

IntechOpen

Time Series Analysis

Frontiers in Research and Practice

*Edited by Jorge Rocha, Cláudia M. Viana,
Inês Boavida-Portugal and Eduardo Gomes*



Time Series Analysis - Frontiers in Research and Practice

*Edited by Jorge Rocha, Cláudia M. Viana,
Inês Boavida-Portugal and Eduardo Gomes*

Published in London, United Kingdom

Time Series Analysis – Frontiers in Research and Practice

<http://dx.doi.org/10.5772/intechopen.1007996>

Edited by Jorge Rocha, Cláudia M. Viana, Inês Boