Time Series Representation Techniques: A Survey

YUNCONG YU, IAV GmbH, Germany and Utrecht University, the Netherlands
TIM BECKER, IAV GmbH, Germany
WOLFGANG AIGNER, St. Pölten University of Applied Sciences, Austria
ALEXANDRU C. TELEA, Utrecht University, the Netherlands
MICHAEL BEHRISCH, Utrecht University, the Netherlands

	Stochastic Process $(1-\sum_{i=1}^{p}\phi_i\mathcal{B}^i)Y_t = (1+\sum_{i=0}^{q}\theta_i\mathcal{B}^i)\varepsilon_t$ (ARMA)	Integral Transform $\mathcal{F}(k)=\sum_{t=0}^{n-1} y_t e^{-j2\pi kt/n}$ (DFT)	Piecewise Represent. $\bar{y}_i = \frac{1}{n_w}\sum_{j=(i-1)n_w}^{in_w} y_j$ (PAA)	ML Model $\boldsymbol{z}=\sigma(XW+\boldsymbol{b})$ (A dense layer in ANN)	Dimens. Reduction $M=U\Sigma V^*$ (SVD)	Miscellan-eous $y(t)=g(t)+s(t)+h(t)+\epsilon_t$ (Prophet)
Examples	ARIMA, (G)ARCH, HMM	DFT, STFT, WT, HHT	(A)PAA, PLR, SAX	Ran. Forest, SVM, ANN	PCA, SOM, t-SNE	Prophet, PIP
Perspective	A time step ≙ a random variable	(Time-)frequency domain	Time domain	(feature space)	Latent embedding space	Diverse, e.g., visual curve shape fitting (PIP)
Typical Func./Task which methods solve	Forecasting	Analysis, denoising, compression	Diverse, e.g. compression	(Directly) classification and regression	Compression, clustering, summarization	Diverse, e.g., forecasting
Typical Discipline where methods originate	Statistics	Integral transform	Data science / data mining	Data science / machine learning	Data science	Diverse, e.g., data science
Typical App. Domain where methods serve	Econometrics/economics/ finance/business	Signal processing, control engineering	Agnostic	Agnostic	Agnostic	Diverse, e.g., business (Prophet)
Advantages	Explainability and control	Explainability	Efficiency and universal applicability	Universal applicability	Universal applicability	Diverse, e.g., intuitive constructs (Prophet)
Disadvantages	Strong assumptions	Strong assumptions	Limited space transform and knowledge extraction	Limited explainability	Little help with method selection and parameteri.	Diverse
Major Consideration	Stationarity	Physical dynamics	No major noteworthy considerations	Problem formulation / task assigned to the model	Form of data manifold	Diverse

Fig. 1. ***Taxonomy of Time Series Representation Techniques.*** *We classify time series representation techniques for multiple analysis tasks, studied in diverse disciplines, and applied in various domains into six categories based on their essential technical affinities.*

Existing State-of-the-Art Reports (STARs) on time series representations/transformations/models are confined to limited downstream tasks, academic disciplines, and application domains, such as deep learning models or forecasting models in econometrics. They typically collect and group Time Series Representation Techniques (TSRTs) loosely, by describing the properties of each TSRT individually while analyzing group properties up to a limited extent. We propose a taxonomy of TSRTs which is interdisciplinary, domain-agnostic, and covers more techniques than existing STARs. We classify TSRTs into six categories based on their technical affinities. The TSRTs in each category are presented in such a logical order that the latter ones fill the gaps of the former ones. Also, we extract and survey nine common data assumptions and analyze the factors affecting the effective choice of TSRTs. Our taxonomy helps researchers and practitioners enter the field Time Series Analysis (TSA) by providing an overview of TSRTs with clear technical

Authors' addresses: Yuncong Yu, IAV GmbH, Gifhorn, Lower Saxony, Germany and Utrecht University, Utrecht, the Netherlands, yuncong.yu@iav.de; Tim Becker, IAV GmbH, IAV GmbH, Germany, tim.becker@iav.de; Wolfgang Aigner, St. Pölten University of Applied Sciences, St. Pölten, Austria, wolfgang.aigner@fhstp.ac.at; Alexandru C. Telea, Utrecht University, Utrecht, the Netherlands, a.c.telea@uu.nl; Michael Behrisch, Utrecht University, Utrecht, the Netherlands, m.behrisch@uu.nl.

Manuscript submitted to ACM

lineages, rich examples, intuitive explanations, and practical tips. Experienced readers may benefit from the comprehensive method collection during a quick search of method candidates and from the suggested research directions.

CCS Concepts: • **Mathematics of computing** → **Time series analysis**; • **Information systems** → *Data mining*.

Additional Key Words and Phrases: Time Series Representations, Time Series Models, Time Series Transformations

ACM Reference Format:

Yuncong Yu, Tim Becker, Wolfgang Aigner, Alexandru C. Telea, and Michael Behrisch. 2018. Time Series Representation Techniques: A Survey. *ACM Comput. Surv.* 37, 4, Article 111 (August 2018), 48 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

A time series can be defined as a sequence of real-valued observations recorded chronologically [37, 56, 80, 100, 176], TSA or time series data mining addresses the tasks of information extraction and knowledge discovery from time series. During analysis, time series are represented in different ways in distinct spaces, under diverse assumptions, and targeting various tasks, by what is known as Time Series Representation Technique (TSRT). TSRTs play key role in TSA. For instance, without AutoRegressive Integrated Moving Average (ARIMA) and its variants, econometrics may not exist; without Fast Fourier transform (FFT) and its followers, signal processing and control engineering would not have achieved considerable progress.

A vast array of TSRTs exists, prompting several key inquiries: Are they *interconnected*? What common *properties* do they possess, and how do they differ? Which *factors* should or could analysts consider when selecting these TSRTs for applications? Our work tries to answer these questions.

Answering such questions is necessary because analysts carrying out TSA may get lost in the huge amount of work in this area. Indeed, TSRTs are separately studied and applied in many disciplines and domains, e.g., stochastic process models (statistics and econometrics) and integral transforms (signal processing). Such fields view time series from distinctive perspectives, make different data assumptions, and serve various downstream tasks, leading to multiple lineages of TSRTs. Separately, we found that most STARs on TSRTs [69, 220, 239, 259] focus on one or a few domains. They also emphasize the analysis of state-of-the-art methods and focus less on their development and inter-relationships. Lastly, surveys explain the most relevant and unique, thus diverse and inconsistent, properties for individual TSRTs, making method and category comparison challenging. For all the above reasons, we believe that an *interdisciplinary* survey on TSRTs is of added value for helping practitioners and researchers with the application, transfer, and innovation of TSRTs.

Our taxonomy classifies TSRTs into six *categories*:

(1) stochastic process, e.g., ARIMA, ARCH, and HMM;
(2) integral transform, e.g., FFT, STFT, and WT;
(3) piecewise representation, e.g., PAA, PLR, and SAX;
(4) machine learning model, e.g., SVM, random forest, and ANN;
(5) dimensionality reduction, e.g., PCA, SOM, and t-SNE; and
(6) miscellaneous, e.g., Prophet and PIP.

We also propose five *factors* to consider when choosing a TSRTs for concrete applications:

(1) physical dynamics, e.g, some TSRTs decompose a time series into different components to model different system dynamics;
(2) data assumptions, e.g, some TSRTs presume normality and stationarity;

(3) task type, e.g., some TSRTs are designed for forecasting, while others geared to anomaly detection;
(4) technique transfer, e.g., some TSRTs represent a numerical-valued time series with a symbolic string, enabling text processing techniques; and
(5) computational resources, e.g., some TSRTs consumes excessive CPU or RAM when handling high-dimensional time series.

Our proposed taxonomy makes three distinctive contributions: 1) we *unite* groups of techniques developed in different disciplines to foster method transfer and combination; 2) we list *factors* to consider when applying TSRTs, helping analysts with effective TSRTs choice; 3) we list the most *representative* TSRTs with eight common properties and properties unique to each method explicitly, as well as typical use cases, helping again practitioners with method choice.

Our STAR is organized as follows. We begin by comparing our STAR with other related state-of-the-art STARs (Section 2). Section 3 adds important concepts and needed background knowledge. Section 4 outlines our literature selection method. Section 5 presents our taxonomy formally. Section 6 introduces the factors to consider when choosing a TSRT. We next provide a comprehensive list of the most representative TSRTs with their data assumptions and typical use cases (Section 7). Finally, Section 8 discusses the limitations of our work and suggests future research directions. Section 9 concludes our survey.

2 COMPARISON WITH OTHER STARS

Table 1 compares representative STARs with ours regarding the six categories we propose. We further elaborate on these differences.

First, our taxonomy is technically diverse. Unlike, e.g., [69] (focus on stochastic processes), [259] (focus on time-domain representations from data mining), or [147, 239] (focus on deep learning), we cover a wider set of TSRTs in different technical lineages.

Second, our taxonomy is interdisciplinary and domain-agnostic. Unlike, e.g., [104] (focus on econometrics), [36] (focus on econometrics as well as system and control engineering), or [229] (focus on statistics), our domain-agnostic work surveys techniques from different academic disciplines and application domains including statistics in econometrics/economics/finance/business, signal processing in meteorology/geology/biology/engineering, and general data mining / machine learning.

Third, our taxonomy is versatile and multi-functional. Unlike, e.g., [74] (focus on retrieval), [220] (focus on forecasting), or [53] (focus on compression), we reviewed methods for multiple tasks, see further Section 3.2.

Besides a broader scope, our exposition of the TSRTs in each category outlines how gaps in earlier methods are covered by succeeding ones, highlighting the lineage of technical advancements. Furthermore, we extracted eight common properties of TSRTs along with unique ones, underscoring a more direct comparison between the methods. Finally, we propose five factors to consider during the selection of TSRTs for practical applications.

3 THEORETICAL BACKGROUND

3.1 Time Series

A time series is a sequence of real-valued observations recorded chronologically [37, 91, 176]. Many real-world data can be modeled as time series – sales development [14, 194], energy consumption transitions [66, 231], stock price fluctuations [76, 91], Internet of Things (IoT) measurements [60, 146], audio recordings [81, 132], concentration changes

STAR	Stochastic Process	Integral Transform	Piecewise Represent.	ML Model	Dimension. Reduction	Miscell.
Keogh [136]		✓	✓		✓	
Ding et al. [74]		✓	✓			
Längkvist et al. [147]				✓		
Box et al. [37]	✓					
Chatfield [48]	✓	✓				
Wilson [259]			✓			✓[1][2]
Salles et al. [220]	✓	✓				
Hamilton [104]	✓	✓				
Deistler et al. [69]	✓	✓				
Mertins [177]		✓				
Triat et al. [239]				✓		
Our STAR	✓	✓	✓	✓	✓	✓

[1] Perceptually Important Point (PIP).
[2] There are diverse primitive transformations like logarithmic transform, Box-Cox transform, and simple differencing, which we exclude from TSRTs, explained in Section 3.3.

Table 1. ***Representative STARs on TSRTs.*** *While most works focus on one or several technical lineages, our work has a broader scope.*

in a chemical process [19, 57], genetic sequences [20, 148], and medical charts like Electroencephalogram (EEG) [189, 277] and Electrocardiogram (ECG) [170, 234].

Formally, we define a time series

$$\left[\boldsymbol{y}_t \in \mathbb{R}^m\right]_{0\leq t<n} = \left[y_{t,v}\right]_{\substack{0\leq t<n \\ 0\leq v<m}} \tag{1}$$

with n time points and m recorded variables as a sequence of n real-valued, m dimensional, vectors. Here, we adopt the square bracket notation for a matrix. The time axis is indexed with $t = 0, \ldots, n-1$ (equidistant sampling assumed), and the variables are indexed with $v = 0, \ldots, m-1$. Please refer to Appendix A for a list of symbols.

Throughout this STAR, we use uppercase letters for matrices and random variables, lowercase letters for scalars, lower- or uppercase letters in boldface for vectors, uppercase letters in special fonts for functions or operations, Two exceptions are the error term ε_t and the conditional heteroscedasticity σ_t^2 in stochastic process models, which are random variables but in lowercase due to convention. We strive to avoid overloading one symbol with multiple meanings. In rare cases, the same symbol may have different meanings, e.g., the symbol j is used as the imaginary unit and an index in different formulas. Nonetheless, the usage should be well distinguishable.

Figure 2 illustrates an example time series (unfaithful hand-drawn based on measurements) from an engine control unit in a car.

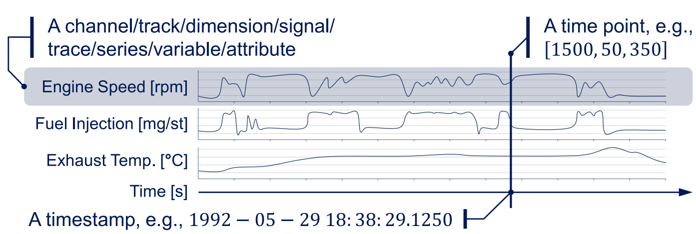

Fig. 2. ***An Example Time Series.*** *A time series consists of sequentially ordered real-valued (potentially multidimensional) time points. The values of the same dimension ordered in ascending order of time in all time points form a channel.*

Along the time axis lies the essential property of time series, i.e., the time points are sequentially ordered [36, 239], and the order carries information. Some sequential data (like genomic sequences [148]) and even some two-dimensional shapes [138, 268] are sequentially ordered, similar to time series and can be addressed with methods in TSA.

If $m > 1$ and the time series records multiple variables, it is called a multivariate, multidimensional [248], high-dimensional [21], multichannel [265] time series, or multiple time series [167]. We adopt the term "multivariate" that is established in statistics/econometrics [37, 104, 229, 255] and data mining / data science [53, 80, 239]. A variable in a time series is also referred to by many names, like "channel" [105], "track" [273], "dimension" [80], "signal" [280], "trace" [199], "series" [36], "variable" [167], and "attribute" [127]. We did not find a prevailing term and will use "channel" in this STAR. Unlike the time points, the channels are unordered. They may be homogeneous, like the closing prices of multiple stocks in consecutive days or signals in an EEG measurement. A time series may also consist of heterogeneous channels, like engine speed, fuel injection, and exhaust temperature from various sensors monitoring the same process in a powertrain, illustrated in Figure 2, which may require scaling or normalization when analyzed jointly.

3.2 Time Series Analysis

The term "Time Series Analysis (TSA)" [37, 104, 229], also known as "time series data mining" [56, 80, 84], refers to the process of information extraction and knowledge discovery from time series data [84]. Conceptually, a time series can be conceived as a discrete-time measurement of the manifestation of certain aspects of a physical process. The ultimate goal of TSA is to ascertain the rules governing the physical process.

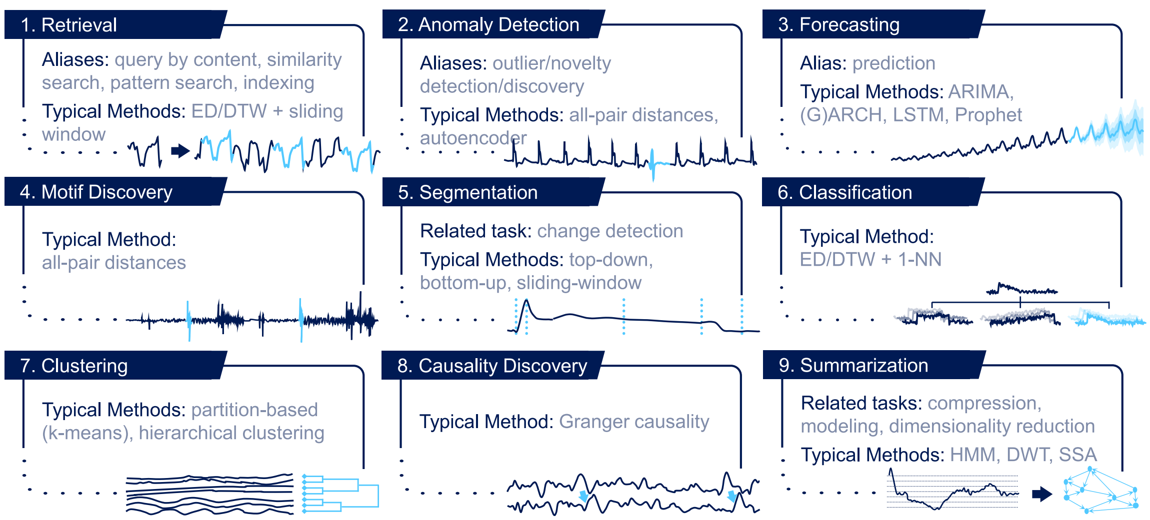

Fig. 3. ***Tasks in TSA.*** *TSA studies methods for information extraction and knowledge discovery from time series data. We found nine typical tasks [56, 80, 84, 136]. Note: the methods mentioned in this figure are not necessarily TSRTs.*

Many established problems exist in the research field of TSA. Figure 3 shows nine typical problems [56, 80, 84, 136]. These are:

(1) Locating patterns similar to a given one, e.g., to trace an event of interest (retrieval, query by content, similarity search, pattern search, indexing) [12].
(2) Detect abnormal events that may relate to errors or novelties (anomaly/outlier/novelty detection) [27, 150, 224];
(3) Extrapolating future development of the data (forecasting/prediction) [23, 52, 174];
(4) Unearth previously unknown recurrent behavior, e.g., for association rule learning (motif discovery) [5, 205, 238];
(5) Splitting the data into consecutive pieces that are internally homogenous and heterogeneous with each other, e.g., to analyze phases in a process individually (segmentation / change detection) [96, 165];

(6) Categorizing time series and thereby the items they measure to known classes (classification) [178, 179];
(7) Dividing multiple time series into previously unknown groups, with characteristics shared by group members but different across groups, to reveal common behavior patterns (clustering) [8, 114, 158];
(8) Finding causal relationships between two channels (causality discovery) [15, 181], which is less mentioned in many surveys on tasks in TSA [80, 84, 136]; and
(9) Summarizing (usually also visualize) the key features of large time series (summarization) [136, 190, 233].

Figure 3 also lists representative methods addressing individual problems, which we find potentially helpful for readers. But we omit their detailed explanation, as it would go beyond the scope of this STAR.

As Figure 4 shows, TSA, viewed in a broader scope, integrates knowledge and techniques from many fields where sequential data analysis is of concern. These fields interpret the data from different perspectives, solve distinctive issues, and propose different techniques.

3.3 Time Series Representation Techniques

We define a TSRT as a method that converts a time series into another form while retaining information on its salient dynamics. The transformed form usually exposes the patterns of interest more prominently or simplifies further data processing. Based on the retained information, many TSRTs support exact, approximate, or even trained [82] inverse transformation.

The transformed time series may stay in the time domain, as with many time-domain piecewise representations [45, 135]. It is also common to see a time series transformed into another domain, like the frequency domain, as with Discrete Fourier Transform (DFT), or an embedding space, as with a Dimensionality Reduction (DR) technique. A TSRT can describe system dynamics explicitly, such as ARIMA and Hidden Markov Model (HMM). It can also generate synthetic data with the same properties as given examples, as with generative machine learning models like Generative Adversarial Network (GAN). Furthermore, a TSRT can prescribe a set of operations, as with integral transforms, piecewise representations, and some DR techniques, or model the time series, as with stochastic processes and machine learning models. It can be deterministic as with a linear regression model, or stochastic as with a stochastic process model. It can make assumptions about the data like following a normal distribution, as with Symbolic Aggregate approXimation (SAX) [161], and stationarity as with AutoRegressive Moving-Average (ARMA) [37], or agnostic about the data properties, as with many machine learning models. It can be geared towards certain tasks explained in Section 3.2 like ARCH only for predicting variance/heteroscedasticity/volatility [79], or be general-purpose like many DR techniques. It may serve in modeling complex system dynamics, or in data denoising and compression. It may incur consequential overhead like model training, or be very fast like many piecewise representations.

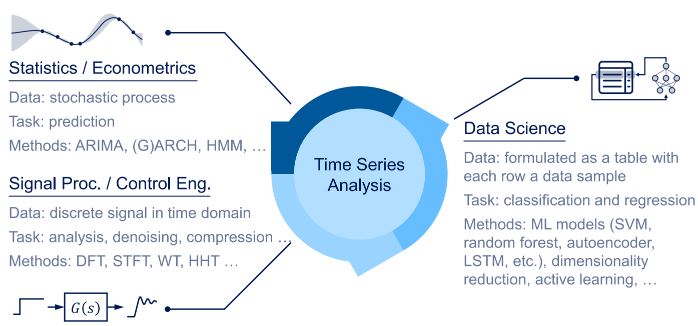

Fig. 4. ***Intersection With Other Disciplines and Domains.*** *TSA intersects other disciplines and domains with different interpretations of the data, tasks, and methods.*

However, we exclude indexing techniques like R-Tree and its variants, also commonly used in TSA, especially in retrieval [2, 85]. Their design purposefully targets a universal and drastic data reduction. We are skeptical of qualifying them as TSRTs because they may not preserve sufficient information on data characteristics. Similar to our consideration, Esling et al. treat TSRT and indexing as two separate topics [80]. Together with the similarity measure, they consider TSRT one of the three major issues in time series data mining [80]. For instance, Fu does not differentiate representation and indexing [56], and Keogh explicitly includes trees as a TSRT [136].

We also omit transformations like Box-Cox transform [34], differencing, etc., which Salles et al. feature in their STAR [220]. Their primary purpose is to make a time series stationary, enabling other TSRTs, especially stochastic process models, which require stationarity. They are relatively simple operations that do not need much explanation. They are useful and extract some features well, but may not preserve the overall general dynamics in the data.

4 LITERATURE SELECTION

4.1 Inclusion Criteria

We include works that convert time series into a representation and solve the downstream task by working on this representation. Apart from this, we did not confine:

Data format, academic discipline, and application domain: Time series data can be *e.g.* stock prices in finance, precipitation in meteorology, genetic sequences in biology, and audio signals in engineering.
Downstream task: We consider all tasks in Section 3.2.
Publication year: We include old works, especially those introducing classic methods; we also check recent publications, especially to get an overview of the most frequent method applications.
Venue: ACM SIGKDD, IEEE ICDM, Springer DMKD, SIAM SDM, ECML PKDD, among others.

We used Scopus and the Web of Science (WoS) core collection for our literature search as these sources provide the world's largest interdisciplinary, domain-agnostic, and cross-venue scientific citation indexes which one can analyze externally.

4.2 Queries

Our literature selection began with review articles (including tutorial papers) and monographs (including book chapters) on TSRTs, including works on general TSA. The technique papers for individual methods proliferate after snowballing, see Figure 5.

We started with searching of survey papers on TSRTs to avoid rediscovering existing ontologies. As explained in Section 2, our scope is much broader than other surveys on TSRTs. Therefore, existing surveys on a high level help as the initial step.

We first searched for survey papers whose title contains "time series" and "representation" (or variants and synonyms). Specifically, the query for Scopus reads "TITLE (time-series AND representation OR transform* OR model) AND (LIMIT-TO (DOCTYPE , "re"))". In WoS, we used the query "TI=(time-series AND (representation OR transform* OR model))" and refined the document type to "Review Article" (WoS does not support specifying document types in the query string directly). In both queries, "time-series" subsumes "time series"; "transform*" also captures "transformation". Scopus and WoS take care of lemmatization like plural forms. Note that the boolean operator OR precedes AND in Scopus but the other way around in WoS. Scopus retrieved 95 documents and WoS 76.

There are four potential problems with these first queries.

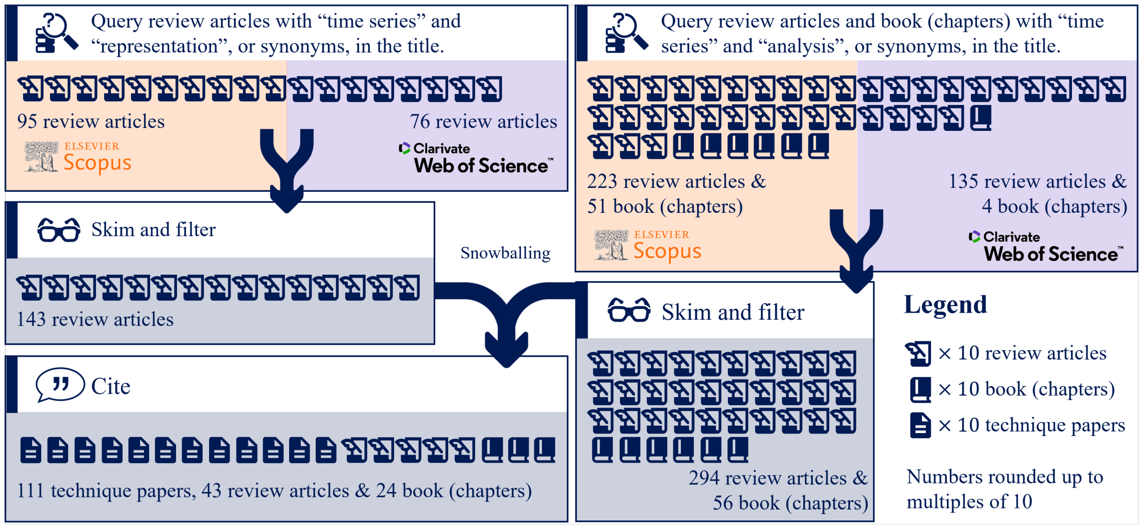

Fig. 5. ***Queries and Results During Literature Selection.*** *Our literature selection began with review articles on TSRTs and works on general TSA. Through snowballing, we cover the essential technique papers for individual methods.*

First, we likely overly rely on knowledge introduced by existing reviews instead of proposing our own. Yet, due to our much broader scope, no existing taxonomy exists on this level. In fact, it would be even more efficient and thus preferable to reuse existing reviews or taxonomies as sub-systems in our larger one.

Second, review articles may not cover up-to-date publications and recent advancements. However, it is not a big problem in our case, because we are not focused on recent advancements but on established technology and knowledge systems. We will search for recent publications on individual methods, especially for discussing their typical applications.

Third, our queries only examined titles but omitted keywords and abstracts. However, if we had loosened the query to also include abstracts and keywords in Scopus alone with "(TITLE-ABS-KEY (time AND series) AND TITLE-ABS-KEY (representation OR transform* OR model)) AND (LIMIT-TO (DOCTYPE , "re"))", the resulted 5428 documents would become unmanageable for us (also consider snowballing).

Fourth, our search misses survey papers and monographs on general TSA, which also present TSRTs. Therefore, we reviewed also publications for general TSA with the query "TITLE (time-series AND analysis OR mining) AND (LIMIT-TO (DOCTYPE , "re") OR LIMIT-TO (DOCTYPE , "bk"))" in Scopus and "TI=(time-series AND (analysis OR mining))" (refined to review articles and book chapters) in WoS. They returned 274 and 139 entries, respectively.

After tidying and merging, we manually inspected the documents and selected those describing time series representations/transformations/models. During the review, we used snowballing to get even more publications, including those introducing a single method frequently seen in the review articles.

Finally, we also paid attention to technique papers, such as [45, 148], which benchmarked various representative TSRTs while presenting their own.

5 TAXONOMY OF TIME SERIES REPRESENTATION TECHNIQUES

We propose a new taxonomy for TSRTs, see Figure 1. Our categories include **stochastic process**, **integral transform**, **pieceswise representation**, **machine learning model**, **dimensionality reduction**, and **miscellaneous**. They are created according to the fundamental dominant *technique* and described further starting with Section 5.1. Two reasons exist for this choice of classification criteria.

First, consider alternative taxonomies like ''functions (e.g., classification model, regression model, generation model), tasks (e.g., forecast model, data compression technique, denoising technique), and certain properties (e.g., deterministic/stochastic model, time-domain / frequency-domain representation)''. A TSRT may have several functions, serve various tasks, and possess multiple properties. In contrast, the fundamental *technique* for a TSRT is relatively unique and stable over time. Hence, there is less ambiguity creating categories for TSRTs. Second, the resulting categories are mostly already treated as established and self-contained disciplines studied as individual subjects. There is no need to define new concepts or explain them extensively. Subsequent divisions can also inherit the existing taxonomy in each discipline.

While technique-oriented, we present the motivations and concepts behind methods. Furthermore, we emphasize *relationships* between methods in each category so as to create a system of connected rather than disjoint knowledge nodes.

As there are overwhelmingly many new but less proven developments, we skip recent method variations of core, established methods. For instance, there are hundreds of extensions of the AutoRegressive Conditional Heteroskedasticity (ARCH) model alone [29], and none seems to dominate. Nor do we claim to be exhaustive with our TSRTs, as our goal is to establish a taxonomy with different technical lineages. Overall, we try to mention as many representative TSRTs as possible.

5.1 Stochastic Process

Statistics, especially econometrics, often view a time series as a realization of a stochastic process. A stochastic (random) process is a sequence of random variables, whose index is usually interpreted as time [43, 102]. Most stochastic process representations are forecasting models.

Established by the work of Box et al. [35, 37], the most influential models in this discipline are the family of AutoRegressive Integrated Moving Average (ARIMA) models, namely, AutoRegressive (AR), Moving-Average (MA), ARMA, ARIMA, Seasonal AutoRegressive Integrated Moving Average (SARIMA), Vector AutoRegressive (VAR), and AutoRegressive Integrated Moving Average with eXogenous inputs (ARIMAX), discussed next.

AR Model. The essential property of time series, namely, the dependency between time points (Section 3.1), is often reflected in the predictive power of preceding time points to explain the next time point. An AR (p) model estimates the value Y_t of a univariate time series at timestamp t as a linear combination of p lagged values with

$$Y_t = \sum_{i=1}^{p} \phi_i Y_{t-i} + \varepsilon_t \tag{2}$$

where Y_t is seen as a random variable (therefore with uppercase letter Y); $\phi_1, \ldots, \phi_p$ are constant coefficients; and $\varepsilon_0, \varepsilon_1, \ldots$, called errors or innovations, are white noise, usually assumed to be independent and identically distributed (i.i.d) random variables following a zero-mean normal distribution [37]. This approach resembles linear regression, hence the word "regression" in AR. AR uses an observed variable to predict the same variable, hence the prefix "auto-". Some works feature a constant bias c in Equation 2. For simplicity, we assume that Y_t is centered (i.e., c has already been subtracted, resulting in Y_t).

MA Model. Instead of historical values, this model relies on current and past errors $\varepsilon_0, \ldots, \varepsilon_t$ to estimate Y_t as

$$Y_t = \sum_{i=0}^{q} \theta_i \varepsilon_{t-i} \tag{3}$$

where ε_t has the same meaning as in the AR model; and θ_0, ..., θ_q are constant coefficients with $\theta_0 = 1$. The MA model captures recent short-term effects, as the information carried in $\varepsilon_{t-(q+1)}$ disappears after $q+1$ time points in Equation 3. In contrast, the AR model keeps track of long-term effects, as the first observation Y_0 still exerts some influence on Y_t, potentially after decaying over time through recursion according to Equation 2,

ARMA Model. The combination of an AR model describing long-term system dynamics and an MA model incorporating short-term shocks yields the ARMA (p, q) model [37, 182] defined as

$$Y_t = \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{i=0}^{q} \theta_i \varepsilon_{t-i} \tag{4}$$

Using the lag operator $\mathcal{B}$ defined as $\mathcal{B}^i Y_t = Y_{t-i}$, Equation 4 can be rewritten as

$$\left(1 - \sum_{i=1}^{p} \phi_i \mathcal{B}^i\right) Y_t = \left(\sum_{i=0}^{q} \theta_i \mathcal{B}^i\right) \varepsilon_t \tag{5}$$

ARIMA Model. The AR, MA, and ARMA models presume data stationarity (see Section 6.2). If nonstationarity exists, it must be removed by e.g., differencing. In a ARIMA (p, d, q) model, an ARMA (p, q) model is preceded by direct differencing to remove the trend via

$$\left(1 - \sum_{i=1}^{p} \phi_i \mathcal{B}^i\right) (1 - \mathcal{B})^d \, Y_t = \left(\sum_{i=0}^{q} \theta_i \mathcal{B}^i\right) \varepsilon_t \tag{6}$$

where d is the degree of differencing.

SARIMA Model. The SARIMA $(p, d, q) \left(\widetilde{p}, \widetilde{d}, \widetilde{q}\right)_{\widetilde{s}}$ model addresses seasonal nonstationarity by seasonal differencing as

$$\left(1 - \sum_{i=1}^{\widetilde{p}} \widetilde{\phi}_i \mathcal{B}^{i\widetilde{s}}\right)\left(1 - \sum_{i=1}^{p} \phi_i \mathcal{B}^p\right) Y_t = \left(\sum_{i=0}^{\widetilde{q}} \widetilde{\theta}_i \mathcal{B}^{i\widetilde{s}}\right)\left(\sum_{i=0}^{q} \theta_i \mathcal{B}^q\right) \varepsilon_t \tag{7}$$

where $\widetilde{p}$ is the seasonal AR order; $\widetilde{d}$ the seasonal differencing degree; $\widetilde{q}$ the seasonal MA order; $\widetilde{s}$ is the length of a season; $\widetilde{\phi}_1, \ldots, \widetilde{\phi}_{\widetilde{p}}$ are constant factors of the seasonal AR component; and $\widetilde{\theta}_0, \ldots, \widetilde{\theta}_{\widetilde{q}}$ are constant factors of the seasonal MA component with $\widetilde{\theta}_0 = 1$.

VAR Model. The above models describe *univariate* time series. In multivariate cases, one regards a previously scalar time point as a vector and the previously scalar coefficients as matrices, so that temporal dynamics and inter-channel relationships can be described simultaneously. The VAR (p) model extends the AR (p) model with

$$\boldsymbol{Y}_t = \sum_{i=1}^{p} \Phi_i \boldsymbol{Y}_{t-i} + \boldsymbol{\varepsilon}_t \tag{8}$$

where $\boldsymbol{Y}_t \in \mathbb{R}^m$ are the values of the m observed variables at time t in a multivariate time series; $\Phi_i \in \mathbb{R}^{m \times m}$, the counterpart of the scalar ϕ_i in Equation 2, is now a matrix; $\boldsymbol{\varepsilon_i} \in \mathbb{R}^m$, the counterpart of the scalar ε_t, is now a white noise vector following a zero-mean multivariate distribution; in case of non-zero-mean observations, Equation 8 needs an extra bias vector $\boldsymbol{c}$. More complex models like Vector AutoRegressive Moving-Average (VARMA) have much higher computational costs during parameter estimation than VAR [168], making the VAR model practically attractive in multivariate cases.

VARX Model. Sometimes, observations of additional variables $\left[\boldsymbol{X}_t \in \mathbb{R}^{m'}\right]_{0\leq t<n}$ (exogenous variables) provide information on the variables of interest $[\boldsymbol{Y}_t]_{0\leq t<n}$ (endogenous variables) that appear on both side of the system dynamic equation. The ARIMA models above can be extended with exogenous variables, yielding the ARIMAX model [24]. A Vector AutoRegressive eXogenous (VARX) model of order p and p', i.e., VARX (p, p'), is given by

$$\boldsymbol{Y}_t = \sum_{i=1}^{p} \Phi_i \boldsymbol{Y}_{t-i} + \sum_{i=0}^{p'} \Gamma_i \boldsymbol{X}_{t-i} + \boldsymbol{\varepsilon}_t \tag{9}$$

where $\Gamma_0, \ldots, \Gamma_{p'}$ are constant coefficient matrices of size $m \times m'$.

SSM. Control engineering often uses deteministic differential equations to describe continous signals. Because time series are usually sampled in discrete time with stochastic error, TSA is more interested in the following form of difference equations with stochastic terms

$$\boldsymbol{S}_t = A\boldsymbol{S}_{t-1} + B\boldsymbol{U}_t + \boldsymbol{\varepsilon}_t \tag{10}$$

$$\boldsymbol{Y}_t = C\boldsymbol{S}_t + D\boldsymbol{U}_t + \boldsymbol{\epsilon}_t \tag{11}$$

where $\boldsymbol{S}_t$ is a vector describing the system state and $\boldsymbol{S}_0$ is often assumed to follow a multivariate normal distribution; $\boldsymbol{U}$ is a vector describing the system inputs; $\boldsymbol{\varepsilon}_t$ and $\boldsymbol{\epsilon}_t$ are white noise, usually assumed to following zero-mean i.i.d multivariate normal distributions; and A, B, C, and D are constant coefficient matrices. This form is also called the linear Gaussian state space model or Dynamic Linear Model (DLM) [229]. Equation 10 describes the system dynamics and Equation 11 the relationship between the system state and observed variables. For instance, the VARX (p) model in Equation 9 can be reformulated as a State-Space Model (SSM) with

$$\boldsymbol{S}_t = \begin{bmatrix} \boldsymbol{Y}_t \\ \vdots \\ \boldsymbol{Y}_{t-p} \end{bmatrix} \quad \boldsymbol{U}_t = \begin{bmatrix} \boldsymbol{X}_t \\ \vdots \\ \boldsymbol{X}_{t-r} \end{bmatrix} \quad A = \begin{bmatrix} \Phi_1 & \Phi_2 & \ldots & \Phi_p \\ I & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & I & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & I \end{bmatrix}$$
$$B = \begin{bmatrix} \Gamma_0 & \ldots & \Gamma_r \end{bmatrix} \quad C = \begin{bmatrix} I & \mathbf{0} & \ldots & \mathbf{0} \end{bmatrix} \quad D = \mathbf{0} \tag{12}$$

where I is the identity matrix and the boldface $\mathbf{0}$ is the zero matrix. One motivation of this representation is the use of the Kalman filter, which can be used to estimate the most plausible $\boldsymbol{s}_t$, both for $t < n$ (potentially containing unobserved variables or missing data in $\boldsymbol{S}_t$) and $t \geq n$ (forecasting). State-space representations also lay the foundation for more complex stochastic process models like HMM.

ARCH Model. The previous models estimate the mean of time points and assume for each time point a constant variance, also called *homoscedasticity* in econometrics. To model variable variance, aka *heteroscedasticity* or volatility in econometrics, [79] introduced the ARCH (q) model

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 \tag{13}$$

where $\varepsilon_t \sim \mathcal{N}\left(0, \sigma_t^2\right)$ still means the error, but not time-invariant and not i.i.d anymore; σ_t^2 is the variance of ε_t; α_i are constant coefficients with $\alpha_0 > 0$ and $\alpha_i \geq 0 \; \forall \, 1 \leq i \leq q$, so that σ_t^2 is positive [37]. "Conditional heteroscedasticity" refers to the conditional time-variant variance $\sigma_t^2 = \operatorname{Var}\left(\varepsilon_t \mid \varepsilon_{t-1}, \ldots, \varepsilon_{t-q}\right)$.

GARCH Model. The ARCH model (Equation 13) parallels the AR model (Equation 2). This was was later extended to Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) akin to ARMA [28]. This extension reduces the otherwise large q needed by ARCH models [37]. A GARCH (p, q) model is defined as

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 \tag{14}$$

with $\alpha_0 > 0$, $\alpha_i \geq 0 \; \forall \, 1 \leq i \leq q$, and $\beta_i \geq 0 \; \forall 1 \leq i < p$, so that σ_t^2 is positive.

Nonlinear Models. We have exdposed so far only *linear* stochastic process models (linearity is defined in Section 6.2). Nonlinear stochastic process models are also rich in variety, e.g, bilinear models [11], Threshold AutoRegressive (TAR) models including the popular Self-Exciting Threshold AutoRegressive (SETAR) model [237], HMMs including the well-known Markov switching model [103], and various nonlinear derivations of the linear (G)ARCH model [29, 226]. Many nonlinear stochastic process models emerge from linear counterparts by making the previously constant coefficients stochastic and conditional on previous information [37].

During application of ARIMA faimily, the Box-Jenkin method systematically prescribes the procedure [37]: 1) model identification including the orders like p and q based on the data nonstationarity (trend or seasonality), Autocorrelation Function (ACF), and Partial Autocorrelation Function (PACF); 2) parameter estimation for ϕ_i, ϑ_i, etc. with likelihood estimation or Bayesian methods; 3) diagnostic checking to verify the convergence of the parameterized model.

Compared with linear stochastic process models, nonlinear ones are less well studied, especially *vs* the types of nonlinearity to tackle and the methodology for systematic model selection. In practice, this issue can be eased by machine learning models at the expense of explainability. It is potentially interesting to study the possible treatment of nonlinearity in stochastic process models. Are there dominant types of nonlinearities in individual domains? Can common types of nonlinearity be found and then removed through certain operations, like differencing for removing some types of nonstationarity? Finally, how can models be parameterized and validated? Admittedly, the best (nonlinear) models mimic the physical dynamics of the observed system, which may not be fully accessible. Still, the modeling may benefit from prior knowledge when possible.

5.2 Integral Transform

Signal processing and control engineering often treat a time series as signals in the time domain. Before further data processing, the signal often undergoes an integral transform. The integral transform is defined as an operation that maps a function from its original space to another image space via integration [68]. Mathematically, an integral transform $\mathcal{I}$ applied to a function $y\left(t\right)$ on an interval $\left[t_1, t_2\right]$ is defined as

$$\mathcal{I}\left(y, \boldsymbol{k}\right) = \int_{t_0}^{t_1} y\left(t\right) \mathcal{K}\left(t, \boldsymbol{k}\right) dt \tag{15}$$

where $\mathcal{K}$ is called the kernel of the transform and $\boldsymbol{k}$ are the parameters of the transform $\mathcal{I}$. In TSA, the function to map by the integral transform is the time series itself, i.e., $y\left(t\right) = y_t$, and $t_0 = 0, \; t_1 = n - 1$. Since t is in this case discrete, we are more interested in the form

$$\mathcal{I}\left(y, \boldsymbol{k}\right) = \sum_{t=0}^{n-1} y_t \mathcal{K}\left(t, \boldsymbol{k}\right) \tag{16}$$

Strictly speaking, this discrete form no longer performs (but can approximate) integration and is thus ineligible for the title integral transform. Other terms, such as discrete integral transform [16, 17] and discrete transform [86, 123], exist in the literature, though not well established and sometimes ambiguous. Consequently, we cling to the term integral transform. Unlike stochastic process representations that are models with trainable parameters, integral transforms are a set of operations with fixed rules. Integral transforms help especially with signal analysis, compression, and filtering.

DFT. The most foundational integral transforms in TSA are the Fourier Transform (FT) and its variants. FT uses harmonic waves as the integral kernel, i.e., $\mathcal{K}(t,\boldsymbol{k}) = e^{-j2\pi f t}$ in Equation 15, and converts the time series into its frequency spectrum, i.e., the parameter f has the physical meaning of frequency. FT is intended for continuous signals that have a time-invariant frequency spectrum and span infinitely to the past and to the future, i.e., $t_0 \to -\infty$, $t_1 \to +\infty$, and $t \in \mathbb{R}$. Its discrete counterpart is the Discrete-Time Fourier Transform (DTFT) with $\mathcal{K}(t,\boldsymbol{k}) = e^{-j2\pi k t}$ in Equation 16 (assuming sampling rate once per time unit). However, DTFT still requires the knowledge of the whole data before and after measurements, i.e., the summation limits are not 0 to $n-1$ as in Equation 16, but $-\infty$ to $+\infty$. By assuming repetition of the time-limited measurement, this problem is circumvented by DFT

$$\mathcal{F}(y,k) = \sum_{t=0}^{n-1} y_t e^{-j2\pi \frac{k}{n} t} \tag{17}$$

where j is the imaginary unit and $0 \le k < n$ [177]. In practice, DFT is usually executed efficiently as FFT via the Cooley-Tukey algorithm [61].

STFT. DFT still assumes a time-invariant frequency spectrum. In other words, DFT measures the presence of trigonometric components of various frequencies, while the temporal information of when these components occur is lost. Short-Time Fourier Transform (STFT) approaches this problem by conceptually conducting FT in a sliding window along the time axis to analyze time-variant frequency in the time-frequency domain. Specifically, it uses the kernel $\mathcal{K}(t,\boldsymbol{k}) = w(t-\tau)\, e^{-j2\pi f t}$, where the extra parameter τ shifts the window function $w(t)$ along the time axis. Since STFT focuses on temporally localized features, a limited-time version is obsolete. For TSA, it is more relevant to examine its discrete form

$$\mathcal{F}(y,f,\tau) = \sum_{t=\tau-n_w/2}^{\tau+n_w/2} y_t w(t-\tau)\, e^{-j2\pi f t} \tag{18}$$

The summation limits consider the valid time interval of the window function $w(t)$ centered at $t = 0$. There are multiple window functions to choose from, such as the Rectangle window $w(t) = 1$, $0 \le t < n_w$, the Hann window $w(t) = \sin^2\left(\frac{\pi t}{n_w}\right)$, $0 \le t < n_w$ and the Gaussian window $w(t) = e^{-\frac{1}{2}\left(\frac{t-(n_w-1)/2}{\sigma(n_w-1)/2}\right)^2}$, where n_w is the window length and σ in the Gaussian window is a preset parameter for window length. STFT is often implemented using FFT [177].

DWT. There is a trade-off between the resolution in the time and the frequency domain: If STFT uses a longer window to increase frequency resolution (more values for k due to larger $n = n_w$ in Equation 17), this yields a "less instantaneous", or temporally less localized, frequency spectrum at a time point, because a longer window averages varying dynamics in larger proximity. A logical next step is to use windows of different sizes. Wavelet Transform (WT) introduced by Morlet et al. [183] for seismic data analysis attacks this problem by conceptually scanning the data with temporally scaled versions of a prototypical time-limited signal $\psi(t)$ called (mother) wavelet. In the framework given by Equation 15, $\mathcal{K}(t,\boldsymbol{k}) = \psi_{a,b}(t) = |a|^{-1/2}\, \psi((t-b)/a)$ holds, where $\psi_{a,b}(t)$ is the wavelet, the parameter a scales the time span of ψ, and the factor $|a|^{-1/2}$ scales its amplitude to preserve energy (integration of the squared signal); the parameter b translates ψ along the time axis, conceptually scanning the data, like τ for STFT. Because ψ is time-limited, it captures temporally localized information at each time point. This form of WT for continuous-time signals is called Continuous Wavelet Transform (CWT). For the discrete-time and time-limited cases, Discrete Wavelet Transform (DWT) proposed by Mallat et al. initially for image processing [171, 172] is more rele. By degenerating a space-limited two-dimensional image into a time-limited one-dimensional time series, DWT for time series is defined as By degenerating a space-limited two-dimensional image into a time-limited one-dimensional time series, DWT for time series is defined as

$$\mathcal{W}\left(\widetilde{a},\widetilde{b}\right)=\sum_{t=0}^{n-1}y_t\psi_{\widetilde{a},\widetilde{b}}(t)\ \text{and}\ \psi_{\widetilde{a},\widetilde{b}}(t)=2^{-\frac{\widetilde{a}}{2}}\psi\left(\frac{t-2^{\widetilde{a}}\widetilde{b}}{2^{\widetilde{a}}}\right) \tag{19}$$

where the temporal scaling parameter is $a=2^{\widetilde{a}}$ and the temporal shifting parameter is $b=2^{\widetilde{a}}\widetilde{b}$. The algorithm will therefore scan the data with higher resolution (smaller b) for higher frequency (smaller a). This adaptive sampling strategy based on the power of 2 is called dyadic sampling [177]. There are multiple wavelets to choose from. Some of them are more suitable for CWT, such as the Morlet wavelet in Equation 20 (where f_0 is the central frequency set by the user) that is used most frequently in time-frequency analysis [177]; some are more geared to DWT, such as the popular Haar wavelet [212]

$$\psi(t)=\pi^{-\frac{1}{4}}e^{j2\pi f_0 t}e^{-\frac{t^2}{2}} \tag{20}$$

$$\psi(t)=\begin{cases}-1 & \text{if } \frac{1}{2}\le t<1\\ 1 & \text{if } 0\le t<\frac{1}{2}\\ 0 & \text{otherwise}\end{cases} \tag{21}$$

Similar to DFT and FFT, DWT also has an accelerated implementation called Fast Wavelet Transform (FWT) [172].

HHT. WT relies on the practitioner's expertise to choose a fixed kernel function which can be demanding and inflexible. The Hilbert-Huang Transform (HHT) [116, 117] fills this gap by deriving kernels from the data. This is achieved by Empirical Mode Decomposition (EMD) that splits a time series into a set of complete and orthogonal Intrinsic Mode Functions (IMFs) in a data-driven way, as indicated by the word "empirical". Yet, as in the time-frequency analysis, instead of the waveforms in the time domain, the analyst is more interested in their intensity at each time. Hilbert transform bridges the gap. It models the measured signal (IMFs in this case) as the real part of a complex signal and derives the hidden imaginary part of the signal. With the real and imaginary part, the energy and phase of the complete complex signal are readily available. Visually, it is like drawing the envelope of the original oscillating IMFs. Mathematically, the Hilbert transform uses the kernel $\mathcal{K}(t,\boldsymbol{k})=p.v.\frac{1}{\pi(k-t)}$ in Equation 15, where $p.v.$ stands for Cauchy principal value for skipping the non-integrable point $k=t$. The kernel has the property of shifting all frequency $\frac{1}{\pi}$ radius back (e.g., cosine becomes sine). Like WT, HHT emerges from the need for seismic data analysis [118], but established as a time-frequency analysis method in various domains, such as analyzing vibration in mechanical engineering, climate patterns in metrology, medical data (like ECG and EEG), etc [65, 116]. Some scientists consider it the most appropriate tool to deal with nonstationary and nonlinear signals [65].

Unlike the stochastic process models with clear lineage, integral transforms have many branches that we cannot cover in the limited space in this section. For instance, we omit the preeminent Laplace transform for continuous signals and Z-transform for discrete signals because they are more geared towards analyzing systems processing signals, e.g., for control system design and digital filter design. They are also seldom seen in TSA. Readers are referred to literature on integral transforms [68, 86, 192] or (digital) signal processing [177, 207] for more information.

Most integral transform representations are mainly used for data compression, denoising, and feature extraction since they originate mainly from signal processing, but they are also seen in other tasks like anomaly detection [59], clustering [77], and retrieval [201]. Method choice is often driven by domain knowledge. For instance, the diagnosis of a bunch of rotation parts in a machine would motivate an analysis in the frequency domain. If the spectrum varies over time, and the variation carries information relevant to the domain, the analyst may go for an analysis in the time-frequency domain.

5.3 Piecewise Representation

With piecewise representations, we are mainly limited to the time-domain representations that are studied much in TSA. Piecewise representations assume piecewise "homogeneity" along the time axis in the data and describe each piece with a simpler representation like a constant, a line segment, a polynomial, and so on.

The simplest piecewise representation is Piecewise Aggregate Approximation (PAA) [137, 139], which uses the mean value to represent each piece. The piece length in Piecewise Constant Approximation (PCA) is fixed. Adaptive Piecewise Constant Approximation (APCA) [137] extends PAA by making the piece length variable and adaptive, so that there can be more and shorter pieces in temporal regions with a concentration of high volatility, while fewer and longer pieces in relatively stationary temporal regions. Instead of mean values, Piecewise Linear Representation (PLR) [135] uses a linear segment to represent each piece. The next extension is Piecewise Polynomial Representation (PPR) [93] which uses a polynomial to represent each piece. Based on PAA, SAX [160, 161] assumes normal distributions of data values and quantifies them. Then, it maps each value-range bin to a symbol to convert a time series into a string to enable methods for text processing like regular expression [273]. This soon became one of the most popular symbolic representations for time series and witnessed many extensions [228, 252]. Some surveys on TSRTs feature a separate category called "symbolic representation" [259]. From our perspective, the dominant technique behind SAX is PAA and quantization. Notwithstanding methods transfer from other domains, assigning symbols does not alter the information much.

Piecewise representations are initially designed for time series retrieval with much consideration of indexing capabilities like lower-bounding existing distance measures [45, 137, 161]. Nonetheless, they make few assumptions on the data and are general-purpose. They are very efficient in terms of compressing data massively and have an edge in capturing temporal dynamics [135, 137]. Hence, piecewise representations can be used especially when analysts need to smooth the data, remove outliers, or compress the data.

5.4 Machine Learning Model

According to [144], Machine Learning (ML) models are statistical algorithms that can learn from data and generalize to unseen data, and thus perform tasks without explicit instructions. Strictly speaking, many models, especially stochastic process models, are also ML models. Yet, we include in this category only the most general ML models, like random forests, Support Vector Machine (SVM) and LSTM. These are applied to time series but also other data, *e.g.*, tabular, text, image, audio, and video.

SVM. A classification model conceptually draws boundaries separating the classes. The initial idea of SVM is to draw linear boundaries [32]. Linear SVMs enjoy good explainability [97]. However, some classes are entangled and cannot be separated linearly in the original feature space. Contrary to the idea of reducing dimensionality in Section 5.5, nonlinear SVM maps samples / data points to a higher-dimensional space, even theoretically up to an infinite-dimensional space with the so-called kernel trick [32]. The new dimensions/features may enable a linear separation of the classes. SVM is mainly used for classification, such as text classification [126], image classification [44], speech recognition [232], though also applicable to regression, such as power load forecasting [262]. Likewise, it finds applications in time series classification [129, 278] and forecasting [223].

Decision Tree Ensembles. Another ML model, namely the decision tree, assigns a sample recursively to two [38] or more subgroups [204]. Visually, it looks like going from the root node of a tree, over several in-between nodes, to one of the leaf nodes. A node is split based on the feature that can result in the cleanest division of samples among the child nodes in the case of classification. For regression, a node is assigned a value, and the node is split to minimize errors between data values and the corresponding child node value. Visually, a decision tree recursively partitions the feature space with hyperplanes, each of which is perpendicular to the axis of a dimension in the original feature space. Like the linear SVM, the decision tree is also well interpretable, but quickly reaches its limitation as data complexity increases. The prediction power increases when combining multiple decision trees to form an ensemble. One variant of such ensembles is random forest [112]. It trains multiple decision trees with different data subsets. During inference, predictions from individual decision trees are aggregated by frequency count (for classification) or averaging (regression). Random forest is particularly effective when the number of features is large compared with the number of training samples [33, 286]. Another common ensemble option is boosting. It trains the decision trees one after another, each improving the result from the predecessor. For instance, the two most popular boosting methods AdaBoost [89] and gradient boosting(e.g., XGBoost [51]) achieve this by overweighting incorrectly predicted samples and minimizing the prediction errors when training the successor, respectively [95]. One noteworthy benefit of tree ensembles is the readily available feature importance estimated based on the usage frequency and effect (reduction of class impurity in nodes) of the features in all trees. It adds to explainability and helps with feature selection [286]. Random forest and boosting methods are frequently used in image classification from remote sensing [22], fraud detection [106, 188, 211], variable estimation in hydrology [187, 242]. In TSA, they have been competing with other ML models in various classification [124, 148] and regression [169, 185, 241] tasks.

MLP. The recent AI tsunami witnesses the great success of ANN, especially in computer vision and natural language processing. An Artificial Neural Network (ANN) usually consists of multiple stacked layers. The most common layer type, the fully connected layer, first linearly transforms the input data sample, weighting various features in the input data sample. The output undergoes an activation function, such as SELU, ReLU, sigmoid, etc, which imbues the transformation with nonlinearity. Visually, the linear hyperplane in the original feature space described by the linear transform bends or curves by the nonlinear activation function, making it a better building block for approximating the function to model. A sequential stack of fully connected layers is called a Multilayer Perceptron (MLP) [215]. In TSA, it is most seen in forecasting [31, 78, 227]

CNN. One issue with MLP is that its number of parameters grows quickly as the layer size and number of layers increase. This issue is alleviated by the convolutional layers used in Convolutional Neural Networks (CNNs). It scans the data with multiple convolution kernels, analogous to multiple biological neurons in the visual cortex stimulated only by specific patterns in local regions of the image from the visual perception. This technique uses fewer parameters, compared to assigning an individual weight to each feature as with a fully connected layer. The efficient parameter usage exploits the spatial translation invariance of a pattern in an image, or the temporal translation invariance of a pattern in a time series. CNNs are predominant in computer vision [283], and 1D CNNs are established in TSA for classification [162, 282] and forecasting [143, 245, 251].

RNN. MLPs and CNNs allow only fixed input data size. In addition, they are memoryless, producing the same output when providing the same input. When modeling an evolving process, as is often the case in TSA, it is sensible to allow a variable input data length and store historical information for future inference. Recurrent layers in Recurrent Neural

Networks (RNNs) address these issues. A recurrent layer feeds the output back to itself. The input is the observed values of one time point, and the RNN makes inference for each time point iteratively. This way, RNNs accept a long sequence of data and considers historical data when making predictions. Modern recurrent layers like the LSTM layers in Long Short-Term Memorys (LSTMs) [113] and the GRU layers in Gated Recurrent Units (GRUs) [54] employ more complex mechanisms to manage the retaining, referencing, and forgetting of historical information, leading to longer memory and better performance. RNN variants are commonly used to process data with a notion of order, such as time series (especially forecasting) [9, 197, 253, 263], text [119, 155], and speech [119]. Regular RNNs predict the present values based on information up to the current input. This is reasonable for, e.g., time series forecasting. However, looking into the inputs in the future is sometimes desired, as in language translation. Bidirectional RNNs solve this problem with two RNNs reading data from both directions. It helps in TSA with interpolation [42] and classification [25, 140].

Attention. RNNs are computationally expensive to train, and their retention of information reaches limitations in challenging tasks like long-text translation. The attention mechanism [18] used extensively in transformers [244] solves this problem by accessing all previously seen data but focusing more on important ones according to the weights estimated by a small sub-model trained with the whole model. Compared with RNN, not only does attention mechanism improve memory and training efficiency, but also explainability as the analyzer can examine it the model is paying attention to the correct part of the information during inference [213, 235]. Initially designed in Natural Language Processing (NLP) [10, 193] and permeating into computer vision [141, 163], the attention mechanism and Transformers are gradually being tested in TSA [3, 276].

Autoencoder and VAE. The previous ANNs are mainly used in supervised learning for classification and regression. Their advancements lie in the design of special layers. In contrast, the unsupervised learning technique autoencoder [145, 218] learns a data representation with the help of its symmetric encoder-decoder architecture. The encoder consists of layers with a decreasing number of neurons, and the decoder layers with an increasing number of neurons. It is trained with the same data both as input and output, so that it learns an identity transform. The output of the encoder is effectively a lower-dimensional latent representation of the original data, which can be used for feature extraction or visualization. The decoder output is a reconstruction from the latent representation, which can be used as denoised/repaired data. The deviation of the reconstruction from the input data is indicative of whether the input data is similar to the training data, which can be used for anomaly detection. Inheriting the basic architecture and functions of the autoencoder, the Variational Autoencoder (VAE) lets the encoder output the meanings and standard deviations of a multivariate normal distribution. The decode draws its input from this random distribution. This alteration grants VAE the power to generate synthetic data by drawing samples from the random distribution and feeding the samples to the decoder. In TSA, autoencoders and VAEs serve similar functions, e.g., anomaly detection [41, 50, 154], feature extraction [148, 261], denoising [115, 219], synthetic data generation [73, 153], but also classification [271] and forecasting [109, 186].

GAN. Though capable of generating synthetic data, the images generated by VAEs are of lower quality than the original data. GAN [99], another ANN for generative learning, is able to generate images of similar quality as the original data. It features two ANNs competing with each other. One of which is called a generator, which creates synthetic data samples based on the input drawn from a random distribution. Another is called a discriminator, which judges if a data sample is from the original data or created by the generator. They are trained one after another in turns. When the discriminator is under training, it receives half of the training samples from the original dataset with labels 1 and another half synthesized by the generator with labels 0. When training the generator, the whole GAN draws samples from the random distribution with all labels being 1. Meanwhile, the parameters of the discriminator are frozen. The training iterates, until the generator creates realistic samples and the discriminator is forced to guess. After successful training, the generator can be used for data synthesis. In addition to data generation [128, 159], GANs are also well-known for style transfer. In TSA, GANs have already contributed to various tasks [39] like data generation [122], classification [249], anomaly detection [149, 281], data augmentation [157], and denoising [166, 166].

Many above techniques can be combined, e.g., convolutional recurrent autoencoder [269, 284], self-attention GAN [279]. ML models make few assumptions on the data, but generally require large amounts of (labeled) training data. They may specialize in certain tasks, as we discuss in Section 6.3. The difference in task suitability notwithstanding, many machine learning models, though diverse in mechanism, architecture, and training method, are often interchangeable in application. Indeed, many ML models listed in Table 5 share similar use cases.

In practice, ML models require organizing the inputs and outputs for the model as columns in a table, where each row corresponds to a sample/instance, and each column an input variable or an output variable. In the case of time series, a row in such a table can correspond to 1) a whole time series, e.g., for time series classification and clustering [8, 120, 178]; 2) a fragment of the time series, perhaps segmented by a sliding window, e.g., for pattern retrieval [148, 274]; 3) a time point in a time series, e.g., for pointwise anomaly detection and time point classification [216, 275]. The time series may need to be flattened in the first two cases if the model does not support multiple channels. Once the data is formatted in such a tabular way, given the similar model API, many models can solve the problem. According to our experience, model selection is not necessarily the primary concern during the application of machine learning, nor is architecture search or hyperparameter optimization. Instead, it is the formulation of the problem, i.e., what to choose as the model input and output. For instance, a classic method for time series retrieval scans the data with a sliding window and classifies patterns in the window as relevant or not. This approach can only retrieve patterns of one size in one scan. Also, the classification is based on the pattern alone, oblivious of its background. If the analyst aims at retrieving the stationarity phases in the data with fewer distinctive features for recognition, it is challenging to pinpoint the start and end time moments of the stationarity phase without context knowledge. An alternative approach [272] is to classify time points in the middle of the sliding window as relevant (in a target pattern) or not; and merge relevant time points with density-based clustering to get the intervals of relevant patterns. The model and its inputs are retrained, but the output – and with this the formulation of the problem for the model – change, raising accuracy and speed significantly and jointly. As such, instead of model comparison with the same model inputs and outputs for already framed problems, we encourage more research in the direction of modeling the problem itself.

5.5 Dimensionality Reduction Technique

Dimensionality Reduction (DR) techniques transform samples / data points from a high-dimensional space to a lower-dimensional one. This helps with discovery and/or visualization of data patterns that are otherwise hard to detect in the original space [95]. Similar to the case with the category Machine Learning Models, we confine the scope of this category to general DR techniques like Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) while excluding techniques that can be used for dimensionality reduction but technically more affiliated to other technical lineages, like autoencoder and most piecewise representations. There are two types of DR techniques, namely linear dimension-transforming methods and nonlinear distance/neighborhood-preserving methods (aka manifold learning) [13, 95]. The former linearly maps the original high-dimensional data space to a lower-dimensional image space, while the latter endeavors to reproduce pairwise distances in the original space between data points in the image space.

Linear Dimension-Transforming Methods. **PCA.** During the literature review, we found that PCA introduced by Pearson et al. [131] is dominant in linear DR techniques. It has been a popular TSRT in various tasks, including but not limited to forecasting [191], segmentation [19], clustering [152], and retrieval [130]. PCA reduces and rearranges the coordinate system axes called Principal Components (PCs), such that the new axes capture as much data variance as possible. Specifically, given n data points with m centered (zero-mean) dimensions represented by a matrix $X \in \mathbb{R}^{n\times m}$, where each row corresponds to a data point and each column a dimension in the original space. The basis vector for the first PC $\boldsymbol{v}_1$ is the unit vector along which data variance ($\|X\boldsymbol{v}_1\|^2 = \boldsymbol{v}_1^T X^T X \boldsymbol{v}_1$) is maximized. The basis vector for the second PC can be computed by iterating this process after subtracting the parts of data explained by already found PCs ($X \coloneqq X - X\boldsymbol{v}_1\boldsymbol{v}_1^T$) because the basis vectors of the PCs are orthogonal. Subsequent PCs can be calculated similarly. Mathematically, computing PCA is equivalent to finding the eigenvectors for the covariance matrix $S = X^T X/(n-1)$. PCA often serves as a feature extraction step in TSA. For instance, distance measures can be applied to PCA embeddings instead of the raw time series [130, 266]. During k-means-based time series clustering, PCA can be used to construct a common projection for all data points / time series in a cluster, and the reconstruction error of each time series projected on the corresponding common projection axes are used to reassign the cluster [152]. A similar technique is proposed for time series classification [151]. It creates a common projection for each class based on the time series in the training data, then, classifies an unlabeled time series to the class whose projection produces the lowest embedding variance for the time series.

SVD. PCA can be implemented efficiently with Singular Value Decomposition (SVD), a technique for decomposing a matrix $M = U\Sigma V^T$ (limited to real cases), where $M \in \mathbb{R}^{n\times m}$ is a matrix to decompose (n is the number of data points and m the number of data point dimensions); $U \in \mathbb{R}^{n\times n}$ is orthonormal and called the left-singular vector; and $V \in \mathbb{R}^{m\times m}$ is orthonormal and called the right-singular vector. Let $M = X$, then, V contains the basis vectors of the PCs as column vectors. This can be proved by substituting X with its SVD decomposition and taking $U^T U = I$ into account, i.e., $S = X^T X/(n-1) = V\Sigma U^T U \Sigma V^T/(n-1) = V\left(\Sigma^2/(n-1)\right)V^T$. SVD itself can be used for dimensionality reduction and seen in TSA [40, 142, 257]. Keogh et al. proposed the first implementation of SVD for time series indexing. Specifically, they scan a univariate time series with a sliding window [137]. Each segment in the window is potentially a hit during retrieval and corresponds to a row in X; the window length $n_w = m$.

SSA. The approach from Keogh et al. is very similar to Singular Spectrum Decomposition (SSA) [217], a DR technique primarily for TSA. It is well established in meteorology [217], has gained much popularity in general TSA [30, 108], and recently enters other fields like image processing [98]. SSA decomposes a time series into interpretable components (e.g., trend, periodic components, and noise). In SSA, the matrix decomposed by SVD is the n_w-lag trajectory matrix defined as $X = \left[x_{i,j} = y_{i+j}\right]_{\substack{0\le i<n_w \\ 0\le j\le n-n_w}}$. In principle, SVD obtains the basis vectors for the PCs of V that best accounts for the variances in X. SSA proceeds by calculating the PCs or embedding for the data points. For instance, a univariate time series segment $\boldsymbol{y} = [y_t]_{t_0\le t<t_0+n_w}$ can be projected to its embedding with $V^T\boldsymbol{y}$. The first PCs with their corresponding basis vectors as characteristic waveforms ideally explain the majority of data variance. Reviewing the mechanism of SSA, its relation with PCA and with the discrete Karhunen–Loève Transform (KLT) (equivalent to PCA but from the perspective of functional analysis and integral transform) is obvious, though the name SSA honors SVD.

LDA. If the class labels are available, the analyst may want to exploit this information. Linear Discriminant Analysis (LDA) [87] finds the best coordinate bases like PCA. However, instead of the axes that best account for the data variance, LDA computes the axes that maximize the separation of the classes, i.e., maximizes the distance between class centers and minimizes intra-class variance. We did not find many applications of LDA in TSA. Gao et al. use LDA when classifying EEG data, but based on extracted features from the time series rather than the time series themselves [94]. Shah uses LDA in time series forecasting, but for model selection, where LDA classifies each time series based on extracted features to the best forecasting method [225].

Nonlinear Distance/Neighborhood-Preserving Methods. **General Mechanism.** Complex data manifolds cannot be separated into clusters by changing the perspectives. For PCA, it is required that the first PCs account for most variances in the data, so that the subsequent usage, e.g., a lower-dimensional scatter plot according to PCA is valid. This does not always hold. Nonlinear DR techniques approach this problem by reproducing the distances or neighborhood structure in the original high-dimensional space in the lower-dimensional image space. For instance, Sammon mapping [222] strives to minimize with e.g. gradient descend the difference between the distances in the two spaces described by a stress function

$$\frac{1}{\sum_{i<j} d_{ij}} \sum_{i<j} \frac{\left(d_{ij} - d'_{ij}\right)^2}{d_{ij}} \tag{22}$$

where i and j are indexes of data points, d_{ij} are the distance between i-th and j-th data points in the original high-dimensional space, and d'_{ij} are the distance between i-th and j-the data points in the projected lower-dimensional space. The choice of distance is agnostic but defaults to Euclidean Distance (ED), and the data points in the image space are randomly initialized [222]. Another popular DR technique t-SNE [111, 243] calculates the pairwise similarity between data points in the high-dimensional original space by inverting their ED according to a normal distribution interpreted as probabilities of the pair being neighbors. Then, it computes the pairwise similarity between initially randomly initialized data points in the lower-dimensional image space by inverting their ED according to a t-distribution interpreted likewise as probabilities. Finally, it minimizes the Kullback–Leibler (KL) divergence between the similarities/probabilities in the two spaces with gradient descent. The similarities or distances between data points may not be treated equally.

Special Traits. Specifically, some techniques like Isomap [64, 256], Locally Linear Embedding (LLE) [173, 285], and t-SNE prioritizes local structures. Like LDA, if categorical labels are available, Uniform Manifold Approximation and Projection (UMAP) that are technologically similar to t-SNE can optionally take advantage of this additional information and be used in a supervised fashion. Interestingly, t-SNE and UMAP, two of the most popular DR techniques for data visualization, are less used in TSA and their applications are still mostly in visualization [70, 260]. In fact, [148] benchmarked seven representations involving UMAP on their data and conjectured that while useful for visualization, techniques like t-SNE and UMAP, are not effective in capturing visual patterns in time series. Luckily, advancements in DR techniques for time-dependent show promising candidates for TSA [247], such as dynamic t-SNE (dt-SNE) [209] and temporal MDS [121].

