output.

Attention scores are taken from the final transformer layer, where classification-relevant information is most consolidated, and averaged across all attention heads for a stable per-token estimate. Two stages are defined. Stage 1 queries the final prompt token: in a causal language model this position aggregates the entire input

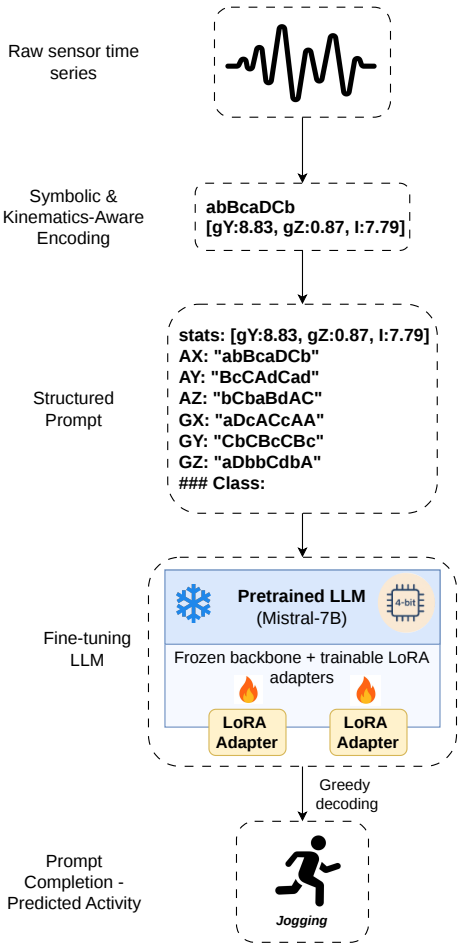


Figure 6.3: Overview of the fine-tuning framework. Raw inertial signals are transformed into the dual-stream symbolic and physical descriptors used to construct prompts; these condition a frozen, 4-bit quantized LLM adapted via trainable LoRA adapters to predict activity labels.

context and directly gates next-token prediction, making it the natural point at which to measure overall input integration. Stage 2 queries the first generated label token, capturing which prompt elements the model attended to at the moment it committed to a prediction.

All scores are normalized by the uniform attention baseline $1/L$, where L is the prompt length in tokens, so that a score of $k \times (1/L)$ means the token received k times more attention than a uniform distribution would assign. This makes scores comparable across prompts and samples of different lengths.

Two levels of analysis follow from these scores. At the token-group level, prompt tokens are split into kinematics-informed descriptors (numeric values) and symbolic SAX tokens (alphabetic characters); mean attention density is computed and normalized per group at each stage to quantify the relative weight the model places on physical context versus symbolic dynamics. At the motif level, the three most-attended

Table 6.2: Hyperparameter configuration for supervised fine-tuning (SFT) via QLoRA. The epoch count refers to the WISDM dataset.

Hyperparameter	**Value**
Optimization Strategy	
Method	QLoRA (4-bit NF4)
Optimizer	Paged AdamW (32-bit)
Precision	bfloat16
LoRA Configuration	
Rank	32
Scaling Factor	16
Target Modules	Attention + Feed-Forward
Dropout	0.1
Training Schedule	
Learning Rate	2×10^{-4}
Scheduler	Constant
Epochs	12
Batch Size	64
Warm-up Ratio	0.03
Gradient Clipping	0.3

SAX tokens within each sensor axis's span are identified from Stage 2 scores, and a fixed-length motif of 5 symbols is extracted around each. Every motif is reported with two separate measures – Stage 2 attention as a ×baseline multiplier and frequency of occurrence across samples – kept distinct rather than combined, so that attention strength and motif stability can be read independently.

6.3 Experiments

6.3.1 Datasets

The proposed methodology is evaluated on two benchmark inertial HAR datasets, selected to test generalization across sensor placement, sampling frequency, and activity vocabulary.

The WISDM Smartphone and Smartwatch Activity and Biometrics Dataset [228] is a widely used HAR benchmark collected from 51 subjects. We focus on a subset of five core activities – *Walking*, *Jogging*, *Stairs*, *Sitting*, and *Standing* – capturing fundamental locomotion and posture-related motion, and commonly used in smartphone-based activity recognition. Sensor data is drawn exclusively from the smartphone, carried in the subject's pocket, using both the accelerometer and gyroscope, each providing triaxial measurements along the x, y, and z axes. Each activity recording in the

selected subset lasts 180 seconds at approximately 20 Hz, yielding roughly 3,600 samples per axis; smartwatch and hand-oriented activities are excluded. The selected activities are approximately class-balanced in the original dataset, each accounting for roughly 5–6% of total recordings, and all readings are annotated with activity labels and subject identifiers, enabling subject-independent evaluation.

We additionally evaluate on MotionSense [229], a smartphone-based HAR benchmark collected from 24 subjects performing six activities. We use the standard subset comprising *Downstairs*, *Upstairs*, *Walking*, *Jogging*, *Standing*, and *Sitting*, capturing both locomotion and posture dynamics. Sensor data includes both accelerometer and gyroscope modalities from an iPhone 6s carried in the front pocket, with triaxial measurements along the x, y, and z axes sampled at 50 Hz.

6.3.2 Preprocessing and Segmentation

Following Section 6.2.2's conditioning pipeline, both datasets are brought into a unified format for sequence modeling using a single segmentation protocol: sliding windows of $n = 300$ samples with 50% overlap (a 150-sample stride). Because WISDM and MotionSense are sampled at different native rates, a fixed sample count translates into different window durations T – 15 seconds for WISDM and 6 seconds for MotionSense – while keeping the symbolic representation's sequence length and vocabulary identical across both datasets, regardless of the original sampling frequency. Each window is compressed into $m = 60$ PAA segments, representing 20% of the window length, and each segment is discretized into one of $a = 4$ base symbols, with the trend-aware case modulation of Section 6.2.3 doubling the effective vocabulary to 8 symbols ($2a$) without increasing the sequence length m. Table 6.3 summarizes these parameters alongside the resulting window and subject counts after segmentation.

A subject-independent evaluation protocol is used throughout to assess generalization to unseen users: subjects are randomly partitioned into training and test sets via an 80/20 split, with all samples from a given subject appearing exclusively in one set, and this split fixed across every experiment to ensure reproducibility and prevent subject-specific motion patterns from leaking into evaluation. The number of training epochs is determined independently per dataset via a validation-based early-stopping procedure, described in full in Section 6.2.5. On WISDM, accuracy peaks at epoch 12 (95.68%) and degrades consistently thereafter (Figure 6.4); on MotionSense, the corresponding epoch is 8. Performance is measured at the window level, with each window yielding a single activity prediction. Overall classification accuracy is reported as the primary metric, complemented by per-class precision, recall, F1-score, and confusion matrices [230] to analyze systematic confusions between activities with similar motion characteristics.

6.3.3 Baselines

SAX_HAR-LLM is compared against a deliberately heterogeneous set of baselines, spanning symbolic, deep learning, kernel-based, and sequence-modeling families, so that any advantage can be attributed to a specific design choice rather than to an unmatched comparison.

Symbolic baselines. BOP and its multichannel variant, that is BOP augmented with MII (Section 2.2.3, Section 4.1) rely on the same SAX-based abstraction as the proposed method, discretizing each window with a SAX word length of 3 and alphabet size of 4 – a parameter choice balancing structural granularity against

Table 6.3: Preprocessing and segmentation parameters, and dataset statistics after segmentation.

Parameter	**WISDM**	**MotionSense**
Sampling Frequency	20 Hz	50 Hz
Conditioning Crop	15 s (start only)	3 s (symmetric)
Low-pass Filter	Butterworth 2^{nd}-order (5 Hz cutoff)	
Windowing Protocol		
Window Size (n, samples)	300	
Window Duration (T)	15 s	6 s
Stride / Overlap	150 samples / 50%	
Dataset Statistics after Segmentation		
Training Windows	3,243	5,340
Validation Windows	810	1,334
Test Windows	1,155	1,809
Total Windows	5,208	8,483
Training Subjects	40	19
Test Subjects	11	5
Feature Extraction		
PAA Segments (m)	60 (20% of window)	
Alphabet Size (a)	4 (effective 8 with trend encoding)	
Trend Threshold (ε)	0.05	
Normalization	Z-normalization over full signal (symbolic stream only)	
Kinematics Descriptors	Extracted from non-normalized signal	

feature sparsity, since increasing either would inflate pattern-histogram dimensionality and degrade classifier performance through the curse of dimensionality [63], and one consistent with prior published work using this same parameterization [61,106,165,231]. Symbolic words are aggregated into frequency histograms and classified with a Random Forest [176] of 100 trees, trained on the training subjects and evaluated on held-out test subjects, mirroring exactly the classifier used for MII and SAA evaluation in Chapter 4.

Deep learning and kernel-based baselines. ResNet [98], InceptionTime [99], and MultiROCKET [95], a kernel-based method are deployed here as strong numerical baselines. DeepConvLSTM [232] integrates convolutional feature extraction with LSTM layers specifically to bridge spatial sensor features and longer-range temporal context, and is included as a HAR-specific deep learning baseline.

Sequence-modeling references. A Vanilla Transformer [25] is trained from scratch on inputs identical to those used for SAX_HAR-LLM, with every training hy-

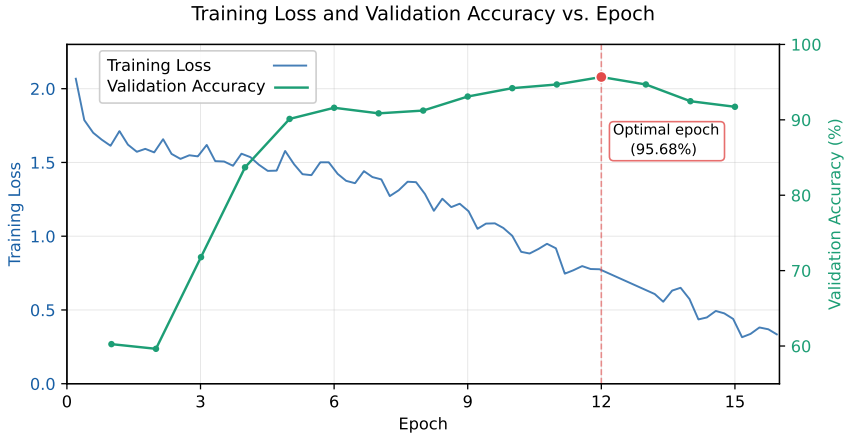


Figure 6.4: Training loss and validation accuracy across 15 training epochs for the WISDM dataset. Validation accuracy peaks at epoch 12 (95.68%) and degrades consistently afterward, motivating epoch 12 as the selected stopping point.

perparameter deliberately matched to the SAX_HAR-LLM configuration (Table 6.2), isolating the presence or absence of pre-trained weights as the only variable between the two. Its full architectural specification is given in Table 6.4. A RoBERTa model, fine-tuned with a sequence-classification head on inputs identical to SAX_HAR-LLM's, is included to isolate the contribution of the autoregressive, prompt-completion architecture from large-scale pre-training itself: it shares SAX_HAR-LLM's input representation and pre-training scale but not its causal, generation-based formulation. Finally, LLM-ABBA [204], adapted to this setting, serves as the closest published symbolic-LLM method and the most direct point of comparison for the central claim that a trend-aware symbolic representation, rather than a generic adaptive symbolic aggregation, is what an LLM backbone benefits from.

For SAX_HAR-LLM itself, the fine-tuned Mistral-7B model (Section 6.2.5) is evaluated using the identical subject-independent partition, preprocessing, and windowing strategy as every baseline, ensuring that all comparisons differ only in representation and architecture, never in data split or windowing. Inference uses greedy decoding for deterministic, reproducible predictions: for each test window, the model completes the structured prompt, and the text immediately following the class delimiter is mapped directly to the activity label set, producing exactly one prediction per window and enabling direct comparison with every symbolic, deep-learning, and sequence-modeling baseline under identical conditions.

6.3.4 Setup Summary

Table 6.5 consolidates the dataset specifications, evaluation protocol, and model configuration described above into a single reference summary.

6.4 Results

The presentation of the results proceeds from main quantitative results, to a paired statistical comparison, an ablation study, attention and motif analysis, and finally a computational cost comparison. Because the pre-trained backbone remains frozen and

Table 6.4: Hyperparameter configuration for the Vanilla Transformer baseline. The epoch count refers to the WISDM dataset.

Hyperparameter	**Value**
Model Architecture	
Number of Layers	3
Attention Heads	4
Embedding Dimension	128
FFN Dimension	256
Dropout	0.1
Trainable Parameters	563,845
Optimization Strategy	
Optimizer	AdamW
Precision	bfloat16
Training Schedule	
Learning Rate	2×10^{-4}
Scheduler	Constant
Epochs	12
Batch Size	64
Warm-up Ratio	0.03
Gradient Clipping	0.3
Weight Decay	0.001

inference relies on deterministic greedy decoding, performance variance across runs is minimal despite the inherent stochasticity of adapter training.

6.4.1 Main Classification Results

Table 6.6 and Table 6.7 report class-wise precision (P), recall (R), and F1-score (F1) for SAX_HAR-LLM and every baseline, together with overall classification accuracy, on WISDM and MotionSense respectively.

On WISDM, SAX_HAR-LLM achieves 92.29% overall accuracy, exceeding the symbolic baselines by a meaningful margin – BOP at 83.72% and BOP+MII at 86.67% – which is consistent with the LLM backbone exploiting sequential dependencies within the discretized token string that a bag-of-words histogram cannot represent. Among the remaining baselines, SAX_HAR-LLM surpasses ResNet (86.49%), InceptionTime (85.97%), DeepConvLSTM (83.55%), the Vanilla Transformer (78.01%), LLM-ABBA (61.04%), and RoBERTa (90.82%); MultiROCKET alone ranks higher at 95.50%. Per-class results reveal a clear split: rhythmic activities – *Jogging*, *Walking*, and *Stairs* – are classified with F1-scores between 0.96 and 0.97, while static postures

Table 6.5: Experimental setup, model configuration, and evaluation protocol.

Component	**Configuration**
Dataset Specifications	
Subject Count	WISDM: 51 | MotionSense: 24
Sensor Modalities	Smartphone accelerometer & gyroscope (tri-axial)
Activities (WISDM)	Walking, Jogging, Stairs, Sitting, Standing
Activities (MotionSense)	Walking, Jogging, Upstairs, Downstairs, Sitting, Standing
Evaluation Protocol	
Evaluation Level	Window-level classification
Split Strategy	Subject-independent (80% train / 20% test)
Metrics	Accuracy, Precision, Recall, Macro-F1
Model & Fine-Tuning	
LLM Backbone	Mistral-7B-v0.1
Training Method	QLoRA (4-bit quantization, LoRA adapters)
Baselines	BOP, BOP+MII, ResNet, InceptionTime, MultiROCKET, DeepConvLSTM, Vanilla Transformer, LLM-ABBA, RoBERTa
Inference	Prompt completion with greedy decoding

prove harder, with *Sitting* and *Standing* at F1 0.86 and 0.87 respectively. This is a known limitation of amplitude-invariant symbolic representations, and the confusion matrix (Figure 6.5) confirms it: substantial reciprocal errors between the two static classes persist even with the kinematics-informed descriptors included specifically to resolve their orientation ambiguity.

Results on MotionSense reinforce this picture: SAX_HAR-LLM reaches 91.71% accuracy, well above the symbolic baselines (BOP at 78.28%, BOP+MII at 80.93%), ResNet (86.07%), the Vanilla Transformer (83.58%), and RoBERTa (90.22%), the latter receiving identical inputs. The model is especially strong on rhythmic movement, with F1-scores of 1.00 for *Walking* and 0.99 for *Jogging*, and notably outperforms MultiROCKET on *Upstairs* (F1: 0.98 vs. 0.90), suggesting the trend-aware symbolic encoding is particularly effective at capturing the distinct signature of vertical ascending movement. The clearest weakness is *Downstairs* (F1: 0.64), trailing InceptionTime (0.82) and MultiROCKET (0.84): the confusion matrix (Figure 6.5) shows that while most genuine descent instances are correctly identified, the class suffers a high false-positive rate from static activities – 81 *Sitting* and 32 *Standing* instances are misclassified as *Downstairs* – alongside some genuine *Downstairs* samples misread as *Standing*. The adapted LLM-ABBA baseline drops sharply on this dataset, from 61.04% on WISDM to 29.30% here, plausibly reflecting the larger six-class vocabulary and greater class imbalance relative to WISDM's five activities.

Table 6.6: Per-class precision (P), recall (R), and F1-score (F1), together with overall accuracy (%), on the WISDM dataset for SAX_HAR-LLM and all baseline methods.

Model	Jogging	Sitting	Stairs	Standing	Walking	Accuracy (%)
SAX_HAR-LLM	P: 1.00 R: 0.94 F1: 0.97	P: 0.89 R: 0.83 F1: 0.86	P: 0.92 R: 1.00 F1: 0.96	P: 0.84 R: 0.90 F1: 0.87	P: 0.98 R: 0.95 F1: 0.96	92.29
BOP	P: 0.97 R: 0.99 F1: 0.98	P: 0.71 R: 0.61 F1: 0.66	P: 0.87 R: 0.96 F1: 0.91	P: 0.66 R: 0.73 F1: 0.69	P: 0.99 R: 0.90 F1: 0.94	83.72
BOP+MII	P: 0.95 R: 0.99 F1: 0.97	P: 0.82 R: 0.69 F1: 0.73	P: 0.87 R: 0.96 F1: 0.91	P: 0.72 R: 0.83 F1: 0.77	P: 1.00 R: 0.90 F1: 0.95	86.67
ResNet	P: 1.00 R: 0.92 F1: 0.96	P: 0.82 R: 0.72 F1: 0.77	P: 0.80 R: 1.00 F1: 0.89	P: 0.76 R: 0.84 F1: 0.80	P: 1.00 R: 0.84 F1: 0.92	86.49
InceptionTime	P: 1.00 R: 0.99 F1: 0.99	P: 0.80 R: 0.72 F1: 0.76	P: 0.82 R: 0.97 F1: 0.89	P: 0.75 R: 0.82 F1: 0.78	P: 0.96 R: 0.81 F1: 0.88	85.97
MultiROCKET	P: 1.00 R: 0.98 F1: 0.99	P: 0.89 R: 0.93 F1: 0.91	P: 0.97 R: 1.00 F1: 0.98	P: 0.93 R: 0.88 F1: 0.90	P: 1.00 R: 0.98 F1: 0.99	**95.50**
DeepConvLSTM	P: 0.87 R: 0.99 F1: 0.93	P: 0.87 R: 0.69 F1: 0.77	P: 0.82 R: 0.91 F1: 0.86	P: 0.75 R: 0.91 F1: 0.82	P: 0.90 R: 0.68 F1: 0.77	83.55
Vanilla Transformer	P: 0.99 R: 0.92 F1: 0.95	P: 0.86 R: 0.78 F1: 0.82	P: 0.63 R: 0.70 F1: 0.66	P: 0.80 R: 0.87 F1: 0.83	P: 0.66 R: 0.64 F1: 0.65	78.01
LLM-ABBA	P: 0.76 R: 0.96 F1: 0.85	P: 0.50 R: 0.02 F1: 0.03	P: 0.96 R: 0.39 F1: 0.55	P: 0.50 R: 0.98 F1: 0.66	P: 0.53 R: 0.71 F1: 0.60	61.04
RoBERTa	P: 1.00 R: 0.97 F1: 0.98	P: 0.84 R: 0.82 F1: 0.83	P: 0.93 R: 0.96 F1: 0.94	P: 0.81 R: 0.87 F1: 0.84	P: 0.99 R: 0.93 F1: 0.96	90.82

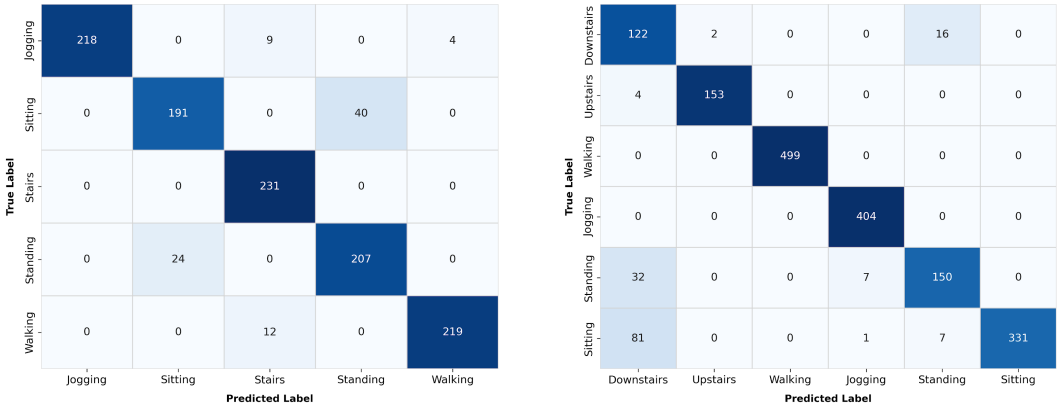
6.4.2 Statistical Comparison

To test whether the accuracy differences in Tables 6.6 and 6.7 are statistically meaningful, SAX_HAR-LLM is compared against MultiROCKET, InceptionTime, and RoBERTa using three tests. A McNemar test [233] is applied at the window level, but is anticonservative in this setting: the 50% stride produces overlapping, subject-clustered windows that violate the independence assumption, so its p-values are treated as indicative only.

The window-level analysis is therefore complemented by two subject-level tests, which treat subjects rather than windows as the unit of observation. A paired permutation test assesses whether the observed mean accuracy difference across subjects could arise by chance, using 10,000 random sign-flips under the null hypothesis of no systematic effect. A bootstrap confidence interval independently quantifies how stable that difference would be across hypothetical redraws from the same population, resampling subjects with replacement across 10,000 iterations. Both tests operate

Table 6.7: Per-class precision (P), recall (R), and F1-score (F1), together with overall accuracy (%), on the MotionSense dataset for SAX_HAR-LLM and all baseline methods.

Model	Downstairs	Upstairs	Walking	Jogging	Standing	Sitting	Accuracy (%)
SAX_HAR-LLM	P: 0.51 R: 0.87 F1: 0.64	P: 0.99 R: 0.97 F1: 0.98	P: 1.00 R: 1.00 F1: 1.00	P: 0.98 R: 1.00 F1: 0.99	P: 0.87 R: 0.79 F1: 0.83	P: 1.00 R: 0.79 F1: 0.88	91.71
BOP	P: 0.52 R: 0.70 F1: 0.59	P: 0.91 R: 0.61 F1: 0.73	P: 0.84 R: 0.85 F1: 0.84	P: 0.92 R: 0.94 F1: 0.93	P: 0.70 R: 0.85 F1: 0.77	P: 0.86 R: 0.71 F1: 0.78	78.28
BOP+MII	P: 0.52 R: 0.71 F1: 0.60	P: 0.91 R: 0.60 F1: 0.72	P: 0.83 R: 0.85 F1: 0.84	P: 0.94 R: 0.92 F1: 0.93	P: 0.74 R: 0.92 F1: 0.82	P: 0.92 R: 0.76 F1: 0.83	80.93
ResNet	P: 0.45 R: 0.80 F1: 0.58	P: 0.64 R: 0.86 F1: 0.73	P: 0.93 R: 0.63 F1: 0.75	P: 1.00 R: 0.74 F1: 0.85	P: 0.99 R: 1.00 F1: 1.00	P: 1.00 R: 1.00 F1: 1.00	86.07
InceptionTime	P: 0.82 R: 0.82 F1: 0.82	P: 0.89 R: 0.75 F1: 0.82	P: 1.00 R: 0.96 F1: 0.98	P: 0.97 R: 0.98 F1: 0.98	P: 0.90 R: 1.00 F1: 0.95	P: 1.00 R: 1.00 F1: 1.00	94.91
MultiROCKET	P: 0.84 R: 0.84 F1: 0.84	P: 0.89 R: 0.92 F1: 0.90	P: 0.99 R: 0.99 F1: 0.99	P: 1.00 R: 0.96 F1: 0.98	P: 1.00 R: 1.00 F1: 1.00	P: 1.00 R: 1.00 F1: 1.00	**97.40**
DeepConvLSTM	P: 0.48 R: 0.67 F1: 0.56	P: 0.85 R: 0.81 F1: 0.83	P: 0.90 R: 0.76 F1: 0.82	P: 0.78 R: 0.88 F1: 0.83	P: 1.00 R: 1.00 F1: 1.00	P: 1.00 R: 1.00 F1: 1.00	88.78
Vanilla Transformer	P: 0.61 R: 0.44 F1: 0.51	P: 0.49 R: 0.37 F1: 0.42	P: 0.68 R: 0.85 F1: 0.76	P: 0.95 R: 0.94 F1: 0.95	P: 0.98 R: 0.97 F1: 0.98	P: 1.00 R: 0.96 F1: 0.98	83.58
LLM-ABBA	P: 0.29 R: 0.69 F1: 0.41	P: 0.84 R: 0.27 F1: 0.41	P: 0.13 R: 0.30 F1: 0.18	P: 0.46 R: 0.98 F1: 0.63	P: 0.78 R: 0.25 F1: 0.38	P: 0.00 R: 0.00 F1: 0.00	29.30
RoBERTa	P: 0.62 R: 0.89 F1: 0.73	P: 0.91 R: 0.76 F1: 0.83	P: 0.97 R: 0.78 F1: 0.86	P: 0.67 R: 0.96 F1: 0.79	P: 1.00 R: 0.99 F1: 0.99	P: 1.00 R: 0.97 F1: 0.98	90.22



(a) WISDM dataset (b) MotionSense dataset

Figure 6.5: Confusion matrices for SAX_HAR-LLM on both benchmark datasets.

under limited power – 11 subjects on WISDM, 5 on MotionSense – so a non-significant

Table 6.8: Statistical comparison between SAX_HAR-LLM and selected baselines on WISDM (11 test subjects, 1,155 windows) and MotionSense (5 test subjects, 1,809 windows). McNemar p-values are indicative only, given overlapping windows and subject clustering. The subject-level permutation test and bootstrap 95% CI treat subjects as the unit of observation. A positive mean difference indicates SAX_HAR-LLM performs better. Statistical significance is assessed at $alpha = 0.05$.

Dataset	Baseline	Mean Diff (pp)	McNemar p-value	Permutation p-value	Bootstrap 95% CI (pp)
WISDM	MultiROCKET	−3.20	< 0.001	0.159	[−7.27, +0.43]
	InceptionTime	+6.32	< 0.001	0.057	[+1.13, +13.25]
	RoBERTa	+1.47	0.075	0.373	[−1.65, +4.24]
MotionSense	MultiROCKET	−4.99	< 0.001	0.061†	[−11.85, −0.78]
	InceptionTime	−2.82	< 0.001	0.189†	[−6.94, −0.05]
	RoBERTa	+1.27	0.004	0.375†	[−0.57, +3.11]

† MotionSense has only 5 test subjects, yielding $2^5 = 32$ distinct sign assignments; permutation p-value resolution is limited to multiples of $1/32 \approx 0.031$ and should be interpreted with caution.

permutation result should be read as inconclusive rather than as evidence of equivalence. On MotionSense in particular, the small subject pool means individual subjects exert disproportionate influence on the bootstrap CI. Full results are in Table 6.8.

The results in Table 6.8 tell a coherent story across both datasets. On WISDM, permutation tests return no significant differences, but 11 subjects provide little power, so absence of significance here is not absence of effect. The bootstrap CIs are more informative: SAX_HAR-LLM holds a reliable edge over InceptionTime ([+1.13, +13.25] pp), MultiROCKET leads consistently ([−7.27, +0.43] pp, upper bound near zero), and the comparison with RoBERTa ($p = 0.373$, CI [−1.65, +4.24] pp) points to practical equivalence. On MotionSense, permutation testing is constrained by the $2^5 = 32$ resolution limit, but the bootstrap CIs again provide direction: MultiROCKET ([−11.85, −0.78] pp) and InceptionTime ([−6.94, −0.05] pp) both show intervals excluding zero against SAX_HAR-LLM, with the gap traceable largely to the *Downstairs* class (F1: 0.64). The RoBERTa comparison ([−0.57, +3.11] pp) produces the tightest CI in the entire analysis, reinforcing near-equivalence across both benchmarks. The overall picture is that SAX_HAR-LLM is competitive with strong numerical methods, statistically similar with RoBERTa, and uniquely positioned among them in offering full representational transparency.

6.4.3 Ablation Study

The proposed SAX_HAR-LLM design (Section 6.2.3) combines (i) a symbolic stream encoding discretized temporal patterns via trend-aware symbolic quantization, and (ii) a kinematics-informed stream capturing axis-specific descriptors, with both streams jointly processed by the LLM. To isolate the contribution of each component, a stepwise ablation study compares the full model against two reduced variants:

Ablation_1 (Symbolic Only): relies solely on SAX token sequences, without the kinematics-informed descriptors.

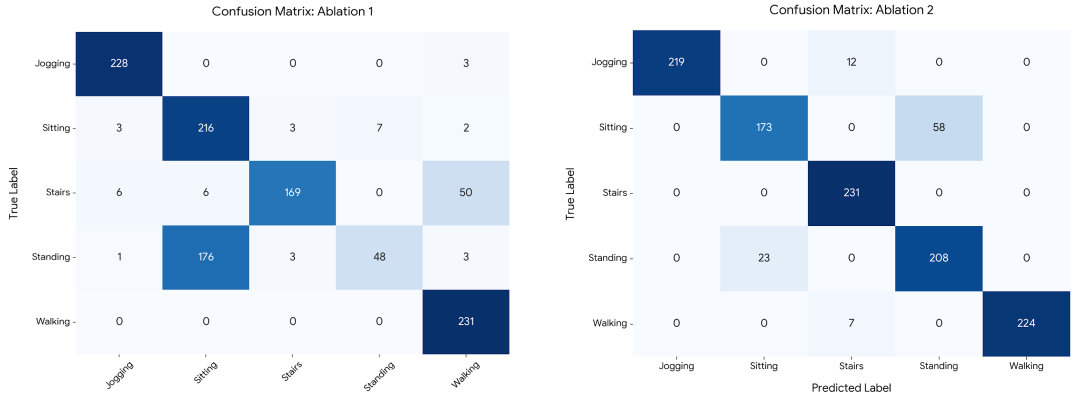
Ablation_2 (No Trend-Aware): receives standard SAX sequences without the

case-based slope encoding of Section 6.2.3, together with the kinematics-informed descriptors.

All variants are evaluated on WISDM under the same protocol, data splits, and metrics as the main comparison. Per-class metrics and confusion matrices are reported in Table 6.9 and Figure 6.6. The full model reaches 92.29% accuracy; removing the kinematics-informed descriptors (Ablation_1) drops this to 77.23%, while removing the trend-aware encoding (Ablation_2) reduces it more modestly to 91.34%. The sharp drop in Ablation_1 points to the kinematics-informed stream as a critical component, while the smaller degradation in Ablation_2 reflects a more targeted contribution from the case-modulation encoding.

Table 6.9: Ablation study results: per-class precision (P), recall (R), and F1-score (F1), together with overall accuracy (%), on WISDM.

Model	Jogging	Sitting	Stairs	Standing	Walking	Accuracy (%)
Ablation_1	P: 0.96 R: 0.99 F1: 0.97	P: 0.54 R: 0.94 F1: 0.69	P: 0.97 R: 0.73 F1: 0.83	P: 0.87 R: 0.21 F1: 0.34	P: 0.80 R: 1.00 F1: 0.89	77.23
Ablation_2	P: 1.00 R: 0.95 F1: 0.97	P: 0.88 R: 0.75 F1: 0.81	P: 0.92 R: 1.00 F1: 0.96	P: 0.78 R: 0.90 F1: 0.84	P: 1.00 R: 0.97 F1: 0.98	91.34
SAX_HAR-LLM	P: 1.00 R: 0.94 F1: 0.97	P: 0.89 R: 0.83 F1: 0.86	P: 0.92 R: 1.00 F1: 0.96	P: 0.84 R: 0.90 F1: 0.87	P: 0.98 R: 0.95 F1: 0.96	**92.29**



(a) Symbolic-only stream (b) No trend-aware symbolization

Figure 6.6: Confusion matrices for the ablation variants on WISDM.

Ablation_1 isolates what happens when an LLM processes SAX sequences without any physical grounding. Dynamic activities remain well-handled: nearly all *Jogging* (228/231) and *Walking* (231/231) instances are correctly classified, since their rhythmic, high-energy motion patterns survive symbolic discretization. Static postures, however, collapse: *Standing* recall falls to 20.8% (48/231), with 176 instances misclassified as *Sitting*. The cause is straightforward – z-normalization removes absolute amplitude, leaving both postures as low-variance symbolic sequences that are indistinguishable

in the symbolic domain. Without gravitational context to separate them, the model defaults to *Sitting* as the more frequently seen static class during training. The kinematics-informed stream therefore resolves static-posture ambiguity without compromising the model's ability to exploit temporal patterns in dynamic activities – confirming that the two streams are complementary by design, not redundant.