Axes to Apply Dimensionality Reduction. In this section, we used the term "data point", whose counterpart in TSA deserves explanation. Because time series can be multivariate, analysts may apply DR to the temporal domain, possibly piecewise (without overlapping) or with a sliding window (with overlapping) [92], as well as to multiple tracks for each time point [19, 130], or to both axes simultaneously [266]. According to our experience, brute-force application of DR on raw time series does not necessarily contribute much to knowledge extraction. Instead, it is preferred to extract features according to domain knowledge and apply DR on this tabular data. For instance, to diagnose the regeneration failure in diesel particulate filters, we extracted features like the particulate mass before exiting each regeneration, the accumulated total fuel injection during each regeneration, the maximal exhaust temperature during each regeneration, etc. We applied various DR techniques on these features, rather than on the raw signals. Not only did the results reveal various error roots in clearer clusters, but the extracted features contributed to explainability.

Other Techniques with Dimensionality Reduction Function. Other noteworthy DR techniques not mentioned previously included Multidimensional Scaling (MDS) [164, 210, 246] and Self-Organising Map (SOM) [92, 110]. However, we list the autoencoder under the category Machine Learning Model because it is technically more affiliated with ANNs. Please also note that while virtually all piecewise representations and PIP (Section 5.6) claim explicitly that they conduct dimensionality reduction, we only register dimension-transforming and distance/neighborhoold-preserving techniques in this category, because, as mentioned at the beginning of this section, our categorization is based on technical affinities, rather than functions, which each technique can serve many.

DR techniques are relatively general-purpose and serve in many tasks. There seem to be no universally applicable hints about choosing the best DR technique to represent a time series. Nor are there always clear rules for parameter setting. For instance, it is advised to set t-SNE's perplexity parameter (the estimated number of neighbors), which controls the balance between local and global structures, between 5 and 50 [243, 254]. Nonetheless, complex datasets may deviate from such general settings. For instance, it requires setting the perplexity over 500 to obtain a visually meaningful 2D scatter plot from a 3D mammoth point cloud [58]. Therefore, it makes sense to try several techniques, vary their parameters, and choose the empirically best in specific use cases. Nonetheless, comprehensive study in [83] concluded that most DR techniques have relatively good and robust default parameter settings, though the validity of this finding in TSA remains to be examined. Generally, linear dimension-transformation methods tend to preserve the global patterns while nonlinear distance/neighborhood-preserving methods are good at revealing local patterns [13].

5.6 Miscellaneous techniques

Lastly, we group TSRTs that are hard to categorize, e.g., because they combine techniques from multiple previous categories, under the category "miscellaneous".

Prophet Model. One of the most impactful TSRTs in this category is Prophet from Meta (Facebook back then) [236]. In a nutshell, this combines 1) a model for describing the trend, like the saturating growth model or PLR suggested in the original paper; 2) Fourier series for capturing seasonal fluctuations; and 3) a (for each time point) binary "holiday" component accounting for user-given impulses in the data due to short events like big limited-time discounts. Specifically, the Prophet is defined as

$$y_t = g(t) + \sum_{i=1}^{o} \left(a_i \cos\left(\frac{2\pi it}{\widetilde{s}}\right) + b_i \sin\left(\frac{2\pi it}{\widetilde{s}}\right)\right) + z(t)\,\kappa \tag{23}$$

where $g(t)$ is the selected trend model; o is a hyperparameter for the number of Fourier series terms to use; a_i and b_i are Fourier coefficients for cos and sin, respectively; $\widetilde{s}$ is the season length, like 7 (days) for weekly seasonality, or 365.25 (days) for yearly seasonality; $z(t) = 1$ if the analyst has specified an event at time t, otherwise $z(t) = 0$; and κ follows a zero-mean normal distribution. The Prophet model is most active for univariate forecasting, especially in business [1, 221].

PIP. Another influential TSRT in this category is Perceptually Important Point (PIP) [55]. It represents a time series with visually salient points like the main peaks and troughs in the time series curve, see Figure 6. PIP centers around the temporal dynamics. It is versatile and finds applications in forecasting [240], classification [264], and motif discovery [90]. Various PIP algorithms exist. The classic one creates a list containing all time points in the original time series, sorted in descending order

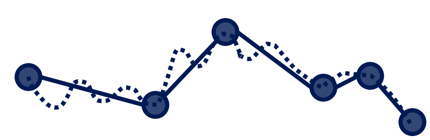

Fig. 6. ***Perceptually Important Point.*** *It represents a time series with visually salient points, usually the prominent peaks and troughs.*

of their importance [55]. Initially, the ranked list contains the first and the last time point in the original time series. The algorithm adds one time point to the ranked list iteratively, until the ranked list contains all time points in the original time series. The time point added in each iteration has the largest sum of (Euclidean) distances to the two temporally adjacent time points in the ranked list.

6 FACTORS TO CONSIDER DURING REPRESENTATION SELECTION

According to our literature analysis, there are five primary factors to consider when using TSRTs: 1) physical dynamics, 2) data assumptions, 3) task suitability, 4) technique transfer, and 5) computational resources. We describe these next and also provide examples.

6.1 Physical Dynamics

The first factor, physical dynamics, suggests that the best data representations ideally model the physical dynamics, e.g., derived from differential equations backed up by physical rules behind the data. This factor is especially relevant for integral transforms (Section 5.2). For instance, the physical properties of audio signals encourage the use of FFT because the sound is by nature a linear combination of vibrations of a range of time-invariant frequencies that may trace back to different physical sources. Energy consumption data typically consists of 1) a trend implying technological advances, behavior shifts, and long-term environment changes; 2) a seasonal fluctuation reflecting cyclic alternations of days and nights as well as seasons; and 3) a “holiday” component or random residuals capturing unusual events or unexplained errors. Accordingly, the Prophet model designed to capture the three components may fit the data well [236].

Since physical dynamics are primarily domain-specific, it follows that there may not be a universally (i.e., domain-agnostic) optimal TSRT for all tasks in TSA, just like that no time series similarity measure consistently outperforms the other [63]. As [45, 137] show, simple TSRTs like PAA may outperform complex ones like FFT, DWT, and SVD in capturing temporal shapes in the time domain in a time series. In these cases, the usage of the complex TSRTs is not substantiated by physical dynamics. As a result, not many benefits can be expected.

6.2 Data Assumptions

The second factor, data assumptions, requires that the analyst checks the fulfillment of the data assumptions made by the TSRT and make adjustments if necessary. Compared with physical dynamics, which entails essential assumptions on the domain-specific physics of the observed system or process, our second factor assumes certain properties of the data themselves, e.g., stationarity or linearity required by the method. This factor is most prevalent among stochastic process models (Section 5.1).

A typical example is when a model type has a parametric variant with a fixed number of parameters and a non-parametric variant that adds parameters as data increase, e.g., linear SVMs vs. nonlinear SVMs, or ANNs vs. Gaussian Processs (GPs) [184, 258]. When possible, the former is preferable for efficiency and ease of use; the more computationally expensive latter one may be needed when the performance of the former is insufficient.

There are established typologies of time series data. For instance, in signal processing, a signal can be categorized as continuous/discrete, deterministic/stochastic, time-limited/time-unlimited, causal/acausal, symmetric/asymmetric, periodic/aperiodic, energy/power/other, or (in terms of value range) bounded/unbounded [177]. However, they may not all be noteworthy in our scope. For example, time series data are by definition in Section 3.1 always discrete; causality (constantly zero for negative time) is often necessary to ensure a stationary initial state of the system in control engineering and may not be of interest in other applications. In [273], Yu et al. used SAX as a TSRT. However, the data values in the time series do not follow a normal distribution as required by SAX [161]. Consequently, they have to manually modify the algorithm to fit the use case.

We selected eight common data assumptions, which appear mostly frequently in the TSRTs we reviewed. These common data assumptions are

Stationarity: (Primarily for stochastic processes) the time series has a time-invariant mean, and the autocovariance of the time series depends only on the time lag.

Linearity: (Primarily for stochastic processes) a value in the time series can be described as a linear combination of values at other time points plus i.i.d random variables; for DR techniques): Data are explained by a linear combination of latent variables, i.e., linear DR techniques.

Markov Property: The values of the next time point depend only on the values of the current time point. Namely, the observed system is memoryless [229, 287]. Note that by encapsulating the values of multiple time points in a vector, some models like the ARIMA family (which do not have this property) can be formulated to have the property.

(A)Periodicity: Values in the time series repeat (or not) after a fixed number of time points.

Univariance: The time series has only one channel.

Normality: The values, errors, or other components of a time series are normally distributed.

Kernel: A prior choice of a kernel function is necessary, which, in turn, assumes certain properties of the time series data. Note that an integral transform always has a kernel by definition in Equation 15, but the kernel can be fixed or selected by the user according to prior knowledge. The latter is of concern here.

T+S+X: The time series can be decomposed as the sum of trend components (T), seasonal components (S), and some other components (X, e.g. noise, residual, remainder, error, innovation, irregularity, or holiday). A time series model consisting of interpretable components is called a structural model, and this kind of decomposition is the most common one [107, 229].

Most of the selected common data assumptions are unequivocal. However, the definition of stationarity and linearity requires refinement because they are overloaded with various meanings, especially when crossing the borders of disciplines.

Stationarity. We adopt the weak/weak-sense/wide-sense/autocovariance stationarity stating that a time series has a time-invariant mean and the autocovariance depends only on the time lag [75, 229].

Stationarity is mainly assumed by some stochastic process models. Stochastic processes see a time series as a realization of a sequence of random variables $[Y_t]_{0\leq t<n}$. The means and autocovariance refer to the random variables, namely, $\mathrm{E}\left(Y_t\right) = \mathrm{E}\left(Y_0\right)$ and $\mathrm{Cov}\left(Y_t, Y_{t+\Delta t}\right) = \mathrm{Cov}\left(Y_0, Y_{\Delta t}\right)$. If not observed this way, even the notion of "mean" and "variance" may not make sense. Let the time lag δt be 0, it follows that the variance of the time series should also be time-invariant.

In contrast to weak stationarity, the strong/strict/strict-sense stationarity requires that the joint distribution of the random variables in any same-sized subsequence of the time series is the same / time-invariant [75, 229]. To be precise,

$$\begin{aligned} &P\left(Y_t \leq c_0,\ Y_{t+1} \leq c_1 \dots Y_{t+\Delta t} \leq c_{\Delta t}\right) = \\ &P\left(Y_{t'} \leq c_0,\ Y_{t'+1} \leq c_1 \dots Y_{t'+\Delta t} \leq c_{\Delta t}\right) \end{aligned} \tag{24}$$

where P denotes probability, and $c_i \in \mathbb{R}$ [229]. In practice, strong stationarity is less used because many real-world time series violate this property, and its verification is costly.

Many stochastic processes require the data to be stationary. For instance, it is necessary to conduct statistic tests like the Dickey–Fuller test to validate a trained AR model to examine the existence of a unit root [229]. In non-stationary cases, the time series must be specially treated to remove non-stationarity, e.g., by differencing [220].

The term "stationarity" is overloaded with multiple meanings. For instance, it is common to see the comparison between FT, STFT, WT in the literature, where the first only applies to "stationary" data and the other two to "non-stationary" data [118]. Stationarity here means a time-invariant spectrum. In fact, DFT can be applied to non-stationary (in the sense of our adopted meaning) time series.

Linearity. Linearity in this STAR means that the TSRTs can only capture linear dynamics of the observed system reflected in the data.

Linear stochastic process models assume that the present value Y_t or present variance σ_t^2 is a linear combination of other variables, including previous values of endogenous variables or their errors, current or previous values of exogenous variables, or previous variance. Nonlinear stochastic process models often have nonlinearity by making the constant parameters in linear ones changeable according to certain assumptions.

Linearity as a data assumption is not often concerned for integral transforms and piecewise representations. Many integral transforms, like FT and WT are considered "linear", which is rather a convenient property instead of a constraining assumption. This "linearity" refers to the transform operation $\mathcal{I}$ itself, i.e.,

$$\mathcal{I}\left(\alpha y_t + \beta y'_t\right) = \alpha \mathcal{I}\left(y_t\right) + \beta \mathcal{I}\left(y'_t\right) \ \forall t \tag{25}$$

where y_t and y'_t are two signals / univariate time series, and α, β are constants. Unlike e.g. linear stochastic process models, where linearity governs the data dynamics, the linearity of the operation in integral transforms does not necessarily reflect linear data dynamics.

Like linear stochastic process models, linear machine learning models assume a linear relationship between their output and input variables, e.g. linear regression and linear SVM. Nonetheless, nonlinear machine learning models like ANNs are more representative in this category.

Linear DR techniques assume that the time series to represent can be explained by a linear combination of latent components. It can be regarded as a linear transformation of coordinates or a linear projection of the data from a higher-dimensional space to a lower-dimensional one. Data distributed on anything but a hyperplane in a space cannot be disentangled linearly. Nonlinear DR techniques aim at preserving in the lower-dimensional image space the distance in the original high-dimensional space, or the neighborhood structure, during which nonlinearity may emerge.

As a side note, we do not count the additive linear decomposition of a time series into trend, seasonal, and other components, as described by the last data assumption, as linear because these components may contain nonlinearity individually.

6.3 Task Suitability

The third factor, task suitability, implies that TSRTs may not be general-purpose. Namely, they may perform well in some tasks (Section 3.2) but are unproven or possibly even fail in others. In the latter case, we may not know all facts because failed results tend to remain undisclosed. As such, the same failure is potentially found repeatedly by different researchers unaware of each other's work. This factor influences stochastic process models (Section 5.1) and machine learning models (Section 5.4) most. However, we attribute different reasons to their preferred usage in certain tasks.

For instance, many TSRTs emerging from econometrics are geared to prediction, like ARIMA for forecasting the mean, and ARCH for the variance/heteroscedasticity/volatility. They are seldom used in other tasks like classification [250]. Such TSRT-task combination can result from historical backgrounds or domain needs and may not reflect the general ability of the TSRTs themselves. A promising research direction is to benchmark the suitability of various TSRTs commonly used for one task also in others. We imagine, then, for instance, creating a skill matrix, where columns denote tasks and rows denote TSRT, also showing cells of ineffective matches.

Different machine learning models may favor different tasks. Therefore, we mention the suitable tasks for each model in Section 5.4. For instance, we mentioned in Section 5.4 that autoencoders are typically for anomaly detection and denoising while VAEs for data generation, though not really prohibited in other tasks. In contrast to stochastic process models, in our view, the reason for the concentrated usage of some machine learning models in certain tasks lies in the model function and structure itself. We believe that the structure of the model determines the model's function and ultimately influences the model's task suitability. For an autoencoder, for instance, the output layer reproduces the data, whose dissimilarity to the original data measured by the reconstruction error reflects the novelty of the input data and thus naturally relates to anomaly detection. The embedding layer of an autoencoder has much fewer neurons than the input, which can be interpreted readily as dimensionality reduction; when using 2 or 3 neurons, this can be directly used for data visualization. Should the analyst stick to the model for other tasks due to its certain merits that the alternative options lack, either the task needs to be formulated or the model modified so that they align with each other.

6.4 Technique Transfer

Our penultimate factor, technique transfer, is to enable techniques that solve similar problems in another discipline.

For instance, it is a common practice to describe time series as a difference equation or state space representation [104]. They are standard models in signal processing and control engineering and enables analysis akin to techniques in these disciplines, e.g., stability analysis [37].

In the example mentioned in Section 6.2, the authors utilized a symbolic representation, SAX, with nice properties like an ordered alphabet and the numerosity reduction [161], which can be exploited by text retrieval techniques, in this case, regular expression, to describe the distortions in time series patterns.

Another interesting example is to transfer a time series to an image (called time series image) first with techniques like recurrent plots; then, use image processing networks to classify the time series (images) [67] or conduct vision-based anomaly detection [62].

6.5 Computational Resources

Our last factor, computational resources, considers execution time and memory consumption. This is especially relevant in sensor monitoring and IoT where data volumes are huge. For instance, a Boeing 787 can generate half a terabyte of sensor data per flight [214]; Moreover, data processing is sometimes performed on portable devices with limited computing resources. On one hand, this factor concerns the TSRT itself, especially for machine learning models (Section 5.4) when model training should be taken into consideration. On the other hand, analysts may want to accelerate subsequent operations on the representation.

There are three avenues to alleviate such problems. To begin with, many TSRTs have (hyper)-parameters controlling the data compression rate, e.g., the resolution of the spectrum for WT, the number of states and transitions in a Markov model, the number of hidden units in the output layer of the encoder in an autoencoder, and so on. Secondly, DR techniques (Section 5.5) and piecewise representations (Section 5.3) can compress the data significantly and drastically lower the needs for computational resources. Lastly, there are techniques designed to boost efficiency instead of effectiveness, like various time series indexing methods [12, 49, 195].

7 REPRESENTATIVE TIME SERIES REPRESENTATION TECHNIQUES

Based on our literature review, we provide Table 5 listing the most representative TSRTs, exposing their data assumptions and limitations, and enumerating their most typical use cases.

The common data assumptions are introduced in Section 6.2. We elaborate on them together with data assumptions unique to each TSRT in free text next to the common data assumptions.

The use cases reflect classic and established applications rather than the newest research. We emphasize but do not limit use cases to TSA, because 1) typical use cases of many methods do not lie in TSA but in e.g. image or language processing; 2) listing more use cases inspires more applications in TSA since a major portion of methods in TSA are transferred from other fields, see Section 3.2.

8 DISCUSSION

We next discuss additional points that influenced the creation of our survey.

Tasks Preference. Some TSRTs in our taxonomy solve the task directly, e.g., LSTM for time series prediction. Others are essentially preprocessing, i.e., create output that enters other methods dedicated to the task. Although this could be controversial, we do not consider it a problem for TSRTs to favor certain tasks, as also mentioned in Section 3.3. Consider the example of large language models. They are widely recognized as representations or models of languages. Yet, they are technically classification models, whose basic task is to pick words in their vocabulary to fill in missing words in a text. Whereas, they can be adapted to other tasks like holding conversations.

Establishment over Cutting Edge. Though we call our paper a STAR, we concentrate more on classic methods. Our main goal is not to present recent advancements but to create a taxonomy based on technical lineages. Firstly, consolidated methods form relatively stable branches of knowledge and innovation; recent advancements are like leaves, which are potentially so diversified that it is hard to trace the essential techniques. Secondly, classic methods are what one should begin with in practice, because they are proven, have many implementations, and accumulated experience. Since we cannot find a comprehensive taxonomy on TSRTs on the root methods, we need to start one. Lastly, mature methods are likely familiar to our readership, which may be of help when they peruse our proposal since the readers do not need to learn many new methods first.

Non-Exhaustive Methods. TSA is a broad topic, so our scope is quite extensive compared to existing STARs. Needless to say, the presented TSRTs have only scratched the surface. With the limited representative selections of TSRTs in each category, we strive to depict the paradigms of different technique strains.

Category "Miscellaneous". The category "Miscellaneous" in Section 5.6 accommodates TSRTs that do not fit into other categories. This implies that other five categories fail to cover all TSRTs. This is especially the case when 1) some TSRTs like Prophet are ensembles of equally important components from many other categories; 2) some TSRTs like PIP do not have many variations.

Blending of Categories. Our categorization of some methods is debatable. A method may have technical traits of multiple categories, and the techniques defining the categories are not mutually exclusive. For instance, we can categorize GP under stochastic processes; yet, GP is also well-accepted as a machine learning model. We categorized autoencoder under the category Machine Learning Model, albeit it can be also fit under Dimensionality Reduction

Technique. KLT is itself primarily an integral transform [177, 208]. Its continuous form can be seen as a model that decomposes a stochastic process into orthonormal functions/series linearly combined with uncorrelated random variables as coefficients. Meanwhile, its discrete form, aka Hotelling Transform, is equivalent to PCA, which is well-known as a DR technique. We tried to separate the categories as disparate as possible. It was carried out consistently following technical lineages. Nonetheless, overlaps are, as explained, unavoidable.

Rules for Method Selection. In this STAR, we have reviewed 38 TSRTs with dedicated paragraph headings. Many more TSRTs are briefly mentioned or explained in groups. Though elucidated with concepts, properties, and typical use cases, the myriad TSRTs may still overwhelm practitioners. For instance, should the analyst prefer ARIMA, LSTM, or Prophet for time series prediction? The analyst can only try a limited number of them. The next step is to create a set of rules for method selection with the help of the taxonomy. Particularly, the rules can use as selection criteria the problems in TSA introduced in Section 3.2, and the data assumptions introduced in Section 6.2.

9 CONCLUSION

In this STAR, we presented a systematic survey of Time Series Representation Techniques (TSRTs). We created a taxonomy of TSRTs based on their essential technical lineages, discussed the factors to consider when choosing a TSRT for a given context, and presented a list of representative TSRTs with their properties/assumptions and typical use cases. Our taxonomy serves as a starting arsenal for data scientists and domain analysts in search of effective TSRTs that may address their problems. It also helps starting researchers on TSA with an overview of the known world before embarking on their exploration into the unknown. Additionally, senior researchers in one or a few disciplines may benefit from our contribution by drawing on knowledge from other disciplines. While many STARs exist on time series, they tend to focus on a narrower and sometimes domain-specific perspective. Our taxonomy organizes TSRTs from diverse disciplines for various downstream tasks into one system. We can expect the development of methods in each category to address more challenging cases and crossbreeding among techniques from different categories. In the future, we would like to study and propose a set of practical criteria and rules for practitioners to choose the first candidate TSRTs for their use cases.

REFERENCES

[1] Christophorus Beneditto Aditya Satrio, William Darmawan, Bellatasya Unrica Nadia, and Novita Hanafiah. 2021. Time Series Analysis and Forecasting of Coronavirus Disease in Indonesia Using ARIMA Model and Prophet. *Procedia Computer Science* 179 (2021), 524–532. https://doi.org/10.1016/j.procs.2021.01.036

[2] Rakesh Agrawal, King-Ip Lin, Harpreet S. Sawhney, and Kyueok Shim. 1995. Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-series Databases. In *Proceeding of the 21th International Conference on Very Large Data Bases*. VLDB Endowment Inc., Zurich, Switzerland, 490–501.