Reintroducing the kinematics-informed descriptors preserves dynamic activity recognition rather than trading it off: *Jogging* and *Walking* F1-scores remain at 0.97 and 0.96 in the full model, showing that the kinematics-informed stream resolves the static-posture ambiguity without degrading the model's ability to exploit strong temporal patterns in periodic motion – evidence that the dual-stream design is complementary rather than redundant.

Ablation_2 isolates a more targeted effect. Removing trend-aware encoding produces a modest but consistent accuracy decrease concentrated specifically on the *Sitting*/*Standing* boundary, indicating that the case-based slope encoding captures subtle local directional changes that matter precisely for separating quasi-static activities with similar amplitude profiles. Dynamic activities – *Walking*, *Jogging*, *Stairs* – remain largely unaffected, confirming that trend-aware encoding closes a specific representational gap rather than delivering a general performance boost – the same gap Section 6.2.3 motivated by appeal to Chapter 3's taxonomy and Chapter 4's SAA.

6.4.4 Attention Distribution and Motif Analysis

This subsection applies the two-stage attention inspection of Section 6.2.6 to the fine-tuned model's predictions on WISDM. As established there, all findings below are descriptive and exploratory: they characterize patterns of predictive focus rather than establishing causal feature importance.

Token-group attention density. Table 6.10 reports normalized attention densities assigned to kinematics-informed and symbolic tokens at both stages. At Stage 1, attention is relatively balanced across token types for every activity, with SAX tokens receiving roughly 58–64% and kinematics-informed descriptors roughly 36–42%, consistent with both information sources being jointly integrated during input contextualization. At Stage 2, attention shifts consistently toward symbolic tokens across all classes (overall: 64.4% SAX vs. 35.6% kinematics), suggesting symbolic dynamics play the primary role at label generation time. Per-class Stage 2 values for symbolic tokens range from 58.3% (*Sitting*) to 68.1% (*Stairs*), indicating that the relative reliance on symbolic versus physical information varies by activity even though symbolic attention dominates throughout.

Axis-level attention. Figure 6.7 reports Stage 2 peak attention (×baseline) per sensor axis and activity class. Gyro-Z consistently dominates across all five classes. For dynamic activities, its peak reaches 8.18× for *Stairs*, 6.76× for *Walking*, and 4.28× for *Jogging*, while every other axis stays below 3.5×. For static activities, Gyro-Z still leads but at considerably lower levels (*Sitting*: 1.45×; *Standing*: 1.61×). This aligns with the physical role of the gyroscope's Z-axis under the rWISDM orientation protocol (Section 6.2.2): with the Y-axis aligned to gravity, Gyro-Z becomes the primary carrier of rotational dynamics during locomotion.

The dynamic/static contrast maps directly onto the attention structure. Dynamic activities show sharp, well-localised Gyro-Z peaks, indicating that the model con-

Table 6.10: Two-stage attention density analysis. Mean normalized attention densities assigned to kinematics-informed and symbolic tokens at Stage 1 (input contextualization) and Stage 2 (label generation).

Class	Samples	Stage 1		Stage 2	
		Kinematics	Symbolic	Kinematics	Symbolic
Overall	1155	0.403	0.597	0.356	0.644
Jogging	231	0.410	0.590	0.330	0.670
Sitting	231	0.416	0.584	0.417	0.583
Stairs	231	0.364	0.636	0.319	0.681
Standing	231	0.421	0.579	0.357	0.643
Walking	231	0.400	0.600	0.351	0.649

centrates on specific symbolic transitions within that axis. Static activities show near-baseline attention across every axis, including Gyro-Z – a pattern consistent with what z-normalization produces for low-variance postures: uniform symbolic sequences in which no single transition stands out. The absence of a dominant peak is therefore not a failure to find a signal, but plausibly the signal itself – the model may be reading global sequence uniformity as the distinguishing feature of static classes.

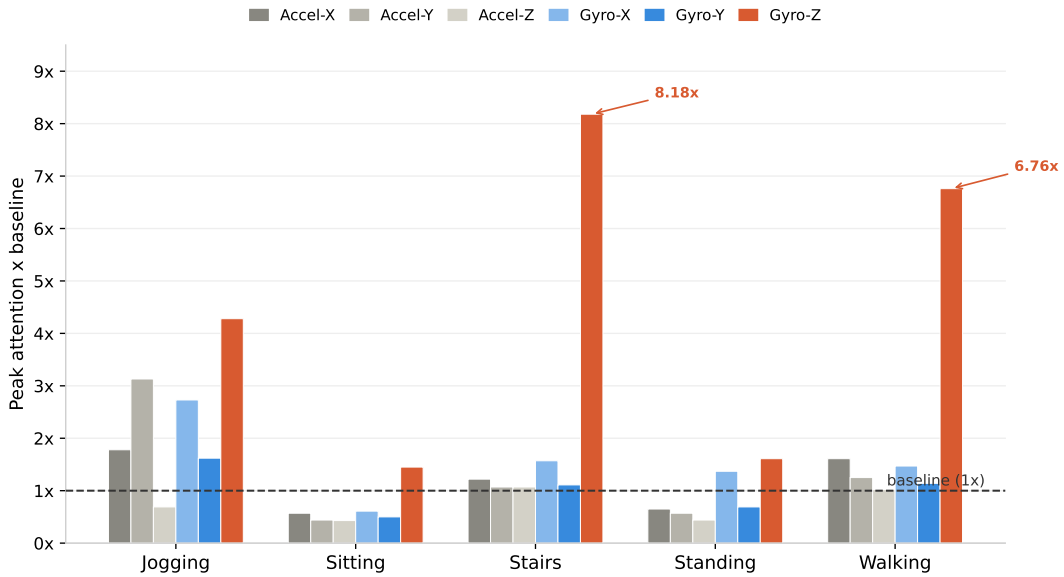


Figure 6.7: Peak attention × baseline per sensor axis and activity class (Stage 2). The dashed line marks the uniform attention baseline ($1/L$). Gyro-Z consistently receives the highest attention across all activity classes, with dynamic activities (*Stairs*: 8.18×, *Walking*: 6.76×, *Jogging*: 4.28×) exhibiting substantially stronger local focus than static activities (*Sitting*: 1.45×, *Standing*: 1.61×), which show near-baseline, distributed attention consistent with uniformly flat symbolic sequences.

Motif-level analysis. Table 6.11 lists the three most prominent Gyro-Z motifs per activity class, selected by weighing attention strength against frequency of occurrence. This combination is necessary: attention strength alone elevates rare but highly-attended patterns, frequency alone elevates recurring but weakly-attended ones, and neither criterion on its own identifies motifs that are both discriminative and stable. The contrast between static and dynamic classes is clear. For *Sitting* and *Standing*, the leading motifs are high-frequency and near-baseline in attention – `ccccb` (count = 5, 1.45×) and `bbbBb` (count = 4, 1.61×) respectively – reflecting diffuse focus over uniform, low-variance sequences. The trend-aware encoding supports this reading: both motifs are predominantly lowercase, encoding flat or falling slopes with minimal amplitude variation, consistent with the stable gravitational orientation that characterizes static postures.

Dynamic activities present a different picture. For *Jogging*, all three top motifs appear at count = 2 with moderate-to-high attention (3.11–4.28× baseline), suggesting the model spreads its focus across several transition-rich patterns rather than concentrating on one. *Stairs* shows the opposite: high-attention, low-frequency motifs dominate – `ACcDd` (8.18×, count = 1), `CcABc` (7.07×, count = 1), `aCdcb` (4.29×, count = 2) – indicating that the model attends strongly to specific symbolic transitions when they occur, even when those transitions do not recur consistently across samples. The top motif `ACcDd` encodes a low-amplitude rising segment leading into a sustained high-amplitude region, which is physically interpretable as the rotational signature of weight transfer during stair climbing. *Walking* also yields exclusively low-frequency motifs (count = 1 throughout), pointing to subject-specific Gyro-Z dynamics that do not converge on a shared symbolic pattern. All of these readings are illustrative: attention serves here as a proxy for model focus, not as a verified indicator of causal importance.

Across all five activities, the attention analysis produces a physically coherent picture: symbolic dynamics consistently drive label generation, Gyro-Z draws the strongest focus, static classes are associated with distributed attention over low-variance recurring motifs, and dynamic classes with concentrated attention on specific high-energy transitions that vary across subjects. This pattern aligns with the dual-stream rationale of Section 6.2.3 – symbolic sequences encoding temporal structure, kinematics-informed descriptors supplying the physical grounding that normalization removes.

Per-head concentration on Gyro-Z. To determine whether the Gyro-Z attention for *Stairs* is distributed evenly across heads or driven by a small number of specialized ones, a per-head analysis is run across all 231 *Stairs* test samples. For each sample, attention matrices from the final transformer layer are extracted without head-averaging, and peak attention on the Gyro-Z token span is recorded for each of Mistral-7B's 32 heads at the Stage 2 query position. The result, shown in Figure 6.8, is strongly concentrated: head 24 alone reaches 46.07× baseline and accounts for roughly 38% of total cross-head attention on this axis. A secondary cluster of four heads (h00, h02, h11, h19) contributes at 5–10×, while the remaining 27 sit at or below baseline. Removing head 24 reduces the cross-head mean to 2.38×, confirming it as the dominant carrier of the Gyro-Z signal. The head-averaged 8.18× reported in Figure 6.7 is therefore directionally accurate but masks a much sharper underlying concentration, consistent with the model having developed a functionally specialized head for rotational dynamics during stair climbing. As elsewhere in this analysis, the

Table 6.11: Top-3 symbolic motifs on the Gyro-Z axis per activity class (Stage 2 attention), ranked by jointly considering attention strength and motif frequency. Gyro-Z is selected as the reporting axis because it consistently receives the highest peak attention across all five activity classes (Figure 6.7). Peak attention is reported as a multiple of the uniform baseline ($1/L$); count indicates motif frequency across test samples.

Class	Motif	Peak × Baseline	Count
Jogging	`dbAdb`	4.28	2
	`AdcAd`	4.08	2
	`dbBda`	3.11	2
Sitting	`ccccb`	1.45	5
	`Dcbcb`	3.90	1
	`bCbCb`	2.47	2
Stairs	`ACcDd`	8.18	1
	`CcABc`	7.07	1
	`aCdcb`	4.29	2
Standing	`bbbBb`	1.61	4
	`aACCb`	3.51	1
	`BaAbc`	3.47	1
Walking	`bAcCc`	6.76	1
	`dbbbC`	5.44	1
	`dcbaD`	4.05	1

finding is descriptive rather than causally verified.

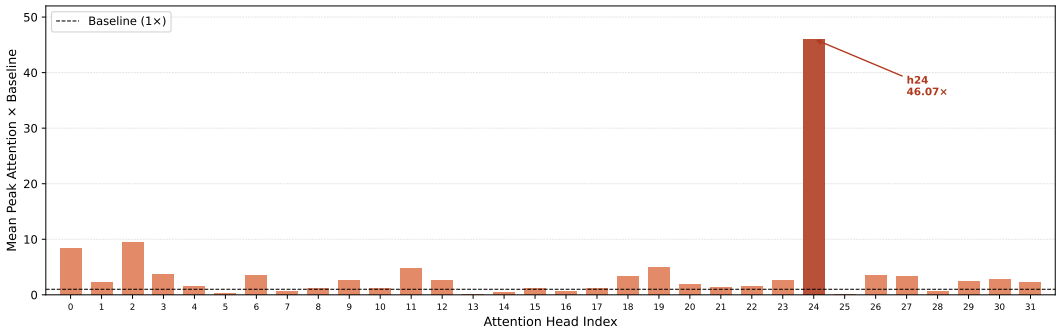


Figure 6.8: Per-head mean peak attention × baseline on the Gyro-Z axis (Stage 2, final transformer layer) for the Stairs class ($n = 231$). Each bar is the mean peak attention of one of Mistral-7B's 32 attention heads over the Gyro-Z token span, normalized by the uniform baseline ($1/L$). The dashed line marks baseline (1×). Head 24 (dark red) dominates at 46.07×, accounting for 38% of total cross-head attention. A secondary cluster appears at heads 0, 2, 11, and 19 (5–10×); remaining heads sit near or below baseline.

Perturbation experiment. To move beyond descriptive attention patterns and test whether Gyro-Z actually influences predictions, the Gyro-Z token sequence is replaced at inference time with symbols drawn uniformly at random from the trend-aware alphabet $\{a, b, c, d, A, B, C, D\}$, while all other prompt elements remain unchanged. If the model genuinely relies on this axis, corrupting it should degrade per-class recall; the comparison against original predictions is reported in Table 6.12.

Table 6.12: Per-class accuracy (recall) before and after Gyro-Z token randomization at inference time. Drop is reported in percentage points (pp); a negative value indicates an accuracy improvement after perturbation.

Class	Original (%)	Perturbed (%)	Drop (pp)
Jogging	94.37	90.48	+3.90
Sitting	82.68	84.42	−1.73
Stairs	100.00	100.00	+0.00
Standing	89.61	88.74	+0.87
Walking	94.81	92.64	+2.16

An overall accuracy drop of 1.04 pp indicates that the model is substantially robust to Gyro-Z corruption, with the remaining axes and kinematics descriptors compensating effectively. The per-class breakdown is more informative and aligns well with both the attention weights of Figure 6.7 and the motifs of Table 6.11. *Jogging* suffers the largest drop (+3.90 pp), consistent with its distributed attention across three recurring motifs (count = 2 each) and pointing to genuine reliance on Gyro-Z transitions. *Walking* shows a moderate drop (+2.16 pp), reflecting strongly-attended but subject-specific transitions that do not recur across samples. *Stairs*, despite carrying the highest attention (8.18×), shows no drop at all: its count-1 motifs are not systematically relied upon, and other streams compensate when the axis is corrupted. Static activities are largely unaffected, consistent with their near-baseline attention and the model's apparent reliance on global sequence uniformity rather than any specific local transition. These results reinforce the descriptive framing of the attention analysis and demonstrate that the dual-stream design remains robust even under targeted corruption of its most-attended axis.

6.4.5 Computational Cost

Table 6.13 reports fine-tuning time and per-window inference latency for all methods on MotionSense. Symbolic and deep learning baselines range from fractions of a second to roughly 82 seconds of fine-tuning, with inference latencies below 0.1 ms per window. RoBERTa is more expensive but still accessible at 184 seconds and 0.818 ms per window, reflecting its compact 125M-parameter architecture. Causal LLM-based methods occupy a different order of magnitude: LLM-ABBA requires roughly 2 hours of fine-tuning and SAX_HAR-LLM 3.42 hours, with inference latencies of 561 ms and 574 ms respectively – approximately 10,000× slower than MultiROCKET (0.055 ms). This gap is a direct consequence of autoregressive token generation at inference time, as opposed to the kernel-based feature extraction that makes MultiROCKET fast. SAX_HAR-LLM is not designed to be computationally competitive; the case for it rests on the interpretability that its symbolic representation and attention-based analysis provide.

Table 6.13: Computational cost comparison across all methods on MotionSense. Inference latency is measured per window. SAX_HAR-LLM and LLM-ABBA fine-tuning times reflect QLoRA fine-tuning for 8 epochs with batch size 64 on an NVIDIA RTX 6000 Ada Generation GPU (48 GB VRAM); all other methods were measured on an Intel Core i9-14900K workstation (128 GB RAM, CPU only). Absolute times are therefore not directly comparable across LLM-based and non-LLM methods, though the order-of-magnitude differences remain informative. As a hardware requirement, SAX_HAR-LLM requires a minimum of 4,567 MB (≈4.5 GB) of GPU VRAM at inference time.

Method	**Fine-tuning Time**	**Inference Latency per Window (ms)**
BOP	0.13 s	0.011
BOP+MII	0.15 s	0.011
ResNet	8.89 s	0.022
DeepConvLSTM	10.80 s	0.027
InceptionTime	18.5 ± 5.2 s	0.029 ± 0.001
Vanilla Transformer	5.28 s	0.073
MultiROCKET	82.4 ± 4.3 s	0.055 ± 0.008
RoBERTa	184.0 ± 3.1 s	0.818 ± 0.007
LLM-ABBA	1.99 h	561.5
SAX_HAR-LLM	3.42 ± 0.07 h	573.6 ± 2.6

6.5 Discussion

Across both benchmarks, SAX_HAR-LLM ranks 2.5 on average, second on WISDM and third on MotionSense. The ranking itself is less telling than the gap it conceals: 5.62 percentage points above BOP+MII on WISDM and 10.78 on MotionSense. A bag-of-patterns classifier aggregates symbolic occurrences into a frequency histogram, losing the order in which they appear; the LLM operates on the full token sequence, where position, local transitions, and long-range structure all remain available. The fact that this improvement is observed in two datasets with different sampling rates, sensor configurations, and sets of activities suggests that the symbolic interface with the language model is substantively useful and not merely structurally convenient.

Against numerical baselines, SAX_HAR-LLM outperforms ResNet, DeepConvLSTM, and the Vanilla Transformer on both datasets and InceptionTime on WISDM. MultiROCKET is the only method that ranks above it consistently, by 3.21 points on WISDM and 5.69 on MotionSense. This gap has a straightforward interpretation: MultiROCKET operates on thousands of random convolutional features that carry no physical meaning, while SAX_HAR-LLM's predictions are tied to specific symbolic patterns on specific sensor axes. The accuracy difference is the price of that transparency, and whether it is worth paying depends on the application – reasonable in settings where decisions must be auditable, harder to justify where latency or memory are the binding constraints.

A consistent difficulty runs across both datasets: dynamic activities are classified reliably, static postures are not. Z-normalization is the structural cause – it preserves temporal shape while erasing gravitational orientation, leaving *Sitting* and *Standing* indistinguishable in the symbolic domain without additional physical grounding. The *Downstairs* class on MotionSense (F1: 0.64) is the sharpest failure, though nearly

every method struggles here; only InceptionTime (0.82) and MultiROCKET (0.84) handle it well. For this framework specifically, two factors converge: descent dynamics produce flat, lowercase-dominated sequences that resemble static-posture patterns, and the kinematics descriptors target the *Sitting/Standing* ambiguity rather than the direction of locomotion. Addressing this would require descriptors sensitive to step periodicity or vertical impulse asymmetry.

The ablation results make the contribution of each component precise. With only the symbolic stream (Ablation_1), dynamic activities are classified almost perfectly – no physical descriptor, no magnitude information, just an ordered sequence of alphabetic tokens. SAX strings evidently retain enough temporal structure for the pre-trained backbone to discriminate dynamic motion on its own. What they cannot do is separate static postures after normalization has removed the amplitude information that distinguishes them. The Vanilla Transformer, receiving exactly the same inputs as the full model but initialized from scratch, reaches only 78.01%, confirming that the advantage comes from pre-trained sequential priors rather than from the representation itself. The comparison between Ablation_1 and the Vanilla Transformer is particularly instructive: both reach roughly 78% accuracy, but through opposite deficiencies – one has the backbone but not the physics, the other has the physics but not the backbone. Each path exposes a different necessary condition, and the full model requires both.

The attention analysis reveals structure that was never explicitly taught. Gyro-Z dominates attention across all five activity classes, in line with its physical role as the primary carrier of rotational dynamics once the Y-axis is gravity-aligned under the rWISDM protocol. For dynamic activities this manifests as sharp, localised peaks on specific symbolic transitions; for static activities attention is flat and near-baseline across every axis, consistent with the uniformly low-variance sequences that normalization produces for these classes. The perturbation experiment gives this a predictive interpretation: corrupting Gyro-Z reduces recall for *Jogging* and *Walking*, where the model draws on recurring transition patterns, but leaves *Stairs* unaffected, its highly activity-specific motifs compensated by the remaining streams. These findings are correlational rather than causal, but the capacity to link a prediction to a concrete motif on a named sensor axis sets this framework apart from every numerical baseline in this comparison.

6.6 Synthesis

This chapter set out to answer a structural question rather than an optimization one: not how much further SAX-based accuracy can be pushed, but whether a symbolic representation, suitably designed, can serve directly as the input vocabulary of a pre-trained language model. The evidence assembled across Sections 6.4 and 6.5 answers it affirmatively, with three qualifications that matter for how the result should be read. Symbolic abstraction alone is sufficient for discriminating activities whose kinematic signature survives normalization; kinematics-informed descriptors intervene only where z-normalization renders classes symbolically indistinguishable, acting as a targeted auxiliary mechanism rather than a foundational requirement. The controlled comparison against a Vanilla Transformer trained on identical inputs isolates the pre-trained backbone, rather than the symbolic representation alone, as the decisive contributor to that success. And the attention analysis suggests, without establishing causally, that fine-tuning on symbolic sensor data steers the model toward physically meaningful predictive focus.

This result also closes a loop that runs through several earlier chapters of this thesis. Chapter 3's taxonomy identified trend loss as a recurring SAX weakness; Chapter 4 answered it once with SAA, carrying trend in a structurally separate symbolic branch; this chapter answers it a second time by folding trend into the case of the existing symbol instead (Section 6.2.3), a choice whose value becomes concrete only once the symbolic string has to serve as an LLM token sequence, where every additional position has a cost SAA's separate-branch design does not need to pay. More broadly, this is the first chapter in which the SAX-based representations developed across Chapters 4 and 5 are paired with a sequence model that brings its own pre-trained priors, rather than with a frequency-histogram classifier built from nothing – the same shift in spirit, if not in mechanism, that motivates pairing SAX-derived representations with learned graph embeddings as an alternative route to a richer classifier, explored independently in the next chapter. It is also the point at which this thesis's two methodological families – SAX-based time series representation, and the language-model-based methods this thesis takes up more fully in its later chapters – meet directly for the first time, with the dual-stream design serving as a concrete answer to what a principled point of contact between them looks like.

Several limitations follow directly from the design choices examined above and define natural next steps. Fine-tuning and deploying a 7-billion-parameter backbone is substantially more costly, in time and GPU memory, than any symbolic or even kernel-based baseline considered here; as distillation techniques mature and smaller backbones demonstrate comparable sequential-modeling capacity, this cost should fall, making the symbolic-LLM pairing increasingly viable for resource-constrained deployment. On the evaluation side, the single 80/20 subject split may carry non-trivial variance, and a subject-level k-fold protocol would offer a more robust generalization estimate, a direction best pursued jointly with a broader study of subject-level generalization across multiple datasets and HAR frameworks.

Beyond HAR, the framework extends in the near term to other sensing domains by replacing the kinematics-informed stream with descriptors appropriate to the new target: the symbolic-LLM interface itself is domain-agnostic, since SAX applies to any continuous time series and the LLM requires only a discrete, ordered token sequence. The design question in a new domain reduces to identifying which classes become symbolically indistinguishable after normalization and choosing descriptors that restore the lost discriminative information – a substantially more tractable problem than redesigning the representational pipeline from scratch, and one for which fault diagnosis in rotating machinery and EEG-based brain-computer interfaces are natural candidates. In the longer term, a more fundamental direction would eliminate domain-specific descriptors altogether by integrating symbolic tokens with a numerical tokenization strategy that preserves magnitude information directly within the token sequence, addressing normalization-induced ambiguity at the representational level rather than compensating for it after the fact. A related and more immediate gap, raised already in Section 6.2.5, is that the case relationship between uppercase and lowercase symbol variants is learned implicitly through fine-tuning rather than structurally encoded in the tokenizer; designing a tokenization scheme that preserves this relationship explicitly, together with a systematic sensitivity analysis of the trend threshold ε, is a natural and currently open next step. Finally, two extensions would relax the framework's dependence on labeled data from known subjects and activities: symbolic prompts augmented with natural-language descriptions of expected motion signatures, which could in principle support zero-shot classification of unseen activities if the model

genuinely exploits structured regularities in SAX sequences rather than memorizing them; and incremental LoRA adaptation for new subjects or activities without full retraining, which would make the framework practical for longitudinal monitoring settings where data accumulates continuously. Both ultimately turn on the same open question raised throughout this chapter – whether pre-trained sequential priors transfer to symbolic sensor data through structural alignment with natural language, or through a deeper sequential-modeling bias – a question future interpretability work on this framework should address directly.

This page has been left blank intentionally.

Chapter 7

Graph Neural Embeddings for Time Series Classification

Chapters 4 and 5 extended SAX along two different axes – richer per-segment descriptors in Chapter 4, and optimized parameter search in Chapter 5 – while leaving the representation used for classification itself unchanged: a frequency histogram over SAX words, that is the BOP model introduced in Section 2.2.3. BOP's histogram is position-invariant but that same invariance has a direct cost: two time series whose SAX words occur in entirely different orders can produce identical histograms. The sequences

$$\texttt{abbc} \rightarrow \texttt{bbcd} \rightarrow \texttt{bcde} \rightarrow \texttt{cdee}$$
$$\texttt{cdee} \rightarrow \texttt{bcde} \rightarrow \texttt{bbcd} \rightarrow \texttt{abbc}$$

are reverses of one another, yet BOP aggregates both into the same word-count vector: the transition structure that distinguishes them is exactly what the histogram discards. Section 2.4.1 already flagged this as a limitation of dictionary-based methods more broadly – BOP's "abstraction of temporal relationships between patterns limits its expressive capacity" – and pointed, in a single sentence, to representing SAX words as a graph rather than a histogram as one way to recover what is lost. This chapter develops that direction in full.

The approach explored here keeps the SAX vocabulary entirely intact – no new symbol, no modified discretization rule: each distinct SAX word becomes a node, and the order in which words actually occur in the sequence is encoded as edges between them, so that the resulting structure carries the transition information a histogram cannot. A GNN then learns embeddings over this structure, which are evaluated, alongside BOP and combinations of the two, using a standard classifier. Two distinct schemes for building this graph are considered, differing in how node features and edges are constructed, and the resulting embeddings are evaluated against BOP, against each other, and in combination with BOP, across a diverse set of benchmark time-series datasets.

This chapter opens a direction rather than closing one [112]: having developed and evaluated symbolic representations across the preceding chapters, the natural next question is whether richer structural models can extract further value from the same SAX vocabulary. GNNs offer one principled answer, and this chapter provides a first empirical grounding for that idea. The constructions and evaluation presented here are a starting point, with the more thorough investigation of architecture choices,

This chapter is based on the following publication: [C4].

parameter sensitivity, and multivariate extensions left as a clear and well-motivated direction for continued work beyond this thesis. Section 7.1 positions this question within the broader literature on graph-based time-series modeling; Section 7.2 details the two graph-construction schemes and the classification pipeline built on them; Section 7.3 reports classification results across the benchmark datasets; and Section 7.4 reflects on what the comparison reveals and where the most promising avenues for further development lie.

7.1 Background

Time-series classification arises across domains as varied as finance, healthcare, and manufacturing, where handcrafted feature methods offer interpretability but limited expressiveness, and deep models trained directly on the raw signal gain predictive power at the cost of transparency and computational overhead. Graph-based methods offer a different way to represent the structure: rather than treating a time series as a sequence of independent observations, a graph can encode dependencies between observations, patterns, or entire series directly as structure, and GNN architectures in particular allow this structure to scale to large datasets while remaining flexible about what a node or an edge is taken to represent [234–238].

Graph structure has been applied to time-series classification along several distinct works. Mishra et al. [239, 240] introduced Graft, a topological graph-based framework that represents symbolic patterns as nodes and encodes temporal dependency and co-occurrence as edges, supporting path-based clustering and anomaly detection in addition to classification. Bünger et al. [241] demonstrated that graph-based semi-supervised methods, such as Graph Convolutional Networks (GCNs), can leverage both labeled and unlabeled time series jointly for classification. A different framing appears in SimTSC [242], which treats classification itself as a node-classification problem: nodes represent entire time series, and edges encode pairwise similarity between them, so the graph spans the whole dataset rather than a single sequence.

In alighnment with the approach developed in this chapter, Ma et al. [243] paired graph embedding directly with the BOP algorithm, explicitly seeking to recover the sequential structure that BOP's histogram discards while keeping a symbolic representation as the graph's underlying basis. This is the most directly related prior work, sharing both the motivating gap addressed here – BOP's position-invariance – and the general strategy for closing it, namely using graph structure to recover what the histogram loses.

A parallel body of work applies graph structure to forecasting rather than classification, illustrating how broadly the same underlying idea – letting edges carry dependency structure a flat feature vector cannot – has been put to use. SDGL (Static and Dynamic Graph Learning Network) [244] learns a static graph for stable relationships and a dynamic graph for time-varying ones, capturing both long- and short-term dependencies simultaneously. Ye et al. [245] instead used a hierarchical structure with dilated convolutions to model evolving associations between series over multiple scales, while Yu et al.'s RGSL (Regularized Graph Structure Learning) [246] combines explicit, domain-knowledge graphs with implicit, learned ones to improve both interpretability and forecasting accuracy. Outside forecasting specifically, Adams et al. [247] examined graph-based monitoring of predictive processes, showing that graph embeddings preserve structural information in event-based time series that other representations lose. Surveying this landscape more broadly, Silva et al. [248]

organize the many ways a time series can be turned into a network – visibility graphs, transition networks, and proximity graphs among them – into a single taxonomy.

The graph neural network model formalized by Scarselli et al. [249] learns a representation for each node by iteratively aggregating information from its neighbors – a form of message passing in which a node's embedding comes to reflect not only its own features but the structure of its local neighborhood. This is what makes a GNN a natural fit for the graphs constructed in Section 7.2: a SAX word's embedding can be shaped by which words tend to precede or follow it, exactly the transition information a histogram representation discards. Edge-conditioned convolution operators extend this idea further by letting the aggregation itself depend on edge attributes rather than only on node identity, allowing edge weight – transition frequency, or distance-based decay, depending on the construction – to directly influence how neighboring information is combined; this is the mechanism the GNN described in Section 7.2.3 relies on.

7.2 Methodology

7.2.1 SAX Transformation

The pipeline starts from the same symbolic representation used throughout this thesis: a time series is reduced via PAA (Section 2.2.1) and discretized into a SAX string using Gaussian breakpoints (Section 2.2.2), with the same alphabet size a and word length w notation established there. No modification is made to either step here – the contribution of this chapter lies entirely in what is built on top of the resulting symbolic sequence, not in how that sequence is produced.

7.2.2 Graph Formation

Two distinct schemes are used to turn a SAX-transformed sequence into a graph, differing in how nodes are represented and how edges are formed. Both take the same SAX string as input and are evaluated under identical downstream conditions, so that any difference in classification performance can be attributed to the construction itself.

Transition Graph Embeddings (TGE)

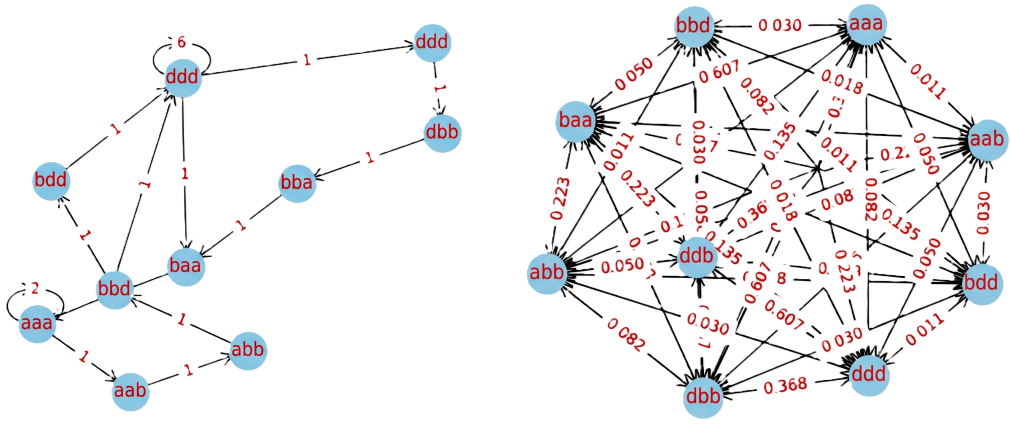
TGE constructs a directed graph in which each unique SAX word appearing in the sequence is a node, with directed edges placed between consecutively occurring words – directly capturing the sequential dependency a BOP histogram discards. Node features are one-hot vectors over a vocabulary built across the full dataset, giving a high-dimensional but sparse node representation. Edge weights are set by transition frequency: the more often one word is immediately followed by another across the sequence, the stronger their connecting edge. The resulting graph structure therefore reflects observed co-occurrence patterns directly, with transition likelihoods implicitly encoded through edge weight rather than through any separate probabilistic model.