[3] Sabeen Ahmed, Ian E. Nielsen, Aakash Tripathi, Shamoon Siddiqui, Ravi P. Ramachandran, and Ghulam Rasool. 2023. Transformers in Time-Series Analysis: A Tutorial. *Circuits, Systems, and Signal Processing* 42 (2023), 7433–7466. https://doi.org/10.1007/s00034-023-02454-8

[4] Suzanne Aigrain and Daniel Foreman-Mackey. 2023. Gaussian Process Regression for Astronomical Time Series. *Annual Review of Astronomy and Astrophysics* 61, 1 (2023), 329–371. https://doi.org/10.1146/annurev-astro-052920-103508

[5] Sara Alaee, Ryan Mercer, Kaveh Kamgar, and Eamonn Keogh. 2021. Time Series Motifs Discovery Under DTW Allows More Robust Discovery of Conserved Structure. *Data Mining and Knowledge Discovery* 35, 3 (2021), 863–910. https://doi.org/10.1007/s10618-021-00740-0

[6] Abdulkader Aljandali and Motasam Tatahi. 2018. Economic Forecasting using ARIMA Modelling. In *Economic and Financial Modelling with EViews*, Abdulkader Aljandali and Motasam Tatahi (Eds.). Springer International Publishing, Cham, 111–142. https://doi.org/10.1007/978-3-319-92985-9{_}7

[7] Aziz Alotaibi. 2020. Deep Generative Adversarial Networks for Image-to-Image Translation: A Review. *Symmetry* 12, 10 (2020), 1705. https://doi.org/10.3390/sym12101705

[8] Ali Alqahtani, Mohammed Ali, Xianghua Xie, and Mark W. Jones. 2021. Deep Time-Series Clustering: A Review. *Electronics* 10, 23 (2021), 3001. https://doi.org/10.3390/electronics10233001

[9] Ibtissam Amalou, Naoual Mouhni, and Abdelmounaim Abdali. 2022. Multivariate Time Series Prediction by RNN Architectures for Energy Consumption Forecasting. *Energy Reports* 8 (2022), 1084–1091. https://doi.org/10.1016/j.egyr.2022.07.139

[10] Haadia Amjad, Mohammad Ashraf, Syed Sherazi, Saad Khan, Muhammad Fraz, Tahir Hameed, and Syed Bukhari. 2023. Attention-Based Explainability Approaches in Healthcare Natural Language Processing. In *Proceedings of the 16th International Joint Conference on Biomedical Engineering Systems and Technologies*. SCITEPRESS - Science and Technology Publications, Lisbon, Portugal, 689–696. https://doi.org/10.5220/0011927300003414

[11] Oliver D. Anderson, C. W. J. Granger, and A. P. Andersen. 1979. An Introduction to Bilinear Time Series Models. *Journal of the Royal Statistical Society Series C: Applied Statistics* 28, 3 (1979), 305. https://doi.org/10.2307/2347208

[12] Henrik André-Jönsson. 2002. *Indexing Strategies for Time Series Data*. Department of Computer and Information Science, Linköpings universitet, Linköping, Sweden.

[13] N. Andrienko, G. Andrienko, G. Fuchs, A. Slingsby, C. Turkay, and S. Wrobel. 2020. *Visual Analytics for Data Scientists*. Springer, [S.l.].

[14] Angela P. Ansuj, M. E. Camargo, R. Radharamanan, and D. G. Petry. 1996. Sales Forecasting Using Time Series and Neural Networks. *Computers & Industrial Engineering* 31, 1-2 (1996), 421–424. https://doi.org/10.1016/0360-8352(96)00166-0

[15] Charles K. Assaad, Emilie Devijver, and Eric Gaussier. 2022. Survey and Evaluation of Causal Discovery Methods for Time Series. *Journal of Artificial Intelligence Research* 73 (2022), 767–819. https://doi.org/10.1613/jair.1.13428

[16] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, and Y. Shkolnisky. 2008. A Framework for Discrete Integral Transformations I—The Pseudopolar Fourier Transform. *SIAM Journal on Scientific Computing* 30, 2 (2008), 764–784. https://doi.org/10.1137/060650283

[17] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, Y. Shkolnisky, and I. Sedelnikov. 2008. A Framework for Discrete Integral Transformations II—The 2D Discrete Radon Transform. *SIAM Journal on Scientific Computing* 30, 2 (2008), 785–803. https://doi.org/10.1137/060650301

[18] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. VLDB Endowment Inc., San Diego, CA, USA.

[19] Zoltan Banko, Laszlo Dobos, and Janos Abonyi. 2011. Dynamic Principal Component Analysis in Multivariate Time-Series Segmentation. *Conservation, Information, Evolution-towards a sustainable engineering and economy* 1, 1 (2011), 11–24.

[20] Ziv Bar-Joseph. 2004. Analyzing Time Series Gene Expression Data. *Bioinformatics (Oxford, England)* 20, 16 (2004), 2493–2503. https://doi.org/10.1093/bioinformatics/bth283

[21] Sumanta Basu and George Michailidis. 2015. Regularized Estimation in Sparse High-Dimensional Time Series Models. *The Annals of Statistics* 43, 4 (2015), 1535–1567. https://doi.org/10.1214/15-AOS1315

[22] Mariana Belgiu and Lucian Drăguţ. 2016. Random Forest in Remote Sensing: A Review of Applications and Future Directions. *ISPRS Journal of Photogrammetry and Remote Sensing* 114 (2016), 24–31. https://doi.org/10.1016/j.isprsjprs.2016.01.011

[23] Konstantinos Benidis, Syama Sundar Rangapuram, Valentin Flunkert, Yuyang Wang, Danielle Maddix, Caner Turkmen, Jan Gasthaus, Michael Bohlke-Schneider, David Salinas, Lorenzo Stella, François-Xavier Aubet, Laurent Callot, and Tim Januschowski. 2023. Deep Learning for Time Series Forecasting: Tutorial and Literature Survey. *ACM Comput. Surv.* 55, 6 (2023), 1–36. https://doi.org/10.1145/3533382

[24] A. Benveniste. 1989. Hannan E.J. and M. Deistler: The Statistical Theory of Linear Systems. *Statistical Papers* 30, 1 (1989), 239–242. https://doi.org/10.1007/BF02924328

[25] Navaneeth Bhaskar, M. Suchetha, and Nada Y. Philip. 2021. Time Series Classification-Based Correlational Neural Network With Bidirectional LSTM for Automated Detection of Kidney Disease. *IEEE Sensors Journal* 21, 4 (2021), 4811–4818. https://doi.org/10.1109/JSEN.2020.3028738

[26] Mickaël Binois and Nathan Wycoff. 2022. A Survey on High-dimensional Gaussian Process Modeling with Application to Bayesian Optimization. *ACM Transactions on Evolutionary Learning and Optimization* 2, 2 (2022), 1–26. https://doi.org/10.1145/3545611

[27] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A. Lozano. 2022. A Review on Outlier/Anomaly Detection in Time Series Data. *ACM Comput. Surv.* 54, 3 (2022), 1–33. https://doi.org/10.1145/3444690

[28] Tim Bollerslev. 1986. Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics* 31, 3 (1986), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

[29] Tim Bollerslev. 2010. Glossary to ARCH (GARCH). In *Volatility and Time Series Econometrics*, Tim Bollerslev, Jeffrey Russell, and Mark Watson (Eds.). Oxford University Press, Oxford, 137–163. https://doi.org/10.1093/acprof:oso/9780199549498.003.0008

[30] Pietro Bonizzi, Joël M. H. Karel, Olivier Meste, and Ralf L. M. Peeters. 2014. Singular Spectrum Decomposition: A New Method for Time Series Decomposition. *Advances in Adaptive Data Analysis* 06, 04 (2014), 1450011. https://doi.org/10.1142/S1793536914500113

[31] Pedro Henrique Borghi, Oleksandr Zakordonets, and João Paulo Teixeira. 2021. A COVID-19 Time Series Forecasting Model Based on MLP ANN. *Procedia Computer Science* 181 (2021), 940–947. https://doi.org/10.1016/j.procs.2021.01.250

[32] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, David Haussler (Ed.). ACM, New York, NY, USA, 144–152. https://doi.org/10.1145/130385.130401

[33] Anne-Laure Boulesteix, Silke Janitza, Jochen Kruppa, and Inke R. König. 2012. Overview of Random Forest Methodology and Practical Guidance With Emphasis on Computational Biology and Bioinformatics. *WIREs Data Mining and Knowledge Discovery* 2, 6 (2012), 493–507. https://doi.org/10.1002/widm.1072

[34] G. E. P. Box and D. R. Cox. 1964. An Analysis of Transformations. *Journal of the Royal Statistical Society. Series B (Methodological)* 26, 2 (1964), 211–252. http://www.jstor.org.proxy.library.uu.nl/stable/2984418

[35] George E. P. Box and Gwilym M. Jenkins. 1970. *Time Series Analysis; Forecasting and Control.* Holden-Day, San Francisco.

[36] George E. P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. 2008. *Time Series Analysis.* Wiley, Hoboken, NJ, USA. https://doi.org/10.1002/9781118619193

[37] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. 2015. *Time Series Analysis: Forecasting and Control.* John Wiley & Sons, Hoboken, NJ, USA.

[38] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 2017. *Classification And Regression Trees.* Routledge. https://doi.org/10.1201/9781315139470

[39] Eoin Brophy, Zhengwei Wang, Qi She, and Tomás Ward. 2023. Generative Adversarial Networks in Time Series: A Systematic Literature Review. *ACM Comput. Surv.* 55, 10 (2023), 1–31. https://doi.org/10.1145/3559540

[40] James A. Cadzow, Behshad Baseghi, and Tony Hsu. 1983. Singular-Value Decomposition Approach to Time Series Modelling. *IEE Proceedings F Communications, Radar and Signal Processing* 130, 3 (1983), 202. https://doi.org/10.1049/ip-f-1.1983.0034

[41] Borui Cai, Shuiqiao Yang, Longxiang Gao, and Yong Xiang. 2023. Hybrid Variational Autoencoder for Time Series Forecasting. *Knowledge-Based Systems* 281 (2023), 111079. https://doi.org/10.1016/j.knosys.2023.111079

[42] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. BRITS: Bidirectional Recurrent Imputation for Time Series. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/file/734e6bfcd358e25ac1db0a4241b95651-Paper.pdf

[43] Vincenzo Capasso and David Bakstein. 2021. *An Introduction to Continuous-Time Stochastic Processes.* Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-69653-5

[44] Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, and Asdrubal Lopez. 2020. A Comprehensive Survey on Support Vector Machine Classification: Applications, Challenges and Trends. *Neurocomputing* 408 (2020), 189–215. https://doi.org/10.1016/j.neucom.2019.10.118

[45] Kaushik Chakrabarti, Eamonn Keogh, Sharad Mehrotra, and Michael Pazzani. 2002. Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. *ACM Trans. Database Syst.* 27, 2 (2002), 188–228. https://doi.org/10.1145/568518.568520

[46] Chandralika Chakraborty and P. H. Talukdar. 2016. Issues and Limitations of HMM in Speech Processing: A Survey. *International Journal of Computer Applications* 141, 7 (2016), 13–17. https://doi.org/10.5120/ijca2016909693

[47] Mayank Arya Chandra and S. S. Bedi. 2021. Survey on SVM and Their Application in Image Classification. *International Journal of Information Technology* 13, 5 (2021), 1–11. https://doi.org/10.1007/s41870-017-0080-1

[48] Chris Chatfield. 2003. *The Analysis of Time Series.* Chapman and Hall/CRC. https://doi.org/10.4324/9780203491683

[49] Georgios Chatzigeorgakidis, Skoutas Dimitrios, Kostas Patroumpas, Themis Palpanas, Spiros Athanasiou, and Spiros Skiadopoulos. 2021. Twin Subsequence Search in Time Series. *CoRR* abs/2104.06874 (2021).

[50] Ningjiang Chen, Huan Tu, Xiaoyan Duan, Liangqing Hu, and Chengxiang Guo. 2022. Semisupervised Anomaly Detection of Multivariate Time Series Based on a Variational Autoencoder. *Applied Intelligence* (2022). https://doi.org/10.1007/s10489-022-03829-1

[51] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[52] Zonglei Chen, Minbo Ma, Tianrui Li, Hongjun Wang, and Chongshou Li. 2023. Long Sequence Time-Series Forecasting With Deep Learning: A Survey. *Information Fusion* 97 (2023), 101819. https://doi.org/10.1016/j.inffus.2023.101819

[53] Giacomo Chiarot and Claudio Silvestri. 2023. Time Series Compression Survey. *ACM Comput. Surv.* 55, 10 (2023), 1–32. https://doi.org/10.1145/3560814

[54] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. https://doi.org/10.48550/arXiv.1406.1078

[55] Fu Lai Korris Chung, T. C. Fu, Wing Pong Robert Luk, and Vincent To Yee Ng. 2001. Flexible Time Series Pattern Matching Based on Perceptually Important Points. In *Workshop on Learning from Temporal and Spatial Data in International Joint Conference on Artificial Intelligence.* Conference presentation (not published in journal/proceeding/book), 1–7.

[56] Tak chung Fu. 2011. A Review on Time Series Data Mining. *Engineering Applications of Artificial Intelligence* 24, 1 (2011), 164–181. https://doi.org/10.1016/j.engappai.2010.09.007

[57] William S. Cleveland. 1972. The Inverse Autocorrelations of a Time Series and Their Applications. *Technometrics* 14, 2 (1972), 277–293. https://doi.org/10.1080/00401706.1972.10488914

[58] Andy Coenen and Adam Pearce. 2019. Understanding UMAP. https://pair-code.github.io/understanding-umap/

[59] Aryana Collins Jackson and Seán Lacey. 2020. The Discrete Fourier Transformation for Seasonality and Anomaly Detection of an Application to Rare Data. *Data Technologies and Applications* 54, 2 (2020), 121–132. https://doi.org/10.1108/DTA-12-2019-0243

[60] Andrew A. Cook, Goksel Misirli, and Zhong Fan. 2020. Anomaly Detection for IoT Time-Series Data: A Survey. *IEEE Internet of Things Journal* 7, 7 (2020), 6481–6494. https://doi.org/10.1109/JIOT.2019.2958185

[61] James W. Cooley and John W. Tukey. 1965. An Algorithm for The Machine Calculation of Complex Fourier Series. *Math. Comp.* 19, 90 (1965), 297–301. https://doi.org/10.1090/S0025-5718-1965-0178586-1

[62] Abigail Copiaco, Yassine Himeur, Abbes Amira, Wathiq Mansoor, Fodil Fadli, Shadi Atalla, and Shahab Saquib Sohail. 2023. An Innovative Deep Anomaly Detection of Building Energy Consumption Using Energy Time-Series Images. *Engineering Applications of Artificial Intelligence* 119

(2023), 105775. https://doi.org/10.1016/j.engappai.2022.105775

[63] M. Correll and M. Gleicher. 2016. The Semantics of Sketch: Flexibility in Visual Query Systems for Time Series Data. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 131–140. https://doi.org/10.1109/VAST.2016.7883519

[64] Hamid Dadkhahi, Marco F. Duarte, and Benjamin Marlin. 2015. Isomap Out-Of-Sample Extension for Noisy Time Series Data. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6. https://doi.org/10.1109/MLSP.2015.7324314

[65] Uender Barbosa de Souza, João Paulo Lemos Escola, and Leonardo Cunha Da Brito. 2022. A Survey on Hilbert-Huang Transform: Evolution, Challenges and Solutions. *Digital Signal Processing* 120 (2022), 103292. https://doi.org/10.1016/j.dsp.2021.103292

[66] Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. 2017. A Review on Time Series Forecasting Techniques for Building Energy Consumption. *Renewable and Sustainable Energy Reviews* 74 (2017), 902–924. https://doi.org/10.1016/j.rser.2017.02.085

[67] Johan Debayle, Nima Hatami, and Yann Gavet. 2018. Classification of Time-Series Images Using Deep Convolutional Neural Networks. In *Tenth International Conference on Machine Vision (ICMV 2017)*, Jianhong Zhou, Petia Radeva, Dmitry Nikolaev, and Antanas Verikas (Eds.). SPIE, 23. https://doi.org/10.1117/12.2309486

[68] Lokenath Debnath and Dambaru Bhatta. 2016. *Integral Transforms and Their Applications*. Chapman and Hall/CRC. https://doi.org/10.1201/9781420010916

[69] Manfred Deistler and Wolfgang Scherrer. 2022. *Time Series Models*. Vol. 224. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-031-13213-1

[70] Liliya Anatolievna Demidova and Maksim Anatolievich Stepanov. 2020. Approach to the Analysis of the Multidimensional Time Series Based on the UMAP Algorithm in the Problems of the Complex Systems Proactive Maintenance. In *2020 International Conference on Information Technologies (InfoTech)*. IEEE, 1–4. https://doi.org/10.1109/InfoTech49733.2020.9211008

[71] Qi Deng and Dirk Soffker. 2022. A Review of HMM-Based Approaches of Driving Behaviors Recognition and Prediction. *IEEE Transactions on Intelligent Vehicles* 7, 1 (2022), 21–31. https://doi.org/10.1109/TIV.2021.3065933

[72] Volker L. Deringer, Albert P. Bartók, Noam Bernstein, David M. Wilkins, Michele Ceriotti, and Gábor Csányi. 2021. Gaussian Process Regression for Materials and Molecules. *Chemical reviews* 121, 16 (2021), 10073–10141. https://doi.org/10.1021/acs.chemrev.1c00022

[73] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. 2021. TimeVAE: A Variational Auto-Encoder for Multivariate Time Series Generation. http://arxiv.org/pdf/2111.08095

[74] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. 2008. Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. *Proc. VLDB Endow.* 1, 2 (2008), 1542–1552. https://doi.org/10.14778/1454159.1454226

[75] Paul Doukhan. 2018. *Stochastic Models for Time Series*. Vol. 80. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-76938-7

[76] Xiaoping Du, Lelai Deng, and Kun Qian. 2018. Current Market Top Business Scopes Trend—A Concurrent Text and Time Series Active Learning Study of NASDAQ and NYSE Stocks from 2012 to 2017. *Applied Sciences* 8, 5 (2018). https://doi.org/10.3390/app8050751

[77] Pierpaolo D'Urso and Elizabeth Ann Maharaj. 2012. Wavelets-Based Clustering of Multivariate Time Series. *Fuzzy Sets and Systems* 193 (2012), 33–61. https://doi.org/10.1016/j.fss.2011.10.002

[78] Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. TSMixer: Lightweight MLP-Mixer Model for Multivariate Time Series Forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Ambuj Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (Eds.). ACM, New York, NY, USA, 459–469. https://doi.org/10.1145/3580305.3599533

[79] Robert F. Engle. 1982. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica* 50, 4 (1982), 987. https://doi.org/10.2307/1912773

[80] Philippe Esling and Carlos Agon. 2012. Time-Series Data Mining. *ACM Comput. Surv.* 45, 1 (2012). https://doi.org/10.1145/2379776.2379788

[81] Philippe Esling and Carlos Agon. 2013. Multiobjective Time Series Matching for Audio Classification and Retrieval. *IEEE Transactions on Audio, Speech, and Language Processing* 21, 10 (2013), 2057–2072. https://doi.org/10.1109/TASL.2013.2265086

[82] Mateus Espadoto, Gabriel Appleby, Ashley Suh, Dylan Cashman, Mingwei Li, Carlos Scheidegger, Erik W. Anderson, Remco Chang, and Alexandru C. Telea. 2023. UnProjection: Leveraging Inverse-Projections for Visual Analytics of High-Dimensional Data. *IEEE Transactions on Visualization and Computer Graphics* 29, 2 (2023), 1559–1572. https://doi.org/10.1109/TVCG.2021.3125576

[83] Mateus Espadoto, Rafael M. Martins, Andreas Kerren, Nina S. T. Hirata, and Alexandru C. Telea. 2021. Toward a Quantitative Survey of Dimension Reduction Techniques. *IEEE Transactions on Visualization and Computer Graphics* 27, 3 (2021), 2153–2173. https://doi.org/10.1109/TVCG.2019.2944182

[84] Amin Fakhrazari and Hamid Vakilzadian. 2017. A Survey on Time Series Data Mining. In *2017 IEEE International Conference on Electro Information Technology (EIT)*. 476–481. https://doi.org/10.1109/EIT.2017.8053409

[85] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. 1994. Fast Subsequence Matching in Time-Series Databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data (SIGMOD '94)*. Association for Computing Machinery, New York, NY, USA, 419–429. https://doi.org/10.1145/191839.191925

[86] Jean M. Firth. 1992. *Discrete Transforms*. Springer Netherlands, Dordrecht. https://doi.org/10.1007/978-94-011-2358-7

[87] R. A. Fisher. 1936. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* 7, 2 (1936), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

[88] Christian Francq and Jean-Michel Zakoian. 2019. *GARCH models: Structure, Statistical Inference and Financial Applications* (2 edition ed.). John Wiley & Sons, Hoboken NJ.

[89] Yoav Freund and Robert E. Schapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. System Sci.* 55, 1 (1997), 119–139. https://doi.org/10.1006/jcss.1997.1504

[90] Tak-chung Fu, Fu-lai Chung, Robert Luk, and Chak-man Ng. 2005. Preventing Meaningless Stock Time Series Pattern Discovery by Changing Perceptually Important Point Detection. In *Fuzzy Systems and Knowledge Discovery*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Lipo Wang, and Yaochu Jin (Eds.). Lecture Notes in Computer Science, Vol. 3613. Springer Berlin Heidelberg, Berlin, Heidelberg, 1171–1174. https://doi.org/10.1007/11539506{_}146

[91] Tak-chung Fu, Fu-lai Chung, Robert Luk, and Chak-man Ng. 2007. Stock Time Series Pattern Matching: Template-based vs. Rule-based Approaches. *Engineering Applications of Artificial Intelligence* 20, 3 (2007), 347–364. https://doi.org/10.1016/j.engappai.2006.07.003

[92] Tak-chung Fu, Fu-lai Chung, Vincent Ng, and Robert Luk. 2001. Pattern Discovery From Stock Time Series Using Self-Organizing Maps. In *Workshop notes of KDD2001 workshop on temporal data mining*, Vol. 1.