Distance-Aware Graph Embeddings (DGE)

DGE instead encodes each SAX word as a compact numerical tuple, with dimensionality equal to the word length w and reflecting the word's internal symbolic structure, rather

than a high-dimensional one-hot vector. Edges are also formed differently: rather than connecting only immediately consecutive words, DGE computes node reachability via breadth-first search (BFS), so that words separated by several intervening positions can still be connected. Edge weight is then an exponential decay function of BFS distance, rather than a binary transition indicator, so that nodes remain linked even when not adjacent in the sequence – letting the graph carry longer-range dependency information that TGE's strictly local transitions cannot.

Figure 7.1 contrasts the two constructions directly: TGE's edges following immediate SAX-word transitions, against DGE's distance-weighted edges spanning longer ranges within the same sequence.



(a) TGE (b) DGE

Figure 7.1: Graph representations of a SAX-transformed sequence: (a) TGE, with edges following direct SAX-word transitions; (b) DGE, with distance-weighted edges enabling long-range dependencies.

7.2.3 Graph Embedding Learning

Both graph constructions are passed through a GNN that learns embeddings capturing SAX-word dependencies, applying edge-conditioned convolution (NNConv, Section 7.1) to update node features according to both node content and edge weight, followed by global mean pooling to obtain a single fixed-size embedding per graph. TGE supplies high-dimensional, sparse one-hot node features while DGE supplies a compact, continuous feature space together with edge weights derived from BFS-distance decay. Thus, the two constructions feed the same GNN architecture with structurally different inputs, and the resulting embeddings reflect that difference in how each scheme shapes the patterns available for it to learn.

7.2.4 Classification and Evaluation

Feature Representations

Five feature representations are compared, isolating the contribution of graph-based embedding both on its own and in combination with the established BOP baseline: BOP histograms alone (Section 2.2.3); TGE embeddings alone; DGE embeddings alone; BOP concatenated with TGE embeddings (BOP+TGE); and BOP concatenated with DGE embeddings (BOP+DGE). The two combined representations test whether

graph-based structure adds information beyond what BOP's frequency histogram already captures, rather than simply substituting for it.

Experimental Setup

Each of the five feature representations is classified independently using a Random Forest with 100 trees, with classification performed at the level of individual time series, using each dataset's predefined UCR train/test split. Classification accuracy is the primary evaluation metric. The SAX parameters are held fixed across all datasets and representations, at alphabet size $a = 4$ and word length $w = 3$, a configuration chosen to balance structural granularity against feature sparsity.

Datasets

Datasets are drawn from the UCR Time Series Classification Archive [250], selected to span a deliberately diverse range of time-series characteristics rather than a single domain: periodic signals such as ECG200, sensor-based recordings such as InsectWingbeatSound, and nonstationary, higher-variability series such as CricketX. This diversity is intended to test the generalizability of the proposed graph-based representations across qualitatively different notions of what makes a time series classifiable, rather than to optimize performance on any one dataset type.

7.3 Results

Table 7.1 reports classification accuracy for all five feature representations across the twelve selected UCR datasets, together with each dataset's SAX ratio – the percentage of the original series compressed into a single SAX symbol.

Table 7.1: Classification accuracy (%) of the tested methods on selected UCR Time Series datasets, alongside each dataset's series length and SAX ratio. The SAX Ratio (%) column reports the percentage of the original time series used to form a single SAX symbol.

Dataset	TS Length	SAX Ratio (%)	BOP	TGE	DGE	BOP+TGE	BOP+DGE
Car	577	0.5	48.3	36.7	28.3	46.7	**53.3**
ChlorineConcentration	166	10	55.1	55.3	**55.4**	55.3	55.3
CinCECGTorso	1639	0.1	**53.5**	36.6	42.5	52.0	**53.5**
CricketX	300	0.5	39.7	30.3	20.8	**42.6**	42.3
ECG200	96	15	**81.0**	75.0	76.0	**81.0**	80.0
ECG5000	140	10	91.2	90.4	90.4	**91.7**	91.6
ECGFiveDays	136	5	64.9	64.5	64.0	62.7	**67.1**
FordA	500	5	52.3	52.1	49.5	**54.1**	52.6
FordB	500	10	52.0	51.1	**54.6**	51.4	51.1
GunPoint	150	5	**92.0**	82.0	76.0	85.3	85.3
InsectWingbeatSound	256	15	34.1	34.0	35.1	34.2	**35.8**
ItalyPowerDemand	24	15	88.4	89.3	88.9	**90.9**	90.0
Average	–	–	62.7	58.1	56.8	62.3	**63.2**

On average, BOP remains the strongest standalone representation (62.7%), and BOP+DGE the strongest representation overall (63.2%), narrowly ahead of plain BOP and BOP+TGE (62.3%); TGE and DGE alone follow both (58.1% and 56.8%, respectively). This ordering already signals the central pattern in the table: graph-based embeddings, on their own, rarely match BOP, but combining them with BOP rather than replacing it can recover, and on some datasets exceed, BOP's standalone accuracy.

The dataset-level pattern is least favorable to the proposed graph constructions on *Car* and *CricketX*: TGE and DGE alone fall well below BOP on each (TGE/DGE at 36.7%/28.3% versus BOP's 48.3% on *Car*; 30.3%/20.8% versus 39.7% on *CricketX*), suggesting that for these higher-variability, movement- or sensor-driven series, the transition structure captured by the graph alone is not sufficient to substitute for BOP's frequency information. Combined with BOP, however, the picture changes: BOP+DGE gives the best result on *Car* (53.3%, five points above BOP alone) and BOP+TGE the best on *CricketX* (42.6%), indicating that the graph-based signal is informative once allowed to supplement BOP's histogram rather than stand in for it. *InsectWingbeatSound* is a partial exception to the pattern reflecting that a graph alone is weaker: DGE alone (35.1%) is marginally above BOP (34.1%), though the margin is small enough that it should not be read as a strong reversal of the broader trend.

On the three ECG-derived datasets (*ECG200, ECG5000, ECGFiveDays*) and the *ItalyPowerDemand* dataset, every method performs within a relatively narrow band, consistent with these periodic, well-structured signals being comparatively easy for any of the five representations to separate. Three of the aforementioned datasets show only marginal differences between methods, and the best-performing representation is, in each case, a combination of BOP with one of the graph embeddings rather than BOP alone or a graph embedding alone (BOP+TGE on *ECG5000* and *ItalyPowerDemand*; tied BOP and BOP+TGE on *ECG200*). *ECGFiveDays* stands out from this pattern: BOP+DGE reaches 67.1%, a 2.2-point improvement over BOP's 64.9%, the single largest margin recorded for any combined representation over standalone BOP in the table.

Taken as a whole, these results suggest that the proposed graph constructions are not a wholesale replacement for BOP – on this benchmark suite, BOP alone remains competitive or superior to either graph embedding used in isolation on the majority of datasets – but that they are a productive complement to it, recovering some of the sequential information BOP's histogram discards and, when combined with BOP, matching or exceeding it on several of the datasets examined, most clearly on those compressed at lower SAX ratios.

7.4 Discussion

Table 7.1 shows that TGE and DGE alone rarely outperform BOP, trailing it by a wide margin on average (58.1% and 56.8% versus 62.7%). Their value comes from combination rather than substitution: BOP+TGE and BOP+DGE recover, and on several datasets exceed, BOP's standalone accuracy, suggesting the transition structure a graph encodes is informative but not, by itself, a sufficient replacement for BOP's frequency information. The clearest gains over BOP alone – *Car* (BOP+DGE, +5.0 points), *CricketX* (BOP+TGE, +2.9 points), *ECGFiveDays* (BOP+DGE, +2.2 points) – occur on datasets with low SAX ratios (0.5%, 0.5%, and 5%).

Several limitations follow from how the evaluation was constructed. SAX parameters were held fixed at $a = 4$ and $w = 3$ across all datasets, with no sensitivity analysis – a direct point of contact with Chapter 5's per-dataset parameter search, which this chapter's fixed-parameter evaluation does without. The scope is also narrower than Chapters 5 and 6: a single classifier (Random Forest, 100 trees), no deep-learning baselines, no statistical significance testing, a single predefined train/test split per dataset, and all twelve datasets univariate, so the multivariate setting central to Chapter 4's MII remains untested here.

These limitations point directly to next steps: a systematic SAX parameter selection, ideally paired with Chapter 5's optimization framework so parameters and graph construction are tuned jointly; a more developed graph-learning procedure, whether through alternative edge-conditioned operators, learned edge weighting, or end-to-end training with the downstream classifier; and an extension to multivariate time series following Chapter 4's channel-wise treatment, where each channel contributes its own transition graph fused at the graph or embedding level.

7.5 Synthesis

This chapter closes the methodological arc of this thesis by returning to an early observation: Chapter 2's introduction of BOP noted that discarding word order is a trade-off, and Section 2.4.1 named graph-based embedding as one way to revisit it without abandoning SAX's symbolic vocabulary. The evaluation here gives that proposal a concrete, if modest, answer: graph structure built on SAX-word transitions is not by itself a replacement for BOP's histogram, but combined with it, recovers some of what the histogram discards – most clearly on series compressed into longer symbolic strings.

This closes a pattern running through the preceding chapters, each approaching SAX's core trade-off – compact symbols purchased at the cost of discarded information – from a different angle: richer per-segment descriptors in Chapter 4, principled parameter search in Chapter 5, and, here and in Chapter 6, a more expressive structure built on top of the symbolic sequence rather than a change to its construction. What unites them is a shared discipline: each preserves the underlying representation and asks what can be added around it, rather than trading interpretability away for accuracy.

This chapter is accordingly best read as a starting point rather than a closed result. The next stage of this line of work is to identify what information a graph constructed over SAX words actually needs to carry for a GNN to become a genuinely competitive representation for downstream time-series tasks – a question the present results motivate without yet answering.

This page has been left blank intentionally.

Chapter 8

Conclusions and Future Work

8.1 Summary of Contributions

This thesis set out to advance the state of the art in symbolic time series representation and classification, taking the Symbolic Aggregate approXimation (SAX) method and the Bag-of-Patterns (BOP) model built on top of it as a common foundation. Rather than replacing this foundation, the thesis asked what could be recovered or built around it – preserving the interpretability and computational efficiency that make symbolic representations valuable, while systematically closing the gaps that limit their expressiveness.

The thesis began, in Chapter 3, by conducting a systematic review of the SAX literature and organizing the resulting body of work into a taxonomy structured around the specific weakness each extension addresses. This taxonomy served a dual purpose: it established the conceptual map from which the original contributions depart, and it provided a reference framework against which the design choices of each proposed method can be understood. The recurring gaps it identified – parameter sensitivity, trend loss, limited multivariate support, disregard for sequential order in BOP, and the absence of principled connections to modern sequence models – form the motivating thread running across Chapters 4 to 7.

Chapter 4 addressed two of these gaps independently through a pair of complementary extensions. The Multichannel Intelligent Icon (MII) captured cross-channel symbolic co-occurrence in multivariate time series by forming joint symbolic words from corresponding positions across channels, recovering spatial correlation that per-channel BOP representations treat as independent. The Slopewise Aggregate Approximation (SAA) addressed trend loss through a parallel, independently discretized angle-based branch, carrying slope information in a structurally separate symbolic string that can be combined with the standard mean-based one at the feature level. Evaluated on a human activity recognition benchmark, both methods improved upon the BOP baseline across all eight activity classes, with SAA's within-axis trend information proving especially effective for the static postures that symbolic amplitude representations struggle to separate.

Chapter 5 turned to the parameter selection problem, applying the DIRECT global optimization algorithm to the joint tuning of SAX parameters across four feature extraction strategies for multivariate time series classification. DIRECT's properties – requiring only a bounded search space and raw function evaluations, without gradient information or exhaustive search – make it well suited to optimizing the non-smooth, black-box objective that classification accuracy constitutes as a

function of symbolic parameters. Evaluated on twenty-four UEA benchmark datasets, the proposed approach achieved competitive accuracy against state-of-the-art numerical and kernel-based baselines, demonstrating that principled parameter optimization can substantially close the gap between heuristically configured symbolic methods and more complex alternatives.

Chapter 6 established a direct structural connection between symbolic time series representations and large language models. The core observation is simple: SAX already produces ordered sequences of discrete tokens, which is precisely the input format LLMs are built to process. The chapter developed SAX_HAR-LLM, a dual-stream framework that pairs a trend-aware symbolic encoding – in which slope direction is folded into the case of the existing SAX symbol rather than carried in a separate branch, a design choice motivated by the cost of additional prompt length – with lightweight kinematics-informed descriptors that recover the physical magnitude information normalization removes. A causal LLM fine-tuned via QLoRA on structured prompts derived from inertial sensor data achieved competitive accuracy against both symbolic and deep learning baselines on two HAR benchmarks, while remaining fully auditable through an attention-based interpretability analysis that linked predictions to specific symbolic motifs on specific sensor axes.

Chapter 7 addressed BOP's loss of sequential order by replacing the frequency histogram with a graph structure that explicitly represents word-to-word transitions. Two construction schemes were proposed – Transition Graph Embeddings (TGE), connecting immediately consecutive words, and Distance-Aware Graph Embeddings (DGE), extending connectivity via BFS-based distance decay – and both were evaluated alongside BOP and their respective combinations with BOP on twelve UCR univariate benchmark datasets. The results established that graph-based embeddings complement rather than replace BOP's frequency information, with the clearest gains appearing on datasets compressed into longer symbolic strings where word transitions carry more structure.

8.2 Connecting Threads

Across these four chapters, a consistent pattern emerges. Each contribution preserves the symbolic vocabulary and asks what can be added or restructured around it, rather than abandoning the representation for one that is more expressive but less interpretable. MII and SAA enrich it at the feature construction level. The DIRECT framework refines it at the parameter selection level. SAX_HAR-LLM extends it to the classifier level, replacing a frequency-histogram classifier with a pre-trained sequence model that processes the full ordered token string. The graph-based representations of Chapter 7 recover what the histogram discards, encoding transition dependencies at the representational level itself. What unites these directions is not a shared mechanism but a shared discipline: the symbolic representation is treated as worth preserving, and the question driving each chapter is how much further it can be taken within that constraint.

A second thread connects the chapters through the concept of trend information. The trend-loss weakness identified in Chapter 3's taxonomy is answered two times across the thesis, each time through a structurally different mechanism: SAA addresses it by carrying trend in a separate, independently discretized branch, while SAX_HAR-LLM addresses it by folding trend into the case of the existing symbol, a more compact encoding suited to the prompt-length constraints of an LLM input. These are

not redundant answers to the same question: they reflect genuinely different design constraints, and the comparison between them is itself one of the thesis's contributions.

8.3 Limitations

The contributions of this thesis are subject to several limitations that qualify the scope of the claims made.

In Chapter 4, MII and SAA were each evaluated on a single HAR dataset under a fixed parameter configuration shared across both methods, and neither method was assessed in combination with the other. Whether their respective gains – cross-channel co-occurrence for MII and within-axis trend information for SAA – would compound or partially overlap when used jointly remains an open empirical question.

Chapter 6 evaluates SAX_HAR-LLM on two HAR benchmarks, both relying on smartphone-based inertial sensors, which limits the generalizability of the findings to other sensing modalities and application domains. The subject-independent evaluation uses a single 80/20 split per dataset rather than a cross-validated estimate, so the reported figures carry non-trivial variance that a subject-level k-fold protocol would better quantify. SAX parameters in this framework were fixed rather than optimized – the principled search developed in Chapter 5 was not applied here – leaving open the question of how much performance is left on the table by the chosen configuration. The trend-aware case modulation encodes slope direction as an uppercase/lowercase distinction that the Mistral-7B tokenizer treats as two entirely independent tokens, so the intended relationship between case variants is learned statistically during fine-tuning rather than being structurally guaranteed. The attention-based interpretability analysis is descriptive and correlational throughout: it characterizes where the model concentrates predictive weight, but does not establish which input features causally determine its predictions. Finally, SAX_HAR-LLM's computational demands – approximately 3.42 hours of fine-tuning and roughly 574 ms per-window inference – place it outside the reach of real-time or resource-constrained deployment scenarios.

In Chapter 7, the graph-based evaluation is confined to twelve univariate UCR datasets under a fixed SAX parameterization and a single Random Forest classifier, leaving the multivariate setting – the focus of Chapters 4 and 5 – entirely untested for the proposed constructions. More broadly, the DIRECT-based optimization of Chapter 5 and the graph construction schemes of Chapter 7 were developed independently and never combined: whether jointly optimizing SAX parameters and graph construction would yield further gains remains unexplored.

8.4 Future Directions

The limitations above define the most immediate next steps, but several broader directions also emerge from the body of work as a whole.

The most direct extension of Chapter 4 is a joint evaluation of MII and SAA in combination, together with a systematic sensitivity analysis of their shared parameter configuration. Extending both to a wider set of HAR datasets and pairing their parameter choices with the DIRECT-based optimization of Chapter 5 would place them within the same principled tuning framework developed there.

For SAX_HAR-LLM, the most immediate open question is whether the trend-aware case relationship can be made structurally explicit at the tokenization level rather than relying on statistical learning during fine-tuning. A complementary direction

is the extension of the dual-stream design to other sensing domains by replacing the kinematics-informed descriptors with domain-appropriate physical context, or even fully abandon it. Zero-shot and incremental adaptation scenarios, in which the model must generalize to unseen activities or accumulate new subject data without full retraining, are also natural targets.

For the graph-based approach, a systematic parameter definition paired with the optimization framework of Chapter 5 is the most tractable immediate step, followed by extension to multivariate time series following Chapter 4's channel-wise treatment. More fundamentally, the question of what information a SAX-word transition graph needs to carry for a GNN to become a genuinely competitive representation for downstream classification – rather than a useful complement to BOP – remains open, and is the central question this line of work intends to pursue going forward.

Symbolic representations of time series have long occupied a distinctive position in the machine learning literature. They are theoretically well-founded, computationally efficient, and inherently interpretable, yet their empirical performance has often been underestimated due to evaluation settings that rely on heuristic parameter choices and simple baseline classifiers. Through four complementary contributions, this thesis has shown that the symbolic representation itself is not the primary limiting factor. Rather, its performance depends critically on the methods and learning frameworks that exploit it. While the results presented here demonstrate the potential of symbolic representations across multiple settings, they also indicate that significant opportunities remain for further methodological and empirical advances.

This page has been left blank intentionally.

List of Publications

Journal Publications

[**J1**] **L. Pappa**, P. Karvelis and C. Stylios, "Exploring the diverse world of SAX-based methodologies," *Data Mining and Knowledge Discovery*, vol. 39, no. 1, p. 4, 2025. doi: 10.1007/s10618-024-01075-2.

[**J2**] **L. Pappa**, P. Karvelis and C. Stylios, "Optimized symbolic aggregate approximation methods for lightweight and interpretable multivariate time series classification," *Information Sciences*, vol. 755, Art. no. 123819, Nov. 2026. doi: 10.1016/j.ins.2026.123819.

[**J3**] **L. Pappa**, P. Karvelis and C. Stylios, "Bridging Time Series and Large Language Models via Symbolic Representation for Human Activity Recognition," *Expert Systems With Applications*, vol. 332, Part A, Art. no. 133478, Jan. 2027, doi: 10.1016/j.eswa.2026.133478.

[**J4**] **L. Pappa**, G. Georgoulas, I. Kosti, P. Karvelis and C. Stylios, "Annotation-agnostic legal retrieval: An unsupervised framework for Greek legal document similarity," submitted to *Artificial Intelligence and Law*, 2026.

[**J5**] V. Liagkou, C. Stylios, **L. Pappa** and A. Petunin, "Challenges and opportunities in Industry 4.0 for mechatronics, artificial intelligence and cybernetics," *Electronics*, vol. 10, no. 16, p. 2001, 2021. doi: 10.3390/electronics10162001.

Conference Publications (Peer Reviewed)

[**C1**] **L. Pappa**, P. Karvelis, G. Georgoulas and C. Stylios, "Multichannel symbolic aggregate approximation intelligent icons: Application for activity recognition," in *2020 IEEE Symp. Series on Computational Intelligence (SSCI)*, Canberra, ACT, Australia, 2020, pp. 505–512. doi: 10.1109/SSCI47803.2020.9308497.

[**C2**] **L. Pappa**, P. Karvelis, G. Georgoulas and C. Stylios, "Slopewise aggregate approximation SAX: Keeping the trend of a time series," in *2021 IEEE Symp. Series on Computational Intelligence (SSCI)*, Orlando, FL, USA, 2021, pp. 1–8. doi: 10.1109/SSCI50451.2021.9660130.

[**C3**] **L. Pappa**, P. Karvelis and C. Stylios, "A comparative study on recognizing human activities by applying diverse machine learning approaches," in *2022 44th Annu. Int. Conf. of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Glasgow, Scotland, United Kingdom, 2022, pp. 3661–3664. doi: 10.1109/EMBC48229.2022.9871324.

[**C4**] **L. Pappa**, P. Karvelis, G. Georgoulas and C. Stylios, "SAX-based GNN embeddings for time series," in *2025 25th Int. Conf. on Digital Signal Processing (DSP)*, Pylos, Greece, 2025, pp. 1–5. doi: 10.1109/DSP65409.2025.11075189.

[**C5**] G.-P. Botilias, **L. Pappa**, P. Karvelis and C. Stylios, "Tracking individuals' health using mobile applications and machine learning," in *2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conf. (SEEDA-CECNSM)*, Ioannina, Greece, 2022, pp. 1–6. doi: 10.1109/SEEDA-CECNSM57760.2022.9932927.

[**C6**] **L. Pappa**, P. Karvelis and C. Stylios, "A review of automatic text summarization: The case of Greek legal documents," in *2025 10th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conf. (SEEDA-CECNSM)*, Patras, Greece, 2025, pp. 1–6. doi: 10.1109/SEEDA-CECNSM68644.2025.11329594.

Bibliography

[1] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[2] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

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

[7] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[8] A. Pasos Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 35, no. 2, pp. 401–449, 2021.

[9] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, pp. 3111–3119, 2013.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, pp. 1097–1105, 2012.

[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[14] P. P. Liang, A. Zadeh, and L.-P. Morency, "Foundations & trends in multimodal machine learning: Principles, challenges, and open questions," *ACM computing surveys*, vol. 56, no. 10, pp. 1–42, 2024.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, OpenReview.net, 2021.

[16] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[17] R. Bommasani *et al.*, "On the opportunities and risks of foundation models," 2021.

[18] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[19] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 5998–6008, 2017.

[26] T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.

[27] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 27730–27744, Curran Associates Inc., 2022.

[28] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," 2020.

[29] J. Hoffmann, S. Borgeaud, A. Mensch, *et al.*, "Training compute-optimal large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, Curran Associates Inc., 2022.

[30] H. Touvron, T. Lavril, G. Izacard, *et al.*, "LLaMA: Open and efficient foundation language models," 2023.

[31] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, vol. 2, pp. 2672–2680, MIT Press, 2014.

[32] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[33] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[34] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[35] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–35, 2021.

[36] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 308–318, 2016.

[37] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the 3rd International Conference on Learning Representations, (ICLR)*, 2015.

[38] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, June 2020.

[39] F. J. Baldán and J. M. Benítez, "Multivariate times series classification through an interpretable representation," *Information Sciences*, vol. 569, pp. 596–614, Aug. 2021.

[40] W. Mahmud, A. Z. Fanani, H. A. Santoso, and F. Budiman, "Review of Time Series Classification Techniques and Methods," in *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pp. 452–457, Sept. 2023.

[41] D. Yao and K. Yan, "Time series forecasting of stock market indices based on dlwr-lstm model," *Finance Research Letters*, vol. 68, p. 105821, 2024.

[42] H. Saleh, S. El-Sappagh, M. McCann, S. H. Alsamhi, and J. G. Breslin, "Multivariate multi-horizon time-series forecasting for real-time patient monitoring based on cascaded fine tuning of attention-based models," *Computers in Biology and Medicine*, vol. 194, p. 110406, 2025.

[43] J. D. Willard, C. Varadharajan, X. Jia, and V. Kumar, "Time series predictions in unmonitored sites: A survey of machine learning techniques in water resources," *Environmental Data Science*, vol. 4, p. e7, 2025.

[44] P. Yan, A. Abdulkadir, P.-P. Luley, M. Rosenthal, G. A. Schatte, B. F. Grewe, and T. Stadelmann, "A comprehensive survey of deep transfer learning for anomaly detection in industrial time series: Methods, applications, and directions," *IEEE Access*, vol. 12, pp. 3768–3789, 2024.

[45] M. Attioui and M. Lahby, "A systematic literature review of traffic congestion forecasting: From machine learning techniques to large language models," *Vehicles*, vol. 7, no. 4, p. 142, 2025.

[46] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, pp. 1–34, 2012.

[47] S. Alaee, K. Kamgar, and E. Keogh, "Matrix profile xxii: exact discovery of time series motifs under dtw," in *2020 IEEE international conference on data mining (ICDM)*, pp. 900–905, IEEE, 2020.

[48] S. Alaee, R. Mercer, K. Kamgar, and E. Keogh, "Time series motifs discovery under dtw allows more robust discovery of conserved structure," *Data Mining and Knowledge Discovery*, vol. 35, no. 3, pp. 863–910, 2021.

[49] V. Bettaiah and H. S. Ranganath, "An analysis of time series representation methods: data mining applications perspective," in *Proceedings of the 2014 ACM Southeast Conference*, pp. 1–6, 2014.

[50] J. Shieh and E. Keogh, "iSAX: indexing and mining terabyte sized time series," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, (New York, NY, USA), pp. 623–631, Association for Computing Machinery, May 2008.

[51] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering–a decade review," *Information systems*, vol. 53, pp. 16–38, 2015.

[52] Y. Li and D. Shen, "A new symbolic representation method for time series," *Information Sciences*, vol. 609, pp. 276–303, 2022.

[53] X. Bai, Y. Xiong, Y. Zhu, and H. Zhu, "Time Series Representation: A Random Shifting Perspective," in *Web-Age Information Management* (J. Wang, H. Xiong,

Y. Ishikawa, J. Xu, and J. Zhou, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 37–50, Springer, 2013.

[54] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and knowledge discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[55] L. Pappa, P. Karvelis, and C. Stylios, "Exploring the diverse world of sax-based methodologies," *Data Mining and Knowledge Discovery*, vol. 39, no. 1, p. 4, 2025.

[56] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery - DMKD '03*, (San Diego, California), p. 2, ACM Press, 2003.

[57] C. Ratanamahatana, E. Keogh, A. J. Bagnall, and S. Lonardi, "A novel bit level time series representation with implication of similarity search and clustering," in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 771–777, Springer, 2005.

[58] C. Zhang, Y. Chen, A. Yin, and X. Wang, "Anomaly detection in ecg based on trend symbolic aggregate approximation," *Mathematical Biosciences and Engineering*, vol. 16, no. 4, p. 2154, 2019.

[59] A. Mueen, "Time series motif discovery: dimensions and applications," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 2, pp. 152–159, 2014.

[60] J. Lin, E. Keogh, and S. Lonardi, "Visualizing and discovering non-trivial patterns in large time series databases," *Information visualization*, vol. 4, no. 2, pp. 61–82, 2005.

[61] L. Pappa, P. Karvelis, G. Georgoulas, and C. Stylios, "Slopewise aggregate approximation sax: keeping the trend of a time series," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 01–08, IEEE, 2021.

[62] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and information Systems*, vol. 3, no. 3, pp. 263–286, 2001.

[63] J. Lin, R. Khade, and Y. Li, "Rotation-invariant similarity in time series using bag-of-patterns representation," *Journal of Intelligent Information Systems*, vol. 39, no. 2, pp. 287–315, 2012.

[64] H. Yin, S. Yang, X. Zhu, S. Ma, and L. Chen, "Symbolic representation based on trend features for biomedical data classification," *Technology and Health Care*, vol. 23, no. 2_suppl, pp. S501–S510, 2015.

[65] A. S. Dhanjal and W. Singh, "A comprehensive survey on automatic speech recognition using neural networks," *Multimedia Tools and Applications*, vol. 83, pp. 23367–23412, Mar. 2024. Number: 8.

[66] S. S. W. Fatima and A. Rahimi, "A Review of Time-Series Forecasting Algorithms for Industrial Manufacturing Systems," *Machines*, vol. 12, p. 380, June 2024. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[67] J. Xie, Z. Wang, Z. Yu, Y. Ding, and B. Guo, "Prototype Learning for Medical Time Series Classification via Human–Machine Collaboration," *Sensors (Basel, Switzerland)*, vol. 24, p. 2655, Apr. 2024. Number: 8.

[68] X. Yu, Z. Chen, Y. Ling, S. Dong, Z. Liu, and Y. Lu, "Temporal Data Meets LLM – Explainable Financial Time Series Forecasting," June 2023. Issue: arXiv:2306.11025 arXiv:2306.11025 [cs].

[69] C. Zhang, N. N. A. Sjarif, and R. Ibrahim, "Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022," *WIREs Data Mining and Knowledge Discovery*, vol. 14, no. 1, p. e1519, 2024. Number: 1 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1519.

[70] V. Papastefanopoulos, P. Linardatos, T. Panagiotakopoulos, and S. Kotsiantis, "Multivariate Time-Series Forecasting: A Review of Deep Learning Methods in Internet of Things Applications to Smart Cities," *Smart Cities*, vol. 6, pp. 2519–2552, Oct. 2023. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[71] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 35, pp. 401–449, Mar. 2021. Number: 2.

[72] M. A. Farahani, M. R. McCormick, R. Harik, and T. Wuest, "Time-series classification in smart manufacturing systems: An experimental evaluation of state-of-the-art machine learning algorithms," *Robotics and Computer-Integrated Manufacturing*, vol. 91, p. 102839, Feb. 2025.

[73] M. Du, Y. Wei, X. Zheng, and C. Ji, "Multi-feature based network for multivariate time series classification," *Information Sciences*, vol. 639, p. 119009, Aug. 2023.

[74] Z. Xiao, H. Xing, R. Qu, L. Feng, S. Luo, P. Dai, B. Zhao, and Y. Dai, "Densely Knowledge-Aware Network for Multivariate Time Series Classification," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, pp. 2192–2204, Apr. 2024. Number: 4 Conference Name: IEEE Transactions on Systems, Man, and Cybernetics: Systems.

[75] C. Yang, X. Wang, L. Yao, G. Long, and G. Xu, "Dyformer: A dynamic transformer-based architecture for multivariate time series classification," *Information Sciences*, vol. 656, p. 119881, Jan. 2024.

[76] R. Younis, A. Hakmeh, and Z. Ahmadi, "MTS2Graph: Interpretable multivariate time series classification with temporal evolving graphs," *Pattern Recognition*, vol. 152, p. 110486, Aug. 2024.

[77] P. Schäfer and U. Leser, "WEASEL 2.0: a random dilated dictionary transform for fast, accurate and memory constrained time series classification," *Machine Learning*, vol. 112, pp. 4763–4788, Dec. 2023.

[78] A. Glenis and G. A. Vouros, "SCALE-BOSS: A framework for scalable time-series classification using symbolic representations," in *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*, (Corfu Greece), pp. 1–9, ACM, Sept. 2022.

[79] A. Glenis and G. A. Vouros, "SCALE-BOSS-MR: Scalable Time Series Classification Using Multiple Symbolic Representations," *Applied Sciences*, vol. 14, p. 689, Jan. 2024. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[80] F. Spinnato, R. Guidotti, A. Monreale, and M. Nanni, "Fast, Interpretable, and Deterministic Time Series Classification With a Bag-of-Receptive-Fields," *IEEE Access*, vol. 12, pp. 137893–137912, 2024. Conference Name: IEEE Access.

[81] B. Dhariyal, T. Le Nguyen, and G. Ifrim, "Back to Basics: A Sanity Check on Modern Time Series Classification Algorithms," in *Advanced Analytics and Learning on Temporal Data* (G. Ifrim, R. Tavenard, A. Bagnall, P. Schaefer, S. Malinowski, T. Guyet, and V. Lemaire, eds.), (Cham), pp. 205–229, Springer Nature Switzerland, 2023.

[82] Z. Lee, T. Lindgren, and P. Papapetrou, "Z-Time: efficient and effective interpretable multivariate time series classification," *Data Mining and Knowledge Discovery*, vol. 38, pp. 206–236, Jan. 2024. Number: 1.

[83] A. P. Ruiz, M. Flynn, and A. Bagnall, "Benchmarking Multivariate Time Series Classification Algorithms," July 2020.

[84] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The UEA multivariate time series classification archive, 2018," Oct. 2018. Issue: arXiv:1811.00075 arXiv:1811.00075 [cs, stat].