[93] Jianhua Gao, Weixing Ji, Lulu Zhang, Senhao Shao, Yizhuo Wang, and Feng Shi. 2020. Fast Piecewise Polynomial Fitting of Time-Series Data for Streaming Computing. *IEEE Access* 8 (2020), 43764–43775. https://doi.org/10.1109/ACCESS.2020.2976494

[94] Yongxiang Gao, Zhi Zhao, Yimin Chen, Gehendra Mahara, Jialing Huang, Zhuochen Lin, and Jinxin Zhang. 2020. Automatic Epileptic Seizure Classification in Multichannel EEG Time Series With Linear Discriminant Analysis. *Technology and health care : official journal of the European Society for Engineering and Medicine* 28, 1 (2020), 23–33. https://doi.org/10.3233/THC-181548

[95] Aurélien Géron. 2019. *Hands-on Machine Learning With Scikit-Learn, Keras, and Tensorflow: Concepts, tools, and techniques to build intelligent systems* (second edition ed.). O'Reilly and ProQuest Ebook Central, Sebastopol, CA and Ann Arbor. https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=5892320

[96] Ebrahim Ghaderpour, Spiros D. Pagiatakis, and Quazi K. Hassan. 2021. A Survey on Change Detection and Time Series Analysis with Applications. *Applied Sciences* 11, 13 (2021). https://doi.org/10.3390/app11136141

[97] Michael Gleicher. 2013. Explainers: Expert Explorations With Crafted Projections. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2042–2051. https://doi.org/10.1109/TVCG.2013.157

[98] Nina Golyandina, Anton Korobeynikov, and Anatoly Zhigljavsky. 2018. *Singular Spectrum Analysis with R.* Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-57380-8

[99] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative Adversarial Networks. *Commun. ACM* 63, 11 (2020), 139–144. https://doi.org/10.1145/3422622

[100] D. Gunopulos and G. Das. 2001. Time Series Similarity Measures and Time Series Indexing. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*. New York, 624. https://doi.org/10.1145/375663.375808

[101] Tiantian Guo, Tongpo Zhang, Enggee Lim, Miguel Lopez-Benitez, Fei Ma, and Limin Yu. 2022. A Review of Wavelet Analysis and Its Applications: Challenges and Opportunities. *IEEE Access* 10 (2022), 58869–58903. https://doi.org/10.1109/ACCESS.2022.3179517

[102] Aliakbar Montazer Haghighi, Abburi Anil Kumar, and Dimitar P. Mishev. 2024. *Higher Mathematics for Science and Engineering.* Springer Nature Singapore, Singapore. https://doi.org/10.1007/978-981-99-5431-5

[103] James D. Hamilton. 1989. A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica* 57, 2 (1989), 357. https://doi.org/10.2307/1912559

[104] James Douglas Hamilton. 2020. *Time Series Analysis.* Princeton University Press, Princeton, NJ. https://doi.org/10.1515/9780691218632

[105] Lu Han, Han-Jia Ye, and De-Chuan Zhan. 2024. The Capacity and Robustness Trade-Off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting. *IEEE Transactions on Knowledge and Data Engineering* 36, 11 (2024), 7129–7142. https://doi.org/10.1109/TKDE.2024.3400008

[106] B. Hariharan, R. Siva, S. Sadagopan, Vaibhav Mishra, and Yash Raghav. 2023. Malware Detection Using XGBoost based Machine Learning Models - Review. In *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*. IEEE, 964–970. https://doi.org/10.1109/ICECAA58104.2023.10212327

[107] Andrew C. Harvey and Neil Shephard. 1993. 10 Structural Time Series Models. In *Econometrics*. Handbook of Statistics, Vol. 11. Elsevier, 261–302. https://doi.org/10.1016/S0169-7161(05)80045-8

[108] Hossein Hassani and Dimitrios Thomakos. 2010. A Review on Singular Spectrum Analysis for Economic and Financial Time Series. *Statistics and Its Interface* 3, 3 (2010), 377–397. https://doi.org/10.4310/SII.2010.v3.n3.a11

[109] Hui He, Qi Zhang, Kun Yi, Kaize Shi, Zhendong Niu, and Longbing Cao. 2024. Distributional Drift Adaptation With Temporal Conditional Variational Autoencoder for Multivariate Time Series Forecasting. *IEEE Transactions on Neural Networks and Learning Systems* PP (2024). https://doi.org/10.1109/TNNLS.2024.3384842

[110] Zhiwei He, Jingping Chen, and Mingyu Gao. 2018. Feature Time Series Clustering for Lithium Battery Based on SOM Neural Network. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 358–363. https://doi.org/10.1109/ICIEA.2018.8397742

[111] Geoffrey E. Hinton and Sam Roweis. 2002. Stochastic Neighbor Embedding. *Advances in Neural Information Processing Systems* 15 (2002).

[112] Tin Kam Ho. 1995. Random Decision Forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*. IEEE Comput. Soc. Press, 278–282. https://doi.org/10.1109/ICDAR.1995.598994

[113] S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[114] Christopher Holder, Matthew Middlehurst, and Anthony Bagnall. 2024. A Review and Evaluation of Elastic Distance Functions for Time Series Clustering. *Knowledge and Information Systems* 66, 2 (2024), 765–809. https://doi.org/10.1007/s10115-023-01952-0

[115] Tao Huamin, Deng Qiuqun, and Xiao Shanzhu. 2020. Reconstruction of Time Series With Missing Value Using 2D Representation-Based Denoising Autoencoder. *Journal of Systems Engineering and Electronics* 31, 6 (2020), 1087–1096. https://doi.org/10.23919/JSEE.2020.000081

[116] Norden E. Huang and Nii O. Attoh-Okine. 2005. *The Hilbert-Huang Transform in Engineering*. CRC Press. https://doi.org/10.1201/9781420027532

[117] Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H. Liu. 1998. The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-stationary Time Series Analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences* 454, 1971 (1998), 903–995.

[118] Norden E. Huang and Zhaohua Wu. 2008. A Review on Hilbert–Huang Transform: Method and Its Applications to Geophysical Studies. *Reviews of Geophysics* 46, 2 (2008). https://doi.org/10.1029/2007RG000228

[119] Kazuki Irie, Zoltán Tüske, Tamer Alkhouli, Ralf Schlüter, and Hermann Ney. 2016. LSTM, GRU, Highway and a Bit of Attention: An Empirical Overview for Language Modeling in Speech Recognition. In *Interspeech 2016*. ISCA, ISCA, 3519–3523. https://doi.org/10.21437/Interspeech.2016-491

[120] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep Learning for Time Series Classification: A Review. *Data Mining and Knowledge Discovery* 33, 4 (2019), 917–963. https://doi.org/10.1007/s10618-019-00619-1

[121] Dominik Jäckle, Fabian Fischer, Tobias Schreck, and Daniel A. Keim. 2016. Temporal MDS Plots for Analysis of Multivariate Data. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 141–150. https://doi.org/10.1109/TVCG.2015.2467553

[122] Jinsung Jeon, JEONGHAK KIM, Haryong Song, Seunghyeon Cho, and Noseong Park. 2022. GT-GAN: General Purpose Time Series Synthesis with Generative Adversarial Networks. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc, 36999–37010. https://proceedings.neurips.cc/paper_files/paper/2022/file/f03ce573aa8bce26f77b76f1cb9ee979-Paper-Conference.pdf

[123] Abdul J. Jerri. 1992. *Integral and Discrete Transforms With Applications and Error Analysis*. Monographs and textbooks in pure and applied mathematics, Vol. 162. M. Dekker, New York.

[124] Cun Ji, Xiunan Zou, Yupeng Hu, Shijun Liu, Lei Lyu, and Xiangwei Zheng. 2019. XG-SF: An XGBoost Classifier Based on Shapelet Features for Time Series Classification. *Procedia Computer Science* 147 (2019), 24–28. https://doi.org/10.1016/j.procs.2019.01.179

[125] Hui Jiang. 2010. Discriminative Training of HMMs for Automatic Speech Recognition: A Survey. *Computer Speech & Language* 24, 4 (2010), 589–608. https://doi.org/10.1016/j.csl.2009.08.002

[126] Ammar Ismael Kadhim. 2019. Survey on Supervised Machine Learning Techniques for Automatic Text Classification. *Artificial Intelligence Review* 52, 1 (2019), 273–292. https://doi.org/10.1007/s10462-018-09677-1

[127] T. Kahveci, A. Singh, and A. Gurel. 2002. An Efficient Index Structure for Shift and Scale Invariant Search of Mufti-Attribute Time Sequences. In *Proceedings 18th International Conference on Data Engineering*. IEEE Comput. Soc, 266. https://doi.org/10.1109/ICDE.2002.994720

[128] Amina Kammoun, Rim Slama, Hedi Tabia, Tarek Ouni, and Mohmed Abid. 2023. Generative Adversarial Networks for Face Generation: A Survey. *ACM Comput. Surv.* 55, 5 (2023), 1–37. https://doi.org/10.1145/3527850

[129] Argyro Kampouraki, George Manis, and Christophoros Nikou. 2009. Heartbeat Time Series Classification With Support Vector Machines. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society* 13, 4 (2009), 512–518. https://doi.org/10.1109/TITB.2008.2003323

[130] Aminata Kane and Nematollaah Shiri. 2017. Multivariate Time Series Representation and Similarity Search Using PCA. In *Advances in Data Mining. Applications and Theoretical Aspects*, Petra Perner (Ed.). Lecture Notes in Computer Science, Vol. 10357. Springer International Publishing and Imprint: Springer, Cham, 122–136. https://doi.org/10.1007/978-3-319-62701-4{_}10

[131] F.R.S. Karl Pearson. 1901. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11 (1901), 559–572. https://doi.org/10.1080/14786440109462720

[132] K. Kashino, G. Smith, and H. Murase. 1999. Time-Series Active Search for Quick Retrieval of Audio and Video. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*. IEEE, 2993–2996 vol.6. https://doi.org/10.1109/ICASSP.1999.757470

[133] Jatinder Kaur, Kulwinder Singh Parmar, and Sarbjit Singh. 2023. Autoregressive Models in Environmental Forecasting Time Series: A Theoretical and Application Review. *Environmental science and pollution research international* 30, 8 (2023), 19617–19641. https://doi.org/10.1007/s11356-023-25148-9

[134] Sangramsing Kayte, Monica Mundada, and Jayesh Gujrathi. 2015. Hidden Markov Model based Speech Synthesis: A Review. *International Journal of Computer Applications* 130, 3 (2015), 35–39. https://doi.org/10.5120/ijca2015906965

[135] E. Keogh. 1997. Fast Similarity Search in the Presence of Longitudinal Scaling in Time Series Databases. In *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence*. 578–584. https://doi.org/10.1109/TAI.1997.632306

[136] Eamonn Keogh. 2008. Indexing and Mining Time Series Data. In *Encyclopedia of GIS*, Shashi Shekhar and Hui Xiong (Eds.). Springer US, Boston, MA, 493–497. https://doi.org/10.1007/978-0-387-35973-1_598

[137] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowledge and Information Systems* 3, 3 (2001), 263–286. https://doi.org/10.1007/PL00011669

[138] Eamonn Keogh, Li Wei, Xiaopeng Xi, Sang-Hee Lee, and Michail Vlachos. 2006. Lb_keogh Supports Exact Indexing of Shapes Under Rotation Invariance With Arbitrary Representations and Distance Measures. In *Proceedings of the 32nd international conference on Very large data bases*. 882–893.

[139] Eamonn J. Keogh and Michael J. Pazzani. 2000. A Simple Dimensionality Reduction Technique for Fast Similarity Search in Large Time Series Databases. In *Knowledge Discovery and Data Mining. Current Issues and New Applications*, Takao Terano, Huan Liu, and Arbee L. P. Chen (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 122–133.

[140] Mehak Khan, Hongzhi Wang, Adnan Riaz, Aya Elfatyany, and Sajida Karim. 2021. Bidirectional LSTM-RNN-Based Hybrid Deep Learning Frameworks for Univariate Time Series Classification. *The Journal of Supercomputing* 77, 7 (2021), 7021–7045. https://doi.org/10.1007/s11227-020-03560-z

[141] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in Vision: A Survey. *ACM Comput. Surv.* 54, 10s (2022), 1–41. https://doi.org/10.1145/3505244

[142] Abdolrahman Khoshrou, Andre B. Dorsman, and Eric J. Pauwels. 2017. SVD-Based Visualisation and Approximation for Time Series Data in Smart Energy Systems. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, 1–6. https://doi.org/10.1109/ISGTEurope.2017.8260303

[143] Irena Koprinska, Dengsong Wu, and Zheng Wang. 2018. Convolutional Neural Networks for Energy Time Series Forecasting. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8. https://doi.org/10.1109/IJCNN.2018.8489399

[144] John R. Koza, Forrest H. Bennett, David Andre, and Martin A. Keane. 1996. Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In *Artificial Intelligence in Design '96*, John S. Gero and Fay Sudweeks (Eds.). Springer Netherlands, Dordrecht, 151–170. https://doi.org/10.1007/978-94-009-0279-4{_}9

[145] Mark A. Kramer. 1991. Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. *AIChE Journal* 37, 2 (1991), 233–243. https://doi.org/10.1002/aic.690370209

[146] Raghavendra Kumar, Pardeep Kumar, and Yugal Kumar. 2020. Time Series Data Prediction using IoT and Machine Learning Technique. *Procedia Computer Science* 167 (2020), 373–381. https://doi.org/10.1016/j.procs.2020.03.240

[147] Martin Längkvist, Lars Karlsson, and Amy Loutfi. 2014. A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modeling. *Pattern Recognition Letters* 42 (2014), 11–24. https://doi.org/10.1016/j.patrec.2014.01.008

[148] Fritz Lekschas, Brant Peterson, Daniel Haehn, Eric Ma, Nils Gehlenborg, and Hanspeter Pfister. 2020. PEAX: Interactive Visual Pattern Search in Sequential Data Using Unsupervised Deep Representation Learning. *Computer Graphics Forum* 39, 3 (2020), 167–179. https://doi.org/10.1111/cgf.13971

[149] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*, Igor V. Tetko, Věra Kůrková, Pavel Karpov, and Fabian Theis (Eds.). Lecture Notes in Computer Science, Vol. 11730. Springer International Publishing, Cham, 703–716. https://doi.org/10.1007/978-3-030-30490-4{_}56

[150] Gen Li and Jason J. Jung. 2023. Deep Learning for Anomaly Detection in Multivariate Time Series: Approaches, Applications, and Challenges. *Information Fusion* 91 (2023), 93–102. https://doi.org/10.1016/j.inffus.2022.10.008

[151] Hailin Li. 2016. Accurate and Efficient Classification Based on Common Principal Components Analysis for Multivariate Time Series. *Neurocomputing* 171 (2016), 744–753. https://doi.org/10.1016/j.neucom.2015.07.010

[152] Hailin Li. 2019. Multivariate Time Series Clustering Based on Common Principal Component Analysis. *Neurocomputing* 349 (2019), 239–247. https://doi.org/10.1016/j.neucom.2019.03.060

[153] Hongming Li, Shujian Yu, and Jose Principe. 2023. Causal Recurrent Variational Autoencoder for Medical Time Series Generation. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 7 (2023), 8562–8570. https://doi.org/10.1609/aaai.v37i7.26031

[154] Longyuan Li, Junchi Yan, Haiyang Wang, and Yaohui Jin. 2021. Anomaly Detection of Time Series With Smoothness-Inducing Sequential Variational Auto-Encoder. *IEEE Transactions on Neural Networks and Learning Systems* 32, 3 (2021), 1177–1191. https://doi.org/10.1109/TNNLS.2020.2980749

[155] Lifen Li and Tianyu Zhang. 2021. Research on Text Generation Based on LSTM. *International Core Journal of Engineering* 7, 5 (2021), 525–535. https://doi.org/0067

[156] Xue Li, Hantao Cui, Tao Jiang, Yao Xu, Hongjie Jia, and Fangxing Li. 2018. Multichannel Continuous Wavelet Transform Approach to Estimate Electromechanical Oscillation Modes, Mode Shapes and Coherent Groups from Synchrophasors in Bulk Power Grids. *International Journal of Electrical Power & Energy Systems* 96 (2018), 222–237. https://doi.org/10.1016/j.ijepes.2017.09.043

[157] Zheng Li, Chao Ma, Xiaochuan Shi, Dian Zhang, Wei Li, and Libing Wu. 2021. TSA-GAN: A Robust Generative Adversarial Networks for Time Series Augmentation. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534001

[158] T. Warren Liao. 2005. Clustering of Time Series Data - A Survey. *Pattern Recognition* 38, 11 (2005), 1857–1874. https://doi.org/10.1016/j.patcog.2005.01.025

[159] Willone Lim, Kelvin Sheng Chek Yong, Bee Theng Lau, and Colin Choon Lin Tan. 2024. Future of Generative Adversarial Networks (GAN) for Anomaly Detection in Network Security: A Review. *Computers & Security* 139 (2024), 103733. https://doi.org/10.1016/j.cose.2024.103733

[160] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. 2003. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In *Proc. of 8th ACM SIGMOD Workshop on Research Issues in DM and Knowledge Discovery (DMKD '03)*. Assoc. for Comp. Mach., New York, NY, USA, 2–11. https://doi.org/10.1145/882082.882086

[161] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. 2007. Experiencing Sax: A Novel Symbolic Representation of Time Series. *Data Mining and knowledge discovery* 15, 2 (2007), 107–144.

[162] Chien-Liang Liu, Wen-Hoar Hsaio, and Yao-Chung Tu. 2019. Time Series Classification With Multivariate Convolutional Neural Network. *IEEE Transactions on Industrial Electronics* 66, 6 (2019), 4788–4797. https://doi.org/10.1109/TIE.2018.2864702

[163] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. 2024. A Survey of Visual Transformers. *IEEE Transactions on Neural Networks and Learning Systems* 35, 6 (2024), 7478–7498. https://doi.org/10.1109/TNNLS.2022.3227717

[164] António M. Lopes and J. A. Tenreiro Machado. 2014. Analysis of Temperature Time-Series: Embedding Dynamics Into the MDS Method. *Communications in Nonlinear Science and Numerical Simulation* 19, 4 (2014), 851–871. https://doi.org/10.1016/j.cnsns.2013.08.031

[165] Miodrag Lovrić, Marina Milanović, and Milan Stamenković. 2014. Algoritmic Methods for Segmentation of Time Series: An Overview. *Journal of Contemporary Economic and Business Issues* 1, 1 (2014), 31–53. https://hdl.handle.net/10419/147468

[166] Yonghong Luo, Xiangrui Cai, Ying ZHANG, Jun Xu, and Yuan xiaojie. 2018. Multivariate Time Series Imputation with Generative Adversarial Networks. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/file/96b9bff013acedfb1d140579e2fbeb63-Paper.pdf

[167] Helmut Lütkepohl. 1991. *Introduction to Multiple Time Series Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-02691-5

[168] Helmut Lütkepohl. 2005. *New Introduction to Multiple Time Series Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-27752-1

[169] Cai-Xia Lv, Shu-Yi An, Bao-Jun Qiao, and Wei Wu. 2021. Time Series Analysis of Hemorrhagic Fever With Renal Syndrome in Mainland China by Using an Xgboost Forecasting Model. *BMC infectious diseases* 21, 1 (2021), 839. https://doi.org/10.1186/s12879-021-06503-y

[170] Elizabeth Ann Maharaj and Andrés M. Alonso. 2014. Discriminant Analysis of Multivariate Time Series: Application to Diagnosis Based on ECG Signals. *Computational Statistics & Data Analysis* 70 (2014), 67–87. https://doi.org/10.1016/j.csda.2013.09.006

[171] S. G. Mallat. 1989. Multifrequency Channel Decompositions of Images and Wavelet Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37, 12 (1989), 2091–2110. https://doi.org/10.1109/29.45554

[172] S. G. Mallat. 1989. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 7 (1989), 674–693. https://doi.org/10.1109/34.192463

[173] Peter Mannfolk, Ronnie Wirestam, Markus Nilsson, Freddy Ståhlberg, and Johan Olsrud. 2010. Dimensionality Reduction of fMRI Time Series Data Using Locally Linear Embedding. *Magma (New York, N.Y.)* 23, 5-6 (2010), 327–338. https://doi.org/10.1007/s10334-010-0204-0

[174] Ricardo P. Masini, Marcelo C. Medeiros, and Eduardo F. Mendes. 2023. Machine Learning Advances for Time Series Forecasting. *Journal of Economic Surveys* 37, 1 (2023), 76–111. https://doi.org/10.1111/joes.12429

[175] William Menke and Roger Creel. 2021. Gaussian Process Regression Reviewed in the Context of Inverse Theory. *Surveys in Geophysics* 42, 3 (2021), 473–503. https://doi.org/10.1007/s10712-021-09640-w

[176] Ryan Mercer and Eamonn Keogh. 2022. Matrix Profile XXV: Introducing Novelets: A Primitive that Allows Online Detection of Emerging Behaviors in Time Series. In *2022 IEEE International Conference on Data Mining (ICDM)*. 338–347. https://doi.org/10.1109/ICDM54844.2022.00044

[177] Alfred Mertins. 2023. *Signaltheorie*. Springer Fachmedien Wiesbaden, Wiesbaden. https://doi.org/10.1007/978-3-658-41529-7

[178] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. 2024. Bake off Redux: A Review and Experimental Evaluation of Recent Time Series Classification Algorithms. *Data Mining and Knowledge Discovery* (2024). https://doi.org/10.1007/s10618-024-01022-1

[179] Navid Mohammadi Foumani, Lynn Miller, Chang Wei Tan, Geoffrey I. Webb, Germain Forestier, and Mahsa Salehi. 2024. Deep Learning for Time Series Classification and Extrinsic Regression: A Current Survey. *ACM Comput. Surv.* 56, 9 (2024), 1–45. https://doi.org/10.1145/3649448

[180] M. A. Moni and A. B. M. Shawkat Ali. 2009. HMM Based Hand Gesture Recognition: A Review on Techniques and Approaches. In *2009 2nd IEEE International Conference on Computer Science and Information Technology*. IEEE, 433–437. https://doi.org/10.1109/ICCSIT.2009.5234536

[181] Raha Moraffah, Paras Sheth, Mansooreh Karami, Anchit Bhattacharya, Qianru Wang, Anique Tahir, Adrienne Raglin, and Huan Liu. 2021. Causal Inference for Time Series Analysis: Problems, Methods and Evaluation. *Knowledge and Information Systems* 63, 12 (2021), 3041–3085. https://doi.org/10.1007/s10115-021-01621-0

[182] P. A. Moran and Peter Whittle. 1951. Hypothesis Testing in Time Series Analysis. *Journal of the Royal Statistical Society. Series A (General)* 114, 4 (1951), 579. https://doi.org/10.2307/2981095

[183] J. Morlet. 1983. Sampling Theory and Wave Propagation. In *Issues in Acoustic Signal — Image Processing and Recognition*, C. H. Chen (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 233–261. https://doi.org/10.1007/978-3-642-82002-1{_}12

[184] Kevin P. Murphy. 2013. *Machine Learning: A Probabilistic Perspective* (4. print. (fixed many typos) ed.). MIT Press, Cambridge, Mass.