[85] M. Middlehurst, J. Large, and A. Bagnall, "The Canonical Interval Forest (CIF) Classifier for Time Series Classification," in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 188–195, Sept. 2020.

[86] P. Schäfer, "The boss is concerned with time series classification in the presence of noise," *Data Mining and Knowledge Discovery*, vol. 29, no. 6, pp. 1505–1530, 2015.

[87] P. Schäfer and U. Leser, "Fast and Accurate Time Series Classification with WEASEL," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 637–646, Nov. 2017. arXiv:1701.07681 [cs].

[88] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 43–49, Feb. 1978. Number: 1.

[89] L. Bergroth, H. Hakonen, and T. Raita, "A survey of longest common subsequence algorithms," in *Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000*, pp. 39–48, Sept. 2000.

[90] B. Ghojogh and M. Crowley, "Linear and Quadratic Discriminant Analysis: Tutorial," June 2019. Issue: arXiv:1906.02590 arXiv:1906.02590 [stat].

[91] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, Aug. 2016. arXiv:1603.02754 [cs].

[92] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)," *Neurocomputing*, vol. 307, pp. 72–77, Sept. 2018.

[93] A. Dempster, F. Petitjean, and G. I. Webb, "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, pp. 1454–1495, Sept. 2020. Number: 5.

[94] A. Dempster, D. F. Schmidt, and G. I. Webb, "MINIROCKET: A Very Fast (Almost) Deterministic Transform for Time Series Classification," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 248–257, Aug. 2021. arXiv:2012.08791 [cs].

[95] C. W. Tan, A. Dempster, C. Bergmeir, and G. I. Webb, "MultiRocket: Multiple pooling operators and transformations for fast and effective time series classification," Feb. 2022. Issue: arXiv:2102.00457 arXiv:2102.00457 [cs].

[96] A. Dempster, D. F. Schmidt, and G. I. Webb, "Hydra: competing convolutional kernels for fast and accurate time series classification," *Data Mining and Knowledge Discovery*, vol. 37, pp. 1779–1805, Sept. 2023. Number: 5.

[97] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM Fully Convolutional Networks for Time Series Classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018. arXiv:1709.05206 [cs].

[98] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015. Issue: arXiv:1512.03385 arXiv:1512.03385 [cs].

[99] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "InceptionTime: Finding AlexNet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, pp. 1936–1962, Nov. 2020. Number: 6.

[100] A. Ismail-Fawaz, M. Devanne, S. Berretti, J. Weber, and G. Forestier, "Look into the LITE in deep learning for time series classification," *International Journal of Data Science and Analytics*, Jan. 2025.

[101] J. Lines, S. Taylor, and A. Bagnall, "HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles for Time Series Classification," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1041–1046, Sept. 2016. ISSN: 2374-8486.

[102] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, "HIVE-COTE 2.0: a new meta ensemble for time series classification," *Machine Learning*, vol. 110, pp. 3211–3243, Dec. 2021. Number: 11-12 arXiv:2104.07551 [cs].

[103] A. Shifaz, C. Pelletier, F. Petitjean, and G. I. Webb, "TS-CHIEF: a scalable and accurate forest algorithm for time series classification," *Data Min. Knowl. Discov.*, vol. 34, pp. 742–775, Feb. 2020. Number: 3.

[104] M. Middlehurst, J. Large, G. Cawley, and A. Bagnall, "The Temporal Dictionary Ensemble (TDE) Classifier for Time Series Classification," in *Machine Learning and Knowledge Discovery in Databases* (F. Hutter, K. Kersting, J. Lijffijt, and I. Valera, eds.), (Cham), pp. 660–676, Springer International Publishing, 2021.

[105] P. Senin and S. Malinchik, "Sax-vsm: Interpretable time series classification using sax and vector space model," in *2013 IEEE 13th international conference on data mining*, pp. 1175–1180, IEEE, 2013.

[106] L. G. Pappa, P. Karvelis, and C. D. Stylios, "A comparative study on recognizing human activities by applying diverse machine learning approaches," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3661–3664, IEEE, 2022.

[107] P. Schäfer and M. Högqvist, "Sfa: a symbolic fourier approximation and index for similarity search in high dimensional datasets," in *Proceedings of the 15th international conference on extending database technology*, pp. 516–527, 2012.

[108] P. Schäfer, "Scalable time series classification," *Data Mining and Knowledge Discovery*, vol. 30, no. 5, pp. 1273–1298, 2016.

[109] T. L. Nguyen and G. Ifrim, "Fast Time Series Classification with Random Symbolic Subsequences," in *Advanced Analytics and Learning on Temporal Data* (T. Guyet, G. Ifrim, S. Malinowski, A. Bagnall, P. Shafer, and V. Lemaire, eds.), (Cham), pp. 50–65, Springer International Publishing, 2023.

[110] M. Middlehurst, P. Schäfer, and A. Bagnall, "Bake off redux: a review and experimental evaluation of recent time series classification algorithms," *Data Mining and Knowledge Discovery*, vol. 38, pp. 1958–2031, July 2024. Number: 4.

[111] T. Le Nguyen, S. Gsponer, I. Ilie, M. O'reilly, and G. Ifrim, "Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations," *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 1183–1222, 2019.

[112] L. Pappa, P. Karvelis, G. Georgoulas, and C. Stylios, "Sax-based gnn embeddings for time series," in *2025 25th International Conference on Digital Signal Processing (DSP)*, pp. 1–5, IEEE, 2025.

[113] M. Flynn, J. Large, and T. Bagnall, "The Contract Random Interval Spectral Ensemble (c-RISE): The Effect of Contracting a Classifier on Accuracy," in *Hybrid Artificial Intelligent Systems: 14th International Conference, HAIS 2019, León, Spain, September 4–6, 2019, Proceedings*, (Berlin, Heidelberg), pp. 381–392, Springer-Verlag, June 2019.

[114] A. Dempster, D. F. Schmidt, and G. I. Webb, "QUANT: A Minimalist Interval Method for Time Series Classification," Aug. 2023. Issue: arXiv:2308.00928 arXiv:2308.00928 [cs].

[115] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, "Classification of time series by shapelet transformation," *Data Min. Knowl. Discov.*, vol. 28, pp. 851–881, Apr. 2014. Number: 4.

[116] A. Guillaume, C. Vrain, and W. Elloumi, "Random Dilated Shapelet Transform: A New Approach for Time Series Shapelets," in *Pattern Recognition and Artificial Intelligence* (M. El Yacoubi, E. Granger, P. C. Yuen, U. Pal, and N. Vincent, eds.), (Cham), pp. 653–664, Springer International Publishing, 2022.

[117] I. Karlsson, P. Papapetrou, and H. Boström, "Generalized random shapelet forests," *Data Mining and Knowledge Discovery*, vol. 30, pp. 1053–1085, Sept. 2016. Number: 5.

[118] M. Shokoohi-Yekta, J. Wang, and E. Keogh, "On the Non-Trivial Generalization of Dynamic Time Warping to the Multi-Dimensional Case," in *Proceedings of the 2015 SIAM International Conference on Data Mining (SDM)*, Proceedings, pp. 289–297, Society for Industrial and Applied Mathematics, June 2015.

[119] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh, "Generalizing DTW to the multi-dimensional case requires an adaptive approach," *Data mining and knowledge discovery*, vol. 31, pp. 1–31, Jan. 2017. Number: 1.

[120] J. Mei, M. Liu, Y.-F. Wang, and H. Gao, "Learning a Mahalanobis Distance-Based Dynamic Time Warping Measure for Multivariate Time Series Classification," *IEEE Transactions on Cybernetics*, vol. 46, pp. 1363–1374, June 2016. Number: 6.

[121] A. Bostrom and A. Bagnall, "A Shapelet Transform for Multivariate Time Series Classification," Dec. 2017. Issue: arXiv:1712.06428 arXiv:1712.06428 [cs].

[122] M. Wistuba, J. Grabocka, and L. Schmidt-Thieme, "Ultra-Fast Shapelets for Time Series Classification," Mar. 2015. Issue: arXiv:1503.05018 arXiv:1503.05018 [cs].

[123] P. Schäfer and U. Leser, "Multivariate Time Series Classification with WEASEL+MUSE," Aug. 2018. arXiv:1711.11343 [cs].

[124] W. Homenda, A. Jastrzębska, W. Pedrycz, and M. Wrzesień, "Time series classification with their image representation," *Neurocomputing*, vol. 573, p. 127214, Mar. 2024.

[125] M. G. Baydogan and G. Runger, "Learning a symbolic representation for multivariate time series classification," *Data Mining and Knowledge Discovery*, vol. 29, pp. 400–422, Mar. 2015. Number: 2.

[126] M. G. Baydogan and G. Runger, "Time series representation and similarity based on local autopatterns," *Data Mining and Knowledge Discovery*, vol. 30, pp. 476–509, Mar. 2016. Number: 2.

[127] K. S. Tuncel and M. G. Baydogan, "Autoregressive forests for multivariate time series modeling," *Pattern Recognition*, vol. 73, pp. 202–215, Jan. 2018.

[128] L. Feremans, B. Cule, and B. Goethals, "PETSC: pattern-based embedding for time series classification," *Data Mining and Knowledge Discovery*, vol. 36, pp. 1015–1061, May 2022. Number: 3.

[129] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Networks*, vol. 116, pp. 237–245, Aug. 2019.

[130] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu, "TapNet: Multivariate Time Series Classification with Attentional Prototypical Network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 6845–6852, Apr. 2020. Number: 04.

[131] Z. Xiao, X. Xu, H. Xing, S. Luo, P. Dai, and D. Zhan, "RTFN: A robust temporal feature network for time series classification," *Information Sciences*, vol. 571, pp. 65–86, Sept. 2021.

[132] Y. Wang, Z. Duan, Y. Huang, H. Xu, J. Feng, and A. Ren, "MTHetGNN: A heterogeneous graph embedding framework for multivariate time series forecasting," *Pattern Recognition Letters*, vol. 153, pp. 151–158, Jan. 2022.

[133] Z. Duan, H. Xu, Y. Huang, J. Feng, and Y. Wang, "Multivariate Time Series Forecasting with Transfer Entropy Graph," *Tsinghua Science and Technology*, vol. 28, pp. 141–149, Feb. 2023. Number: 1.

[134] K. Fauvel, E. Fromont, V. Masson, P. Faverdin, and A. Termier, "Local Cascade Ensemble for Multivariate Data Classification," *ArXiv*, May 2020.

[135] K. Fauvel, E. Fromont, V. Masson, P. Faverdin, and A. Termier, "XEM: An explainable-by-design ensemble method for multivariate time series classification," *Data Mining and Knowledge Discovery*, vol. 36, pp. 917–957, May 2022. Number: 3.

[136] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, *et al.*, "The prisma 2020 statement: an updated guideline for reporting systematic reviews," *bmj*, vol. 372, 2021.

[137] M. Butler and D. Kazakov, "Sax discretization does not guarantee equiprobable symbols," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 1162–1166, 2014.

[138] N. D. Pham, Q. L. Le, and T. K. Dang, "Two Novel Adaptive Symbolic Representations for Similarity Search in Time Series Databases," in *2010 12th International Asia-Pacific Web Conference*, pp. 181–187, Apr. 2010.

[139] M. M. Muhammad Fuad, "Genetic algorithms-based symbolic aggregate approximation," in *International Conference on Data Warehousing and Knowledge Discovery*, pp. 105–116, Springer, 2012.

[140] W. Zalewski, F. Silva, F. C. Wu, H. D. Lee, and A. G. Maletzke, "A symbolic representation method to preserve the characteristic slope of time series," in *Brazilian Symposium on Artificial Intelligence*, pp. 132–141, Springer, 2012.

[141] R. Rezvani, P. Barnaghi, and S. Enshaeifar, "A new pattern representation method for time-series data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 7, pp. 2818–2832, 2019.

[142] N. Pavlopoulou and E. Curry, "Iotsax: A dynamic abstractive entity summarization approach with approximation and embedding-based reasoning rules in publish/subscribe systems," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1830–1847, 2021.

[143] M. Kloska and V. Rozinajova, "Distribution-wise symbolic aggregate approximation (dwsax)," in *International conference on intelligent data engineering and automated learning*, pp. 304–315, Springer, 2020.

[144] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "New time series data representation esax for financial applications," in *22nd International Conference on Data Engineering Workshops (ICDEW'06)*, pp. x115–x115, IEEE, 2006.

[145] A. Bondu, M. Boullé, and B. Grossin, "Saxo: An optimized data-driven symbolic representation of time series," in *The 2013 international joint conference on neural networks (IJCNN)*, pp. 1–9, IEEE, 2013.

[146] K. Zhang, Y. Li, Y. Chai, and L. Huang, "Trend-based symbolic aggregate approximation for time series representation," in *2018 Chinese Control And Decision Conference (CCDC)*, pp. 2234–2240, IEEE, 2018.

[147] B. Bai, G. Li, S. Wang, Z. Wu, and W. Yan, "Time series classification based on multi-feature dictionary representation and ensemble learning," *Expert Systems with Applications*, vol. 169, p. 114162, 2021.

[148] N. Tabassum, S. Menon, and A. Jastrzębska, "Time-series classification with safe: Simple and fast segmented word embedding-based neural time series classifier," *Information Processing & Management*, vol. 59, no. 5, p. 103044, 2022.

[149] X. Bai, Y. Xiong, Y. Zhu, and H. Zhu, "Time series representation: a random shifting perspective," in *International Conference on Web-Age Information Management*, pp. 37–50, Springer, 2013.

[150] N. D. Pham, Q. L. Le, and T. K. Dang, "Two novel adaptive symbolic representations for similarity search in time series databases," in *2010 12th international asia-pacific web conference*, pp. 181–187, IEEE, 2010.

[151] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[152] S. Elsworth and S. Güttel, "Abba: adaptive brownian bridge-based symbolic aggregation of time series," *Data Mining and Knowledge Discovery*, vol. 34, no. 4, pp. 1175–1200, 2020.

[153] M. M. M. Fuad, "Genetic Algorithms-Based Symbolic Aggregate Approximation," in *Data Warehousing and Knowledge Discovery* (A. Cuzzocrea and U. Dayal, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 105–116, Springer, 2012.

[154] M. M. M. Fuad, "Differential evolution versus genetic algorithms: towards symbolic aggregate approximation of non-normalized time series," in *Proceedings of the 16th International Database Engineering & Applications Sysmposium*, IDEAS '12, (New York, NY, USA), pp. 205–210, Association for Computing Machinery, May 2012.

[155] O. Aremu, D. Hyland-Wood, and P. McAree, "A Relative Entropy Weibull-SAX framework for health indices construction and health stage division in degradation modeling of multivariate time series asset data," *Advanced Engineering Informatics*, vol. 40, pp. 121–134, 2019.

[156] S. Malinowski, T. Guyet, R. Quiniou, and R. Tavenard, "1d-SAX: A Novel Symbolic Representation for Time Series," in *Advances in Intelligent Data Analysis XII* (A. Tucker, F. Höppner, A. Siebes, and S. Swift, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 273–284, Springer, 2013.

[157] Y. Sun, J. Li, J. Liu, B. Sun, and C. Chow, "An improvement of symbolic aggregate approximation distance measure for time series," *Neurocomputing*, vol. 138, pp. 189–198, Aug. 2014.

[158] F. Ganz, P. Barnaghi, and F. Carrez, "Information Abstraction for Heterogeneous Real World Internet Data," *IEEE Sensors Journal*, vol. 13, pp. 3793–3805, July 2013. Conference Name: IEEE Sensors Journal.

[159] T. L. Nguyen, S. Gsponer, and G. Ifrim, "Time Series Classification by Sequence Learning in All-Subsequence Space," in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 947–958, Apr. 2017.

[160] L. Klus, E. Lohan, C. Granell, and J. Nurmi, "Lossy compression methods for performance-restricted wearable devices," vol. 2626, 2020.

[161] P. Ordonez, T. Armstrong, T. Oates, and J. Fackler, "Using Modified Multivariate Bag-of-Words Models to Classify Physiological Data," in *2011 IEEE 11th International Conference on Data Mining Workshops*, pp. 534–539, Sept. 2011.

[162] H. Park and J.-Y. Jung, "Sax-arm: Deviant event pattern discovery from multivariate time series using symbolic aggregate approximation and association rule mining," *Expert Systems with Applications*, vol. 141, p. 112950, 2020.

[163] Y. Mohammad and T. Nishida, "Robust learning from demonstrations using multidimensional SAX," in *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, pp. 64–71, July 2014.

[164] J. Shieh and E. Keogh, "i sax: indexing and mining terabyte sized time series," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 623–631, 2008.

[165] L. Pappa, P. Karvelis, G. Georgoulas, and C. Stylios, "Multichannel symbolic aggregate approximation intelligent icons: Application for activity recognition," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 505–512, IEEE, 2020.

[166] E. Keogh, L. Wei, X. Xi, S. Lonardi, J. Shieh, and S. Sirowy, "Intelligent icons: Integrating lite-weight data mining and visualization into gui operating systems," in *Sixth International Conference on Data Mining (ICDM'06)*, pp. 912–916, IEEE, 2006.

[167] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *2016 IEEE international conference on pervasive computing and communications (PerCom)*, pp. 1–9, IEEE, 2016.

[168] L. Pappa, P. Karvelis, and C. Stylios, "Optimized symbolic aggregate approximation methods for lightweight and interpretable multivariate time series classification," *Information Sciences*, vol. 755, p. 123819, 11 2026.

[169] Y. Bao and W. Chen, "Automated concept extraction in internet-of-things," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1770–1776, IEEE, 2018.

[170] M. Anacleto, S. Vinga, and A. M. Carvalho, "Msax: Multivariate symbolic aggregate approximation for time series classification," in *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pp. 90–97, Springer, 2019.

[171] L. Zhang, T. Pei, B. Meng, Y. Lian, and Z. Jin, "Two-phase multivariate time series clustering to classify urban rail transit stations," *Ieee Access*, vol. 8, pp. 167998–168007, 2020.

[172] V. Jha and P. Tripathi, "Semantic Modelling of Multivariate Time-Series Data in Cognitive IoT," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, pp. 943–948, Mar. 2023. ISSN: 2575-7288.

[173] Y. Yu, T. Becker, L. M. Trinh, and M. Behrisch, "SAXRegEx: Multivariate time series pattern search with symbolic representation, regular expression, and query expansion," *Computers & Graphics*, vol. 112, pp. 13–21, May 2023. Publisher: Elsevier BV.

[174] G. Chatzigeorgakidis, K. Lentzos, and D. Skoutas, "MultiCast: Zero-Shot Multivariate Time Series Forecasting Using LLMs," in *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, pp. 119–127, Feb. 2024. ISSN: 2473-3490.

[175] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, pp. 157–181, Oct. 1993. Number: 1.

[176] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, Oct. 2001.

[177] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1–30, 2006. Number: 1.

[178] C. A. Ratanamahatana and E. Keogh, "Everything you know about dynamic time warping is wrong," in *Third workshop on mining temporal and sequential data*, vol. 32, pp. 1–11, Citeseer Seattle, 2004.

[179] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in *Proceedings of the 2005 SIAM international conference on data mining*, pp. 506–510, SIAM, 2005.

[180] L. Pappa, P. Karvelis, and C. Stylios, "Bridging time series and large language models via symbolic representation for human activity recognition," *Expert Systems with Applications*, vol. 332, p. 133478, 1 2027.

[181] L. Ferrone and F. M. Zanzotto, "Symbolic, Distributed, and Distributional Representations for Natural Language Processing in the Era of Deep Learning: A Survey," *Frontiers in Robotics and AI*, vol. 6, Jan. 2020. Publisher: Frontiers.

[182] Z. Zhu, M. Galkin, Z. Zhang, and J. Tang, "Neural-symbolic models for logical queries on knowledge graphs," in *Proceedings of the 39th International Conference on Machine Learning* (K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds.), vol. 162 of *Proceedings of Machine Learning Research*, pp. 27454–27478, PMLR, 2022.

[183] L. S. Lorello, M. Lippi, and S. Melacci, "A Neuro-Symbolic Framework for Sequence Classification with Relational and Temporal Knowledge," vol. 1, pp. 5833–5841, Sept. 2025. ISSN: 1045-0823.

[184] M. Asai and C. Muise, "Discrete word embedding for logical natural language understanding: Extended abstract," in *Proceedings of the Workshop on Knowledge Engineering for Planning and Scheduling (KEPS 2020)*, AAAI, 2020. Submission 9.

[185] X. Wu, Y.-L. Li, J. Sun, and C. Lu, "Symbol-LLM: leverage language models for symbolic system in visual human activity reasoning," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, (Red Hook, NY, USA), pp. 29680–29691, Curran Associates Inc., Sept. 2023.

[186] U. Nawaz, M. Anees-ur Rahaman, and Z. Saeed, "A review of neuro-symbolic AI integrating reasoning and learning for advanced cognitive systems," *Intelligent Systems with Applications*, vol. 26, p. 200541, June 2025.

[187] D. Yu, B. Yang, D. Liu, H. Wang, and S. Pan, "A survey on neural-symbolic learning systems," *Neural Netw.*, vol. 166, pp. 105–126, June 2023.

[188] D. M. Onchis and E.-F. Hogea, "Logic Meets Attention: A Neuro-symbolic Approach to Vibration Fault Detection," in *Proceedings of the 1st International Workshop on Advanced Neuro-Symbolic Applications (ANSyA 2025) co-located with the 28th European Conference on Artificial Intelligence (ECAI 2025)* (A. Agiollo, E. Bardhi, G. Ciatto, S. Dumancic, and G. Marra, eds.), vol. 4125 of *CEUR Workshop Proceedings*, (Bologna, Italy), pp. 21–24, CEUR-WS.org, oct 2025.

[189] C. Zhang, Y. Wang, and X. You, "Fault diagnosis in rotating machinery with discretized signal representation leveraging large language models," *Applied Soft Computing*, vol. 189, p. 114487, Mar. 2026.

[190] J. Miao, C. Thongprayoon, S. Suppadungsuk, P. Krisanapan, Y. Radhakrishnan, W. Cheungpasitporn, J. Miao, C. Thongprayoon, S. Suppadungsuk, P. Krisanapan, Y. Radhakrishnan, and W. Cheungpasitporn, "Chain of Thought Utilization in Large Language Models and Application in Nephrology," *Medicina*, vol. 60, Jan. 2024.

[191] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, (Red Hook, NY, USA), pp. 24824–24837, Curran Associates Inc., Aug. 2022.

[192] C. Qi, R. Ma, B. Li, H. Du, B. Hui, J. Wu, Y. Laili, and C. He, "Large language models meet symbolic provers for logical reasoning evaluation," in *The Thirteenth International Conference on Learning Representations*, 2025.

[193] L. Pan, A. Albalak, X. Wang, and W. Wang, "Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, (Singapore), pp. 3806–3824, Association for Computational Linguistics, Dec. 2023.

[194] J. Song, M. E. Akhter, D. Atzil-Slonim, and M. Liakata, "Temporal reasoning for timeline summarisation in social media," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28085–28101, 2025.

[195] S. Wu, Y. Zhang, Y. Yang, Z. Li, G. Yu, L. Chen, J. Xu, and A. Wulamu, "Lgtime: Leveraging llms with feature-aware processing and multi-granularity fusion for zero-shot time series forecasting," *Expert Systems with Applications*, p. 129483, 2025.

[196] T. Chen, X. Ma, Y. Xu, Y. Xu, S. Qian, and L. Cui, "Rts-llm: Restoring time structure for time series forecasting with llms," *Expert Systems with Applications*, p. 130402, 2025.

[197] H. Guo, P. Y. Kwok, Y. Guo, J. Zhao, and D. Gu, "Finlspm: Large stock predict model via numerical prior knowledge from llm," *Expert Systems with Applications*, p. 130294, 2025.

[198] E. Boix-Adsera, O. Saremi, E. Abbe, S. Bengio, E. Littwin, and J. Susskind, "When can transformers reason with abstract symbols?," Apr. 2024.

[199] D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu, "TEMPO: Prompt-based generative pre-trained transformer for time series forecasting," in *The Twelfth International Conference on Learning Representations*, 2024.

[200] X. Zhang, R. R. Chowdhury, R. K. Gupta, and J. Shang, "Large Language Models for Time Series: A Survey," vol. 9, pp. 8335–8343, Aug. 2024. ISSN: 1045-0823.

[201] S. Abdullahi, K. Usman Danyaro, A. Zakari, I. Abdul Aziz, N. Amila Wan Abdullah Zawawi, and S. Adamu, "Time-Series Large Language Models: A Systematic Review of State-of-the-Art," *IEEE Access*, vol. 13, pp. 30235–30261, 2025.

[202] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, (Red Hook, NY, USA), pp. 19622–19635, Curran Associates Inc., Sept. 2023.

[203] H. Xue and F. D. Salim, "Promptcast: A new prompt-based learning paradigm for time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6851–6864, 2023.

[204] X. Chen, E. Carson, and C. Kang, "Llm-abba: Understanding time series via symbolic approximation," *IEEE Transactions on Signal Processing*, pp. 1–14, 2026.

[205] X. Liu, F. Zhou, H. Xiao, Z. Li, S. Liu, and L. Qian, "A survey on large language models for medical time series," *Expert Systems with Applications*, p. 131364, 2026.

[206] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time series forecasting by reprogramming large language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[207] C. Sun, H. Li, Y. Li, and S. Hong, "Test: Text prototype aligned embedding to activate llm's ability for time series," in *International Conference on Learning Representations*, vol. 2024, pp. 37854–37881, 2024.

[208] Z. Pan, Y. Jiang, S. Garg, A. Schneider, Y. Nevmyvaka, and D. Song, "S2ip-llm: semantic space informed prompt learning with llm for time series forecasting," in *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, JMLR.org, 2024.

[209] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, "One fits all: power general time series analysis by pretrained lm," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, (Red Hook, NY, USA), Curran Associates Inc., 2023.

[210] Z. Leng, H. Kwon, and T. Plötz, "Generating virtual on-body accelerometer data from virtual textual descriptions for human activity recognition," in *Proceedings of the 2023 ACM International Symposium on Wearable Computers*, pp. 39–43, 2023.

[211] Y. Zhang, Z. Dong, and W. Xu, "Integrative stock price trend prediction via hierarchical llm text processing and patch-based transformer with co-attention," *Expert Systems with Applications*, p. 130441, 2025.

[212] M. Tan, M. A. Merrill, V. Gupta, T. Althoff, and T. Hartvigsen, "Are language models actually useful for time series forecasting?," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, (Red Hook, NY, USA), Curran Associates Inc., 2024.

[213] L. N. Zheng, C. Dong, W. E. Zhang, L. Yue, M. Xu, O. Maennel, and W. Chen, "Understanding why large language models can be ineffective in time series analysis: The impact of modality alignment," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, (New York, NY, USA), p. 4026–4037, Association for Computing Machinery, 2025.

[214] F. Bellos, N. H. Nguyen, and J. J. Corso, "VITRO: Vocabulary Inversion for Time-series Representation Optimization," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, Apr. 2025. ISSN: 2379-190X.

[215] Z. Li, S. Deldari, L. Chen, H. Xue, and F. D. Salim, "SensorLLM: Aligning large language models with motion sensors for human activity recognition," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, (Suzhou, China), pp. 354–379, Association for Computational Linguistics, Nov. 2025.

[216] I. Hettiarachchi, "Unified Temporal Tokenization: A Hybrid Semantic and Numeric Mapping for Time-Aware Large Language Models," *Frontiers in Computer Science and Artificial Intelligence*, vol. 4, pp. 25–42, Dec. 2025.

[217] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[218] X. Long, B. Yin, and R. M. Aarts, "Single-accelerometer-based daily physical activity classification," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6107–6110, IEEE, 2009.

[219] M. Heydarian and T. E. Doyle, "rWISDM: Repaired WISDM, a Public Dataset for Human Activity Recognition," May 2023. arXiv:2305.10222 [eess].

[220] J. G. Proakis and D. G. Manolakis, *Digital signal processing*. Upper Saddle River, N.J: Pearson Prentice Hall, 4th ed ed., 2007. OCLC: ocm62804704.

[221] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*. John Wiley & Sons, Apr. 2012. Google-Books-ID: 0yR4KUL4VDkC.

[222] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, pp. 74–82, Nov. 2011.

[223] Y. Sun, X. Wang, G. Cao, and S. Mao, "FDALLM: Traffic Data Prediction with Functional Data Analysis and Large Language Models," in *ICC 2025 - IEEE International Conference on Communications*, pp. 1169–1174, June 2025. ISSN: 1938-1883.

[224] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7B," Oct. 2023. arXiv:2310.06825 [cs].

[225] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: efficient finetuning of quantized llms," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, (Red Hook, NY, USA), Curran Associates Inc., 2023.

[226] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models.," *Iclr*, vol. 1, no. 2, p. 3, 2022.

[227] L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, S. Huang, K. Rasul, and Q. Gallouédec, "TRL: Transformers Reinforcement Learning," 2020.

[228] G. Weiss, "WISDM Smartphone and Smartwatch Activity and Biometrics Dataset," 2019.

[229] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Protecting sensory data against sensitive inferences," in *Proceedings of the 1st Workshop on Privacy by Design in Distributed Systems*, pp. 1–6, 2018.

[230] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*. USA: Academic Press, Inc., 4th ed., May 2008.

[231] L. Pappa, P. Karvelis, G. Georgoulas, and C. Stylios, "Sax-based gnn embeddings for time series," in *2025 25th International Conference on Digital Signal Processing (DSP)*, pp. 1–5, IEEE, 2025.

[232] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[233] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[234] H. Chen and H. Eldardiry, "Graph time-series modeling in deep learning: A survey," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 5, pp. 1–35, 2024.

[235] M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King, and S. Pan, "A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, p. 10466–10485, Dec. 2024.

[236] F. Corradini, F. Gerosa, M. Gori, C. Lucheroni, M. Piangerelli, and M. Zannotti, "A systematic literature review of spatio-temporal graph neural network models for time series forecasting and classification," *Neural Networks*, p. 108269, 2025.

[237] G. Dong, M. Tang, Z. Wang, J. Gao, S. Guo, L. Cai, R. Gutierrez, B. Campbel, L. E. Barnes, and M. Boukhechba, "Graph neural networks in iot: A survey," *ACM Transactions on Sensor Networks*, vol. 19, no. 2, pp. 1–50, 2023.

[238] F. Liang, C. Qian, W. Yu, D. Griffith, and N. Golmie, "Survey of graph neural networks and applications," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 9261537, 2022.

[239] K. Mishra, S. Basu, and U. Maulik, "Graft: A graph based time series data mining framework," *Engineering Applications of Artificial Intelligence*, vol. 110, p. 104695, 2022.

[240] K. Mishra, S. Basu, and U. Maulik, "Network science for time series clustering and its applications," in *2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pp. 439–441, IEEE, 2024.

[241] D. Bünger, M. Gondos, L. Peroche, and M. Stoll, "An empirical study of graph-based approaches for semi-supervised time series classification," *Frontiers in Applied Mathematics and Statistics*, vol. 7, p. 784855, 2022.

[242] D. Zha, K.-H. Lai, K. Zhou, and X. Hu, "Towards similarity-aware time-series classification," in *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pp. 199–207, SIAM, 2022.

[243] X. Ma, M. Yu, H. Huang, R. Hou, M. Dong, K. Ota, and D. Zeng, "A time series classification method combining graph embedding and the bag-of-patterns algorithm: X. ma et al.," *Applied Intelligence*, vol. 53, no. 22, pp. 26297–26312, 2023.

[244] Z. L. Li, G. W. Zhang, J. Yu, and L. Y. Xu, "Dynamic graph structure learning for multivariate time series forecasting," *Pattern Recognition*, vol. 138, p. 109423, 2023.

[245] J. Ye, Z. Liu, B. Du, L. Sun, W. Li, Y. Fu, and H. Xiong, "Learning the evolutionary and multi-scale graph structure for multivariate time series forecasting," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2296–2306, 2022.

[246] H. Yu, T. Li, W. Yu, J. Li, Y. Huang, L. Wang, and A. Liu, "Regularized graph structure learning with semantic knowledge for multi-variates time-series forecasting," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22* (L. D. Raedt, ed.), pp. 2362–2368, International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[247] J. N. Adams, G. Park, and W. M. van der Aalst, "Preserving complex object-centric graph structures to improve machine learning tasks in process mining," *Engineering Applications of Artificial Intelligence*, vol. 125, p. 106764, 2023.

[248] V. F. Silva, M. E. Silva, P. Ribeiro, and F. Silva, "Time series analysis via network science: Concepts and algorithms," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 3, p. e1404, 2021.

[249] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[250] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The ucr time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, 2019.