[185] Randa Natras, Benedikt Soja, and Michael Schmidt. 2022. Ensemble Machine Learning of Random Forest, AdaBoost and XGBoost for Vertical Total Electron Content Forecasting. *Remote Sensing* 14, 15 (2022), 3547. https://doi.org/10.3390/rs14153547

[186] Nam Nguyen and Brian Quanz. 2021. Temporal Latent Auto-Encoder: A Method for Probabilistic Multivariate Time Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 10 (2021), 9117–9125. https://doi.org/10.1609/aaai.v35i10.17101

[187] Majid Niazkar, Andrea Menapace, Bruno Brentan, Reza Piraei, David Jimenez, Pranav Dhawan, and Maurizio Righetti. 2024. Applications of XGBoost in Water Resources Engineering: A Systematic Literature Review. *Environmental Modelling & Software* 174 (2024), 105971. https://doi.org/10.1016/j.envsoft.2024.105971

[188] Laukika Nilangekar, Dnyaneshwari Popat Funde, Vaishnavi Kshatri, and Jalindar Gandal. 2023. XG Boost Algorithm for Fraudulent Vishing Detection: A Review Literature. *Vidhyayana - An International Multidisciplinary Peer-Reviewed E-Journal - ISSN 2454-8596* 8, si7 (2023), 61–78. https://www.vidhyayanaejournal.org/journal/article/view/809

[189] Mehrbakhsh Nilashi, Rabab Ali Abumalloh, Hossein Ahmadi, Sarminah Samad, Abdullah Alghamdi, Mesfer Alrizq, Sultan Alyami, and Fatima Khan Nayer. 2023. Electroencephalography (EEG) Eye State Classification Using Learning Vector Quantization and Bagged Trees. *Heliyon* 9, 4 (2023). https://doi.org/10.1016/j.heliyon.2023.e15258

[190] Umit Y. Ogras and Hakan Ferhatosmanoglu. 2006. Online Summarization of Dynamic Time Series Data. *The VLDB Journal* 15, 1 (2006), 84–98. https://doi.org/10.1007/s00778-004-0149-x

[191] Arun Pandey, Hannes de Meulemeester, Bart de Moor, and Johan A.K. Suykens. 2023. Multi-View Kernel PCA for Time Series Forecasting. *Neurocomputing* 554 (2023), 126639. https://doi.org/10.1016/j.neucom.2023.126639

[192] Baidyanath Patra. 2018. *An Introduction to Integral Transforms*. CRC Press. https://doi.org/10.1201/9780429503580

[193] Narendra Patwardhan, Stefano Marrone, and Carlo Sansone. 2023. Transformers in the Real World: A Survey on NLP Applications. *Information* 14, 4 (2023), 242. https://doi.org/10.3390/info14040242

[194] Bohdan Pavlyshenko. 2019. Machine-Learning Models for Sales Time Series Forecasting. *Data* 4, 1 (2019), 15. https://doi.org/10.3390/data4010015

[195] B Peng, P. Fatourou, and Themis Palpanas. 2021. SING: Sequence Indexing Using GPUs. In *IEEE 37th International Conference on Data Engineering*. Chania, 1883–1888. https://doi.org/10.1109/ICDE51399.2021.00171

[196] Joao Pereira and Margarida Silveira. 2018. Unsupervised Anomaly Detection in Energy Time Series Data Using Variational Recurrent Autoencoders with Attention. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 1275–1282. https://doi.org/10.1109/ICMLA.2018.00207

[197] Muskaan Pirani, Paurav Thakkar, Pranay Jivrani, Mohammed Husain Bohara, and Dweepna Garg. 2022. A Comparative Analysis of ARIMA, GRU, LSTM and BiLSTM on Financial Time Series Forecasting. In *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*. IEEE, 1–6. https://doi.org/10.1109/ICDCECE53908.2022.9793213

[198] I. Pitas and A. Karasaridis. 1996. Multichannel Transforms for Signal/Image Processing. *IEEE Transactions on Image Processing* 5, 10 (1996), 1402–1413. https://doi.org/10.1109/83.536889

[199] Plotly Technologies Inc. 2015. Collaborative Data Science. https://plot.ly

[200] Thomas Plötz and Gernot A. Fink. 2011. *Markov Models for Handwriting Recognition*. Springer London, London. https://doi.org/10.1007/978-1-4471-2188-6

[201] I. Popivanov and R. J. Miller. 2002. Similarity Search over Time-series Data using Wavelets. In *Proceedings 18th International Conference on Data Engineering*. 212–221. https://doi.org/10.1109/ICDE.2002.994711

[202] K. M. M. Prabhu. 2018. *Window Functions and Their Applications in Signal Processing*. CRC Press, Boca Raton. https://doi.org/10.1201/9781315216386

[203] Jinchuan Qian, Zhihuan Song, Yuan Yao, Zheren Zhu, and Xinmin Zhang. 2022. A Review on Autoencoder Based Representation Learning for Fault Detection and Diagnosis in Industrial Processes. *Chemometrics and Intelligent Laboratory Systems* 231 (2022), 104711. https://doi.org/10.1016/j.chemolab.2022.104711

[204] J. R. Quinlan. 1986. Induction of Decision Trees. *Machine Learning* 1, 1 (1986), 81–106. https://doi.org/10.1007/BF00116251

[205] E. Ramanujam and S. Padmavathi. 2022. Comprehensive Review on Time Series Motif Discovery Using Evolutionary Techniques. *International Journal of Advanced Intelligence Paradigms* 23, 1/2 (2022), 155. https://doi.org/10.1504/IJAIP.2022.125239

[206] Somayeh Bakhtiari Ramezani, Brad Killen, Logan Cummins, Shahram Rahimi, Amin Amirlatifi, and Maria Seale. 2021. A Survey of HMM-based Algorithms in Machinery Fault Prediction. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1–9. https://doi.org/10.1109/SSCI50451.2021.9659838

[207] K. Deergha Rao and M.N.S. Swamy. 2018. *Digital Signal Processing*. Springer Singapore, Singapore. https://doi.org/10.1007/978-981-10-8081-4

[208] Kamisetty Ramam Rao and Patrick C. Yip. 2018. *The Transform and Data Compression Handbook*. CRC Press. https://doi.org/10.1201/9781315220529

[209] Paulo E. Rauber, Alexandre X. Falcão, and Alexandru C. Telea. 2016. Visualizing Time-Dependent Data Using Dynamic t-SNE. In *Proceedings of the Eurographics / IEEE VGTC Conference on Visualization: Short Papers (EuroVis '16)*. Eurographics Association, Goslar, DEU, 73–77.

[210] Adi Raveh and Charles S. Tapiero. 1980. Finding Common Seasonal Patterns Among Time Series: An MDS Approach. *Journal of Econometrics* 12, 3 (1980), 353–363. https://doi.org/10.1016/0304-4076(80)90061-5

[211] Paulo Angelo Alves Resende and André Costa Drummond. 2019. A Survey of Random Forest Based Methods for Intrusion Detection Systems. *ACM Comput. Surv.* 51, 3 (2019), 1–36. https://doi.org/10.1145/3178582

[212] Manel Rhif, Ali Ben Abbes, Imed Farah, Beatriz Martínez, and Yanfang Sang. 2019. Wavelet Transform Application for/in Non-Stationary Time-Series Analysis: A Review. *Applied Sciences* 9, 7 (2019), 1345. https://doi.org/10.3390/app9071345

[213] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. Association for Computing Machinery, New York, NY, USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[214] Jussi Ronkainen and Antti Iivari. 2015. Designing a Data Management Pipeline for Pervasive Sensor Communication Systems. *Procedia Computer Science* 56 (2015), 183–188. https://doi.org/10.1016/j.procs.2015.07.193

[215] F. Rosenblatt. 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological review* 65, 6 (1958), 386–408. https://doi.org/10.1037/h0042519

[216] Oliver Rösler. 2013. EEG Eye State. https://doi.org/10.24432/C57G7J Accessed on 2024-02-01..

[217] Andrew L. Rukhin. 2002. Analysis of Time Series Structure SSA and Related Techniques. *Technometrics* 44, 3 (2002), 290. https://doi.org/10.1198/004017002320256477

[218] David E. Rumelhart and James L. McClelland. 1986. *Parallel Distributed Processing*. The MIT Press. https://doi.org/10.7551/mitpress/5236.001.0001

[219] Omar M. Saad and Yangkang Chen. 2020. Deep Denoising Autoencoder for Seismic Random Noise Attenuation. *GEOPHYSICS* 85, 4 (2020), V367–V376. https://doi.org/10.1190/geo2019-0468.1

[220] Rebecca Salles, Kele Belloze, Fabio Porto, Pedro H. Gonzalez, and Eduardo Ogasawara. 2019. Nonstationary Time Series Transformation Methods: An Experimental Review. *Knowledge-Based Systems* 164 (2019), 274–291. https://doi.org/10.1016/j.knosys.2018.10.041

[221] K. Krishna Rani Samal, Korra Sathya Babu, Santosh Kumar Das, and Abhirup Acharaya. 2019. Time Series Based Air Pollution Forecasting Using SARIMA and Prophet Model. In *Proceedings of the 2019 International Conference on Information Technology and Computer Communications*. ACM, New York, NY, USA, 80–85. https://doi.org/10.1145/3355402.3355417

[222] J. W. Sammon. 1969. A Nonlinear Mapping for Data Structure Analysis. *IEEE Trans. Comput.* C-18, 5 (1969), 401–409. https://doi.org/10.1109/T-C.1969.222678

[223] Nicholas Sapankevych and Ravi Sankar. 2009. Time Series Prediction Using Support Vector Machines: A Survey. *IEEE Computational Intelligence Magazine* 4, 2 (2009), 24–38. https://doi.org/10.1109/MCI.2009.932254

[224] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly Detection in Time Series: A Comprehensive Evaluation. *Proc. VLDB Endow.* 15, 9 (2022), 1779–1797. https://doi.org/10.14778/3538598.3538602

[225] Chandra Shah. 1997. Model Selection in Univariate Time Series Forecasting Using Discriminant Analysis. *International Journal of Forecasting* 13, 4 (1997), 489–500. https://doi.org/10.1016/S0169-2070(97)00031-9

[226] Sudhi Sharma, Vaibhav Aggarwal, and Miklesh Prasad Yadav. 2021. Comparison of Linear And Non-Linear GARCH Models for Forecasting Volatility of Select Emerging Countries. *Journal of Advances in Management Research* 18, 4 (2021), 526–547. https://doi.org/10.1108/JAMR-07-2020-0152

[227] Md. Shiblee, P. K. Kalra, and B. Chandra. 2009. Time Series Prediction with Multilayer Perceptron (MLP): A New Generalized Error Based Approach. In *Advances in Neuro-Information Processing*, Mario Köppen, Nikola Kasabov, and George Coghill (Eds.). Lecture Notes in Computer Science, Vol. 5507. Springer Berlin Heidelberg, Berlin, Heidelberg, 37–44. https://doi.org/10.1007/978-3-642-03040-6{_}5

[228] Jin Shieh and Eamonn Keogh. 2008. iSAX: Indexing and Mining Terabyte Sized Time Series. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, Ying Li, Bing Liu, and Sunita Sarawagi (Eds.). ACM, New York, NY, USA, 623–631. https://doi.org/10.1145/1401890.1401966

[229] Robert H. Shumway and David S. Stoffer. 2017. *Time Series Analysis and Its Applications*. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-52452-8

[230] Nripendra Kumar Singh and Khalid Raza. 2021. Medical Image Generation Using Generative Adversarial Networks: A Review. In *Health Informatics: A Computational Perspective in Healthcare*, Ripon Patgiri, Anupam Biswas, and Pinki Roy (Eds.). Studies in Computational Intelligence, Vol. 932. Springer Singapore, Singapore, 77–96. https://doi.org/10.1007/978-981-15-9735-0{_}5

[231] Shailendra Singh and Abdulsalam Yassine. 2018. Big Data Mining of Energy Time Series for Behavioral Analytics and Energy Consumption Forecasting. *Energies* 11, 2 (2018), 452. https://doi.org/10.3390/en11020452

[232] R. Solera-Ureña, J. Padrell-Sendra, D. Martín-Iglesias, A. Gallardo-Antolín, C. Peláez-Moreno, and F. Díaz-de María. 2007. SVMs for Automatic Speech Recognition: A Survey. In *Progress in Nonlinear Speech Processing*, Yannis Stylianou, Marcos Faundez-Zanuy, and Anna Esposito (Eds.). Lecture Notes in Computer Science, Vol. 4391. Springer Berlin Heidelberg, Berlin, Heidelberg, 190–216. https://doi.org/10.1007/978-3-540-71505-4{_}11

[233] Pranay Kumar Venkata Sowdaboina, Sutanu Chakraborti, and Somayajulu Sripada. 2014. Learning to Summarize Time Series Data. In *Computational Linguistics and Intelligent Text Processing*, Alexander Gelbukh (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 515–528.

[234] Karsten Sternickel. 2002. Automatic Pattern Recognition in ECG Time Series. *Computer Methods and Programs in Biomedicine* 68, 2 (2002), 109–115. https://doi.org/10.1016/S0169-2607(01)00168-7

[235] Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M. Rush. 2019. Seq2seq-Vis: A Visual Debugging Tool for Sequence-to-Sequence Models. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 353–363. https://doi.org/10.1109/TVCG.2018.2865044

[236] Sean J. Taylor and Benjamin Letham. 2017. Forecasting at Scale. https://doi.org/10.7287/peerj.preprints.3190v2

[237] H. Tong and K. S. Lim. 1980. Threshold Autoregression, Limit Cycles and Cyclical Data. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 42, 3 (1980), 245–268. https://doi.org/10.1111/j.2517-6161.1980.tb01126.x

[238] Sahar Torkamani and Volker Lohweg. 2017. Survey on Time Series Motif Discovery. *WIREs Data Mining and Knowledge Discovery* 7, 2 (2017), e1199. https://doi.org/10.1002/widm.1199

[239] Patara Trirat, Yooju Shin, Junhyeok Kang, Youngeun Nam, Jihye Na, Minyoung Bae, Joeun Kim, Byunghyun Kim, and Jae-Gil Lee. 2024. Universal Time-Series Representation Learning: A Survey. https://doi.org/10.48550/arXiv.2401.03717

[240] Prodromos E. Tsinaslanidis and Dimitris Kugiumtzis. 2014. A Prediction Scheme Using Perceptually Important Points and Dynamic Time Warping. *Expert Systems with Applications* 41, 15 (2014), 6848–6860. https://doi.org/10.1016/j.eswa.2014.04.028

[241] Hristos Tyralis and Georgia Papacharalampous. 2017. Variable Selection in Time Series Forecasting Using Random Forests. *Algorithms* 10, 4 (2017), 114. https://doi.org/10.3390/a10040114

[242] Hristos Tyralis, Georgia Papacharalampous, and Andreas Langousis. 2019. A Brief Review of Random Forests for Water Scientists and Practitioners and Their Recent History in Water Resources. *Water* 11, 5 (2019), 910. https://doi.org/10.3390/w11050910

[243] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine learning research* 9, Nov (2008), 2579–2605.

[244] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \L ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[245] Ronny Velastegui, Luis Zhinin-Vera, Gissela E. Pilliza, and Oscar Chang. 2021. Time Series Prediction by Using Convolutional Neural Networks. In *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1*, Kohei Arai, Supriya Kapoor, and Rahul Bhatia (Eds.). Advances in Intelligent Systems and Computing, Vol. 1288. Springer International Publishing, Cham, 499–511. https://doi.org/10.1007/978-3-030-63128-4{_}38

[246] J. Fernando Vera and José M. Angulo. 2023. An MDS-Based Unifying Approach to Time Series K-Means Clustering: Application in the Dynamic Time Warping Framework. *Stochastic Environmental Research and Risk Assessment* 37, 12 (2023), 4555–4566. https://doi.org/10.1007/s00477-023-02470-9

[247] E. F. Vernier, R. Garcia, I. P. Da Silva, J. L. D. Comba, and A. C. Telea. 2020. Quantitative Evaluation of Time–Dependent Multidimensional Projection Techniques. *Computer Graphics Forum* 39, 3 (2020), 241–252. https://doi.org/10.1111/cgf.13977

[248] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. 2006. Indexing Multidimensional Time-Series. *The VLDB Journal* 15, 1 (2006), 1–20. https://doi.org/10.1007/s00778-004-0144-2

[249] Milena Vuletić, Felix Prenzel, and Mihai Cucuringu. 2024. Fin-GAN: Forecasting and Classifying Financial Time Series via Generative Adversarial Networks. *Quantitative Finance* 24, 2 (2024), 175–199. https://doi.org/10.1080/14697688.2023.2299466

[250] Jinghui Wang and Shugang Tang. 2020. Time Series Classification Based on ARIMA and Adaboost. *MATEC Web of Conferences* 309 (2020), 03024. https://doi.org/10.1051/matecconf/202030903024

[251] Kang Wang, Kenli Li, Liqian Zhou, Yikun Hu, Zhongyao Cheng, Jing Liu, and Cen Chen. 2019. Multiple convolutional neural networks for multivariate time series prediction. *Neurocomputing* 360 (2019), 107–119. https://doi.org/10.1016/j.neucom.2019.05.023

[252] Lin Wang, Faming Lu, Minghao Cui, and Yunxia Bao. 2019. Survey of Methods for Time Series Symbolic Aggregate Approximation. In *Data Science*, Xiaohui Cheng, Weipeng Jing, Xianhua Song, and Zeguang Lu (Eds.). Communications in Computer and Information Science, Vol. 1058. Springer Singapore, Singapore, 645–657. https://doi.org/10.1007/978-981-15-0118-0{_}50

[253] Muhammad Waqas and Usa Wannasingha Humphries. 2024. A Critical Review of RNN and LSTM Variants in Hydrological Time Series Predictions. *MethodsX* 13 (2024), 102946. https://doi.org/10.1016/j.mex.2024.102946

[254] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. 2016. How to Use t-SNE Effectively. *Distill* 1, 10 (2016). https://doi.org/10.23915/distill.00002

[255] William W. S. Wei. 2019. *Multivariate Time Series Analysis and Applications.* John Wiley & Sons, Hoboken NJ.

[256] Xiaoqing Weng and Shimin Qin. 2012. Classification of Multivariate Time Series Using Supervised Isomap. In *2012 Third Global Congress on Intelligent Systems*. IEEE, 136–139. https://doi.org/10.1109/GCIS.2012.31

[257] Xiaoqing Weng and Junyi Shen. 2008. Classification of Multivariate Time Series Using Two-Dimensional Singular Value Decomposition. *Knowledge-Based Systems* 21, 7 (2008), 535–539. https://doi.org/10.1016/j.knosys.2008.03.014

[258] Christopher K. I. Williams and Carl Edward Rasmussen. 2006. *Gaussian Processes for Machine Learning*. Vol. 2. MIT press Cambridge, MA.

[259] Seunghye J. Wilson. 2017. Data Representation for Time Series Data Mining: Time Domain Approaches. *WIREs Computational Statistics* 9, 1 (2017). https://doi.org/10.1002/wics.1392

[260] Kwan Yeung Wong and Fu-lai Chung. 2019. Visualizing Time Series Data with Temporal Matching Based t-SNE. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8. https://doi.org/10.1109/IJCNN.2019.8851847

[261] Timothy Wong and Zhiyuan Luo. 2018. Recurrent Auto-Encoder Model for Large-Scale Industrial Sensor Signal Analysis. In *Engineering Applications of Neural Networks*, Elias Pimenidis and Chrisina Jayne (Eds.). Communications in Computer and Information Science, Vol. 893. Springer International Publishing, Cham, 203–216. https://doi.org/10.1007/978-3-319-98204-5{_}17

[262] Changhao Xia, Mi Zhang, and Jin Cao. 2018. A Hybrid Application of Soft Computing Methods With Wavelet Svm and Neural Network to Electric Power Load Forecasting. *Journal of Electrical Systems and Information Technology* 5, 3 (2018), 681–696. https://doi.org/10.1016/j.jesit.2017.05.008

[263] Peter T. Yamak, Li Yujian, and Pius K. Gadosey. 2019. A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence.* ACM, New York, NY, USA, 49–55. https://doi.org/10.1145/3377713.3377722

[264] Lijuan Yan, Yanshen Liu, and Yi Liu. 2020. Interval Feature Transformation for Time Series Classification Using Perceptually Important Points. *Applied Sciences* 10, 16 (2020), 5428. https://doi.org/10.3390/app10165428

[265] Jian Bo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15).* AAAI Press, 3995–4001.

[266] Kiyoung Yang and Cyrus Shahabi. 2004. A PCA-Based Similarity Measure for Multivariate Time Series. In *Proceedings of the 2nd ACM international workshop on Multimedia databases*, Shu-Ching Chen and Mei-Ling Shyu (Eds.). ACM, New York, NY, USA, 65–74. https://doi.org/10.1145/1032604.1032616

[267] Zheng Yang, Binbin Xu, Wei Luo, and Fei Chen. 2022. Autoencoder-Based Representation Learning and Its Application in Intelligent Fault Diagnosis: A Review. *Measurement* 189 (2022), 110460. https://doi.org/10.1016/j.measurement.2021.110460

[268] Lexiang Ye and Eamonn Keogh. 2009. Time Series Shapelets: A New Primitive for Data Mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, John Elder, Françoise Soulié Fogelman, Peter Flach, and Mohammed Zaki (Eds.). ACM, New York, NY, USA, 947–956. https://doi.org/10.1145/1557019.1557122

[269] Chunyong Yin, Sun Zhang, Jin Wang, and Neal N. Xiong. 2022. Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52, 1 (2022), 112–122. https://doi.org/10.1109/TSMC.2020.2968516

[270] Umaporn Yokkampon, Abbe Mowshowitz, Sakmongkon Chumkamon, and Eiji Hayashi. 2022. Robust Unsupervised Anomaly Detection With Variational Autoencoder in Multivariate Time Series Data. *IEEE Access* 10 (2022), 57835–57849. https://doi.org/10.1109/ACCESS.2022.3178592

[271] Wennian Yu, Il Yong Kim, and Chris Mechefske. 2021. Analysis of Different Rnn Autoencoder Variants for Time Series Classification and Machine Prognostics. *Mechanical Systems and Signal Processing* 149 (2021), 107322. https://doi.org/10.1016/j.ymssp.2020.107322

[272] Yuncong Yu. 2024. NOOPS: Nonmyopic Feedback-Driven Multivariate Time Series Pattern Search. osf.io/6k4j2

[273] Yuncong Yu, Tim Becker, Le Minh Trinh, and Michael Behrisch. 2023. SAXRegEx: Multivariate time series pattern search with symbolic representation, regular expression, and query expansion. *Computers & Graphics* 112 (2023), 13–21. https://doi.org/10.1016/j.cag.2023.03.002

[274] Yungcong Yu, Dylan Kruyff, Jiao Jiao, Tim Becker, and Michael Behrisch. 2022. PSEUDo: Interactive Pattern Search in Multivariate Time Series with Locality-Sensitive Hashing and Relevance Feedback. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 33–42. https://doi.org/10.1109/TVCG.2022.3209431

[275] Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. 2025. Deep Learning for Time Series Anomaly Detection: A Survey. *ACM Comput. Surv.* 57, 1 (2025), 1–42. https://doi.org/10.1145/3691338

[276] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 9 (2023), 11121–11128. https://doi.org/10.1609/aaai.v37i9.26317

[277] Chunfang Zhang and Claudia Czado. 2023. Vine Copula-Based Bayesian Classification for Multivariate Time Series of Electroencephalography Eye States. *Journal of the Royal Statistical Society Series C: Applied Statistics* (2023), qlad038. https://doi.org/10.1093/jrsssc/qlad038

[278] Dongyu Zhang, Wangmeng Zuo, David Zhang, and Hongzhi Zhang. 2010. Time Series Classification Using Support Vector Machine with Gaussian Elastic Metric Kernel. In *2010 20th International Conference on Pattern Recognition*. IEEE, 29–32. https://doi.org/10.1109/ICPR.2010.16

[279] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2019. Self-Attention Generative Adversarial Networks. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 7354–7363. https://proceedings.mlr.press/v97/zhang19d.html

[280] Yuxin Zhang, Yiqiang Chen, Jindong Wang, and Zhiwen Pan. 2021. Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals. *IEEE Transactions on Knowledge and Data Engineering* (2021), 1. https://doi.org/10.1109/TKDE.2021.3102110

[281] Zhijie Zhang, Wenzhong Li, Wangxiang Ding, Linming Zhang, Qingning Lu, Peng Hu, Tong Gui, and Sanglu Lu. 2023. STAD-GAN: Unsupervised Anomaly Detection on Multivariate Time Series with Self-training Generative Adversarial Networks. *ACM Trans. Knowl. Discov. Data* 17, 5 (2023), 1–18. https://doi.org/10.1145/3572780

[282] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. 2017. Convolutional Neural Networks for Time Series Classification. *Journal of Systems Engineering and Electronics* 28, 1 (2017), 162–169. https://doi.org/10.21629/JSEE.2017.01.18

[283] Xia Zhao, Limin Wang, Yufei Zhang, Xuming Han, Muhammet Deveci, and Milan Parmar. 2024. A Review of Convolutional Neural Networks in Computer Vision. *Artificial Intelligence Review* 57, 4 (2024). https://doi.org/10.1007/s10462-024-10721-6

[284] Zhong Zheng, Zijun Zhang, Long Wang, and Xiong Luo. 2022. Denoising Temporal Convolutional Recurrent Autoencoders for Time Series Classification. *Information Sciences* 588 (2022), 159–173. https://doi.org/10.1016/j.ins.2021.12.061

[285] Dazhuo Zhou, JinXia Li, and WenXiu Ma. 2009. Clustering Based on LLE For Financial Multivariate Time Series. In *2009 International Conference on Management and Service Science*. IEEE, 1–4. https://doi.org/10.1109/ICMSS.2009.5305089

[286] Andreas Ziegler and Inke R. König. 2014. Mining Data With Random Forests: Current Options for Real–World Applications. *WIREs Data Mining and Knowledge Discovery* 4, 1 (2014), 55–63. https://doi.org/10.1002/widm.1114

[287] Walter Zucchini and Iain L. MacDonald. 2009. *Hidden Markov Models for Time Series*. Chapman and Hall/CRC. https://doi.org/10.1201/9781420010893

A LIST OF SYMBOLS

The symbols listed in this appendix have fixed meanings. Unlisted symbols used in the main text may vary in meaning according to the context. Nonetheless, unlisted symbols are rare cases.

We use lowercase letters for scalars, including scalar functions (i.e., functions returning one scalar as output), boldface letters for vectors, uppercase letters for matrices or random variables (except the lowercase ε for the error as a random variable). This rule applies to both Latin and Greek symbols, unless they are in special fonts, including `\mathcal` (for established distribution like the normal distribution $\mathcal{N}$, or operators and transforms like the Fourier Transform $\mathcal{F}$) and `\mathbb` (for well-known sets like the set of real numbers $\mathbb{R}$.

Table 2. *Latin Symbols*

Symbol	Meaning
A	The coefficient matrix before $\boldsymbol{S}_{t-1}$ for estimating $\boldsymbol{S}_t$ in a SSM, aka the state/system matrix.
B	The coefficient matrix before $\boldsymbol{U}_t$ for estimating $\boldsymbol{S}_t$ in a SSM, aka the input matrix.
$\mathcal{B}$	The lag operator, e.g., $\mathcal{B}^i Y_t = Y_{t-i}$.
C	The coefficient matrix before $\boldsymbol{S}_t$ for estimating $\boldsymbol{Y}_t$ in a SSM, aka the output matrix.
$\mathrm{Cov}(.,.)$	Covariance.
D	The coefficient matrix before $\boldsymbol{U}_t$ for estimating $\boldsymbol{Y}_t$ in a SSM, aka the feedthrough matrix.
$E(.)$	The expected value.
I	The identity matrix.
$\mathcal{I}$	An integral transform.
$\mathcal{K}$	The kernel of an integral transform.
M	An example matrix.
$\mathcal{N}$	The normal distribution.
$P(.)$	Probability.
$\mathbb{R}$	The set of real numbers.
T	Transpose.
S	Covariance matrix.
$\boldsymbol{S}_t$	The state variables at t-th time point in a SSM.
U	Matrix whose columns are the left singular vectors in SVD.
$\boldsymbol{U}_t$	The input variables at t-th time point in a SSM.
V	Matrix whose columns are the right singular vectors in SVD.
$\mathrm{var}(.)$	Variance.
X	A data set containing multiple samples, each of multiple dimensions.
$\boldsymbol{X}_t$	The exogenous variables at t-th time point in a VARX model.
$\mathcal{W}$	The wavelet transform.
Y_t	The t-th time point as a scalar random variable in a univariate time series.
a	The parameter in WT that scales the mother wavelet.

⋮

Table 2. *Latin Symbols – Continued*

Symbol	Meaning
a_i	Coefficient for cos in i-th term in a Fourier series.
a_i	Fourier coefficient for cos.
$\widetilde{a}$	The parameter in DWT that controls scaling the mother wavelet, $a = 2^{\widetilde{a}}$.
b	The parameter in WT that translates the mother wavelet.
b_i	Coefficient for sin in i-th term in a Fourier series.
$\widetilde{b}$	The parameter in DWT that controls translation of the mother wavelet, $b = 2^{\widetilde{a}}\widetilde{b}$.
c	A constant scalar bias.
c_i	The i-the constant used when defining the strict stationarity.
$\boldsymbol{c}$	A constant vector bias.
d	The degree of differencing in an ARIMA model.
$d_{i,j}$	The distance between the i-th and the j-th data point in the original space.
$d'_{i,j}$	The distance between the i-th and the j-th data point in the image space.
$\widetilde{d}$	The seasonal degree of differencing in a SARIMA model.
e	The base of the natural logarithm and exponential function.
f	Frequency.
f_0	The central frequency in the Morlet wavelet.
g	The trend function in the Prophet model.
i	An index variable with flexible/unlimited usage.
j	The imaginary unit or an index variable with flexible usage when i is used.
k	The parameter in FT that is related to the frequency.
$\boldsymbol{k}$	The parameters of a kernel function.
m	The number of channels in the time series, or more generally, the number of dimensions of the data points in a data set.
m'	The number of exogenous variables in a VARX model.
n	The number of time points in a time series, or more generally, the number of data points in a data set.
n_w	The length (number of time points) of a window function or of a sliding window.
o	The number of Fourier series terms in the Prophet model.
p	The number of lagged values of the (endogenous) variables in an AR model or the number of lagged conditional variances in a GARCH model.
p'	The number of lagged values of the exogenous variables to use in a VARX model.
$\widetilde{p}$	The seasonal AR order in a SARIMA model.
$p.v.$	Cauchy principal value.
q	The number of lagged errors in a MA model or in an ARCH model.
$\widetilde{q}$	The seasonal MA order in a SARIMA model.

⋮

Table 2. *Latin Symbols – Continued*

Symbol	Meaning
$\widetilde{s}$	The length of a season in an SARIMA model or a Prophet model.
t	The time or the zero-based index of time points in a time series.
Δt	The time difference.
v	The zero-based index of channel/variables in a time series.
$\boldsymbol{v}_i$	The unit vector of the i-th principal component.
w	The window function used in STFT.
y_t	The scalar value in t-th time point of a univariate time series.
$y_{t,v}$	The scalar value in t-th time point and v-th channel of a multivariate time series.
y	A continuous scalar function of time.
$\boldsymbol{y}_t$	The values (as a vector random variable) at t-th time point in a multivariate time series.
z	The boolean ($z(t) \in \{0, 1\}$) holiday/event function in the Prophet model.

Table 3. *Greek Symbols*

Symbol	Meaning
Γ_i	The coefficient for $\boldsymbol{X}_{t-i}$ in a VARX model.
Σ	Matrix with the singular values in SVD or summation.
Φ_i	The coefficient for $\boldsymbol{Y}_{t-i}$ in a VAR model.
α	A coefficient with flexible usage.
α_i	The coefficient for ε^2_{t-i} in a (G)ARCH model.
β	A coefficient with flexible usage. Used when α is used.
β_i	The coefficient for σ^2_{t-i} in a GARCH model.
ε_t	The error of Y_t as a random variable.
$\boldsymbol{\varepsilon}_t$	The error of $\boldsymbol{Y}_t$ as a random variable.
θ_i	The coefficient for ε_{t-i} in a MA model.
$\widetilde{\theta}_i$	The coefficient for the i-th seasonal MA component in an SARIMA model.
κ	A parameter following a zeor-mean normal distribution in the Prophet model.
π	The ratio of a circle's circumference to its diameter.
τ	The time shift used, e.g., to shift a window function.
ϕ_i	The coefficient for Y_{t-i} in an AR model.
$\widetilde{\phi}_i$	The coefficient for the i-th seasonal AR component in an SARIMA model.
σ	The standard deviation.
σ^2_t	The conditional variance / heteroscedasticity of Y_t.
ψ	The mother wavelet.
⋮	

Table 3. *Greek Symbols – continued*

Symbol	Meaning
$\psi_{a,b}$	The wavelet from the mother wavelet ψ with scaling parameter a and translation parameter b.

Table 4. *Other Symbols*

Symbol	Meaning
$\|.\|$	The absolute value of a scalar.
$\|\|.\|\|$	The second norm of a vector.
:=	Assignment.

B LIST OF ABBREVIATIONS

ACF	Autocorrelation Function	12
AE	Autoencoder	47
ANN	Artificial Neural Network	16
APCA	Adaptive Piecewise Constant Approximation	15
AR	AutoRegressive	9
ARCH	AutoRegressive Conditional Heteroskedasticity	9
ARIMA	AutoRegressive Integrated Moving Average	2
ARIMAX	AutoRegressive Integrated Moving Average with eXogenous inputs	9
ARMA	AutoRegressive Moving-Average	6
CNN	Convolutional Neural Network	16
CWT	Continuous Wavelet Transform	14
DFT	Discrete Fourier Transform	6
DR	Dimensionality Reduction	6
DLM	Dynamic Linear Model	11
DTFT	Discrete-Time Fourier Transform	13
DWT	Discrete Wavelet Transform	14
ED	Euclidean Distance	20
ECG	Electrocardiogram	4
EEG	Electroencephalogram	4
EMD	Empirical Mode Decomposition	14
FFT	Fast Fourier transform	2
FT	Fourier Transform	13
FWT	Fast Wavelet Transform	14
GAN	Generative Adversarial Network	6
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity	12
GP	Gaussian Process	22
GRU	Gated Recurrent Unit	17
HHT	Hilbert-Huang Transform	14
HMM	Hidden Markov Model	6
i.i.d	independent and identically distributed	9
IMF	Intrinsic Mode Function	14
IoT	Internet of Things	3
KL	Kullback–Leibler	20
KLT	Karhunen–Loève Transform	19
LDA	Linear Discriminant Analysis	19
LLE	Locally Linear Embedding	20
MLP	Multilayer Perceptron	16

44 Yuncong Yu, Tim Becker, Wolfgang Aigner, Alexandru C. Telea, and Michael Behrisch

LSTM	Long Short-Term Memory	17
MA	Moving-Average	9
MDS	Multidimensional Scaling	21
ML	Machine Learning	15
NLP	Natural Language Processing	17
PAA	Piecewise Aggregate Approximation	15
PACF	Partial Autocorrelation Function	12
PC	Principal Component	19
PCA	Principal Component Analysis	18
PCA	Piecewise Constant Approximation	15
PIP	Perceptually Important Point	4
PLR	Piecewise Linear Representation	15
PPR	Piecewise Polynomial Representation	15
RBF	Radial Basis Function	45
RNN	Recurrent Neural Network	16
SAX	Symbolic Aggregate approXimation	6
SOM	Self-Organising Map	21
SSM	State Space Model	11
SARIMA	Seasonal AutoRegressive Integrated Moving Average	9
SETAR	Self-Exciting Threshold AutoRegressive	12
STFT	Short-Time Fourier Transform	13
SSM	State-Space Model	11
SSA	Singular Spectrum Decomposition	19
STAR	State-of-the-Art Report	1
SVD	Singular Value Decomposition	19
SVM	Support Vector Machine	15
TAR	Threshold AutoRegressive	12
t-SNE	t-distributed Stochastic Neighbor Embedding	18
TSA	Time Series Analysis	1
TSRT	Time Series Representation Technique	1
TTS	Text to Speech	45
UMAP	Uniform Manifold Approximation and Projection	20
VAE	Variational Autoencoder	17
VAR	Vector AutoRegressive	9
VARX	Vector AutoRegressive eXogenous	11
VARMA	Vector AutoRegressive Moving-Average	10
WoS	Web of Science	7
WT	Wavelet Transform	13

Manuscript submitted to ACM

C REPRESENTATIVE TIME SERIES REPRESENTATIONS

Due to the space limit, we abbreviated the categories in the second column, i.e., SP: stochastic process, IT: integral transform, PR: piecewise representation, ML: machine learning model, DR: dimensionality reduction technique, Mi: miscellaneous.

Table 5. *Representative Time Series Representation Techniques*

TSRT	Category	Stationarity*	Linearity**	Markov Prop.	(A)Periodicity	Univariance	Normality	Kernel Func.	T+S(+X)***	Annotations to Data Assumptions and Other Limitations	Typical Use Cases****
AutoRegressive Integrated Moving Average (ARIMA)	SP	$\times^{1}$	$\times^{2}$		$\times^{1}$	$\times^{3}$	$\times^{4}$			[1] **Stationarity:** constant mean and variance over time after detrending with the integral part, otherwise special treatment for known non-stationarity. For seasonal non-stationarity, use seasonal ARIMA (SARIMA). [2] **Linearity:** relationship between variables and between current value and historical values are linear. [3] **Univariance:** the original ARIMA is developed for univariate time series; use vector ARIMA (VARIMA) for multivariate time series. [4] **Normality:** the error terms in the moving average part are often assumed to be i.i.d samples from a zero-mean normal distribution.	Forecast in economics/finance/business [6], environmental science [133]
(Generalized) Autoregressive Conditional Heteroskedasticity ((G)ARCH)	SP	$\times^{1}$	$\times^{1}$			$\times^{1}$	$\times^{1}$			[1] **Mean Process:** (G)ARCH only models the variances of the error terms. It requires a model / mean process to model the mean, e.g, an AR model. As a result, the assumptions primarily refer to the error terms. **Conditional Heteroskedasticity:** the variances of errors follow an AR model (ARCH) or ARMA model (GARCH).	Forecast of volatility of stock prices [88].
HMM	SP			$\times^{1}$						[1] **Markov Property:** the next future state (state is hidden for HMM) depends only on the current state and not on the states that occurred before it. Observation Independence (for HMM): the current observation (time series value) does not depend on previous or future observations but only on the current state. **Finite States and Observations:** the number of states and possible observations are finite. **Time-Invariant Transition Probabilities:** the probability of moving from one state to another does not change over time. It can also be regarded as a kind of stationarity.	Speech recognition (HMM is the most frequently used method for speech recognition [232]) [46, 125], gesture/posture recognition [71, 180], handwriting recognition [200], Text to Speech (TTS) [134], predictive maintenance [206].
Gaussian Process (GP)	SP						$\times^{1}$	$\times^{2}$		[1] **Normality:** values at any subset of time points follow a multivariate normal distribution. [2] **Kernal Function:** assumptions made when choosing the kernel function / covariance function (how strongly points in the process should correlate with each other), popular kernel choices are the Radial Basis Function (RBF) kernel (the most frequently used GP kernel, aka Gaussian kernel or squared exponential kernel), the radio quadratic kernel, the exponential sine squared kernel, etc. [258].	General regression [4, 72, 175], interpolation, Bayesian optimization [26] (for complex function/system whose inference is expensive.) General regression, prediction, filling gaps in data, bayesian optimization (e.g., hyperparameter optimization in ML), learning control policies, modeling system dynamics; but generally for modeling complex systems (like a complex engine model) that costs much time to run.

. . .

Table 5. *Representative Time Series Representation Techniques – Continued*

TSRT	Category	Stationarity*	Linearity**	Markov Prop.	(A)Periodicity	Univariance	Normality	Kernel Func.	T+S(+X)***	Annotations to Data Assumptions and Other Limitations	Typical Use Cases****
Discrete Fourier Transform (DFT)	IT	[1]			×[2]	×[3]				[1] **Time-Invariant Spectrum:** the frequency components do not change over time. This is why the literature claims DFT requires stationarity. But the column "stationary" refers to time-invariant mean and variance. This is not an assumption of DFT. [2] **Periodicity:** the input signal is periodic and continues indefinitely. The input of DFT is a period. A finite aperiodic time series is thus assumed to repeat itself infinitely. [3] **Univariance:** though no problem with multidimensional data (e.g. images), DFT for multivariate time series is less established, though there are research in this direction, like [198].	Signal filtering, compression, spetrum analysis, system identification, identifying/modeling cyclic patterns
Short-Time Fourier Transform (STFT)	IT					×[1]		×[2]		[1] **Univariance:** we did not find any multivariate version of STFT. [2] **Kernel:**The window function can be regarded as a kernel. Popular choices include the Gaussian window, the Hann window, the Hamming window, the Blackman window, etc. [202] **Resolution Trade-Off:** time and frequency resolutions cannot be high simultaneously. Larger window size leads to higher frequency resolution but lower time resolution, and vice versa.	Speech recognition, detect and track target in radar/sonar signals, anomaly/novelty detection in machine vibration and seismic data
Wavelet Transform (WT)	IT					×[1]		×[2]		[1] **Univariance:** though without problem for multidimensional data (e.g. images), WT for multivariate time series is less established, though there is research in this direction, like [156]. [2] **Kernel:** popular choices include the Morlet wavelet (the most frequently used), the Shannon wavelet, the Mexican hat wavelet, etc. for CWT and the Haar wavelet, the Coiflet wavelet, the Daubechies wavelet, etc. for DWT [101].	CWT: time-frequency analysis. Note that `cwt` in MATLAB and `scipy.signal.cwt` in Python are discrete versions of the theoretical CWT, practically DWT with higher resolutions, and thus applicable for time-frequency analysis for time series. DWT: compression (e.g. JPEG 2000 and MPEG-4), denoising, feature extraction, pattern recognition
HHT	IT					×[1]				[1] **Univariance:** we did not find any multivariate version of HHT. **Sufficient Oscillatory Behavior:** HHT's first step, EMD reuiqres sufficient oscillatory behavior in the time series for the extraction of meaningful intrisic mode functions. It may not work well on monotonic or very smooth data.	Revealing patterns in science (e.g., seismic and meteorological, astronomical data), medical (e.g., anomaly detection ECG and EEG), engineering (e.g., fault analysis for revolving machine, e.g., for bearing); as preprocessing step for prediction in financial data; image enhancement
Piecewise Aggregate Approximation (PAA)	PR					×[1]				[1] **Univariance:** PAA can be applied to each track individually. However, the interrelationships between tracks are not considered.	Time series smoothing, compression, and indexing
Piecewise Linear Representation (PLR)	PR					×[1]				[1] **Univariance:** same as above.	Time series smoothing, compression, and indexing
(indexable) Symbolic Aggregate approXimation ((i)SAX)	PR					×[1]	×[2]			[1] **Univariance:** same as above. [2] **Normality:** the values in the time series follows a normal distribution.	Time series smoothing, compression, and indexing (for retrieval)
SVM	ML		[1]					×[2]		[1] **Linearity:** can be chosen to be linear or nonlinear via the kernel selection. [2] **Kernel:** common kernel choices include the linear kernel (for linear SVM), the RBF kernel (the most frequently used for nonlinear SVM), the polynomial kernel, the sigmoid kernel, etc. [44]	Text classification [126], image recognition [47], speech recognition [232].
...											

Table 5. *Representative Time Series Representation Techniques – Continued*

TSRT	**Category**	**Stationarity***	**Linearity****	**Markov Prop.**	**(A)Periodicity**	**Univariance**	**Normality**	**Kernel Func.**	**T+S(+X)*****	**Annotations to Data Assumptions and Other Limitations**	**Typical Use Cases********
Random Forest / XGBoost	ML					1				[1] **Univariance:** the time series data can be flattened before fed to the model.	Various classification tasks like medical diagnosis, fraud detection [106, 188, 211], customer segmentation/recommendation; various regression tasks like price/sales/energy-consumption/weather forecasting [187, 242]
Recurrent Neural Network (RNN)	ML									No noteworthy assumptions	Time series prediction, text/audio generation, speed recognition, anomaly detection, video description
Convolutional Neural Network (CNN)	ML									No noteworthy assumptions	Computer vision [283], time series prediction, audio recognition, text embedding generation, anomaly detection
Attention/transform	ML									No noteworthy assumptions	NLP [10, 193], computer vision [141, 163]
(Variational) Autoencoder ((V)AE)	ML					1				[1] **Univariance:** the time series data can be flattened before fed to the model; otherwise, one can prepend layers supporting multi-channel data, or try convolutional (V)AE [148, 270] and recurrent (V)AE [196, 261].	**Autoencoder (AE):** anomaly detection [203, 267], dimensionality reduction / feature extraction (e.g., in pre-training), denoising **VAE:** Text/sequence(e.g., music, molecular structure)/image generation, anomaly detection, image denoising
Generative Adversarial Network (GAN)	ML									No noteworthy assumptions	Image/sequence (e.g. molecular structure) generation [128, 230], image super-resolutioning, image/text-to-image [7], style transfer, data augmentation, anomaly detection
Principal Component Analysis (PCA)	DR		×[1]							[1] **Lineariry:** the majority of the data variance can be explained by the linear combination of the first few principal components.	Data visualization, compression, denoising, feature extraction
Singular Value Decomposition (SVD)	DR		×[1]			2				[1] **Lineariry:** the data can be explained by the linear combination of left sigular vectors that are scaled by singular values and weighted by right sigular vectors. [2] **Multivariance:** according to the SVD implementation from Keogh et al for time series indexing, each column in the matrix to decompose is a segment in a univariate time series segmented by a sliding window [137] Namely, the matrix to decompose is the transpose of the trajectory matrix of the time series. Accordingly, it only works for univariate time series. However, we believe that it is feasible to 1) use the flattened sgements as the columns of the matrix to decompose in multivariate cases; or 2) use the whole multivariate time series directly as the matrix to decompose.	Latent semantic analysis, data compression, denoising
Singular Spectrum Decomposition (SSA)	DR		×			×			×[1]	[1] **Decomposition:** the time series consists of trend, periodic compoents, and noise	Forecast in meteorology, economics, finance, analyzing population dynamics in ecology

. . .

Table 5. *Representative Time Series Representation Techniques – Continued*

TSRT	Category	Stationarity*	Linearity**	Markov Prop.	(A)Periodicity	Univariance	Normality	Kernel Func.	T+S(+X)***	Annotations to Data Assumptions and Other Limitations	Typical Use Cases****
Various Manifold learning techniques	DR									**Single Manifold:** the high-dimensional data lies on or near a lower-dimensional manifold embedded within the higher-dimensional space; and data reside in a single manifold. **Smoothness:** small changes in the high-dimensional data should only result in small changes in the low-dimensional representation. **Uniform sample density:** uniform distribution of points along the manifold, rather than dense in some regions and sparse in others. Individual techniques may have their own data assumptions, for instance, Local Linear Embedding (LLE) assumes local linearity, isomap assumes isotropy (properties change in one direction on the manifold is the same as in any other direction), and Uniform Manifold Approximation and Projection (UMAP) assumes uniformly distributed data on a locally connected Riemannian manifold and that the Riemannian metric is locally constant or approximately locally constant	Data visualization, denoising, feature extraction, anomaly/novelty detection, pattern recognition
Prophet	Mi								×[1]	[1] **Decomposition:** a piecewise linear represented trend, seasonalities described by a Fourier series, and user-provided holidays.	Forecast In business, finance, economics, energy consumption, meteorology
Perceptually Important Point (PIP)	Mi					×[1]				[1] **Univariance:** we did not find any multivariate version of PIP. **Key Points:** there are "key" points in the time series data that capture most of its significant characteristics.	Time series summarization in meteorology and IoT, pattern recognition (e.g., "head-and-shoulders" and "'cup-and-handle") in financial market analysis and healthy monitoring, anomaly detection

Table Footnotes

* "Stationarity" in this table means weak stationarity, namely time-invariant mean, and auto-covariance only depends on the time lag. Please refer to Section 6.2 for details.

** "Linearity" in this table means that the data dynamic can be captured by a linear relationship. It does not mean that the transformation operation is linear or not. For instance, DFT is said to be linear, which is rather a desired property, instead of a data assumption/limitation. Please refer to Section 6.2 for details.

*** "T+S(+X)" in this table means that the time series is assumed to be able to be decomposed to trend components, seasonal/cyclic/periodic components, and possibly other components (noise/residual/holiday). Please refer to Section 6.2 for details.

**** We only add STARs including monographs and book chapters dedicated to the specific TSRT and to the specific use case in this table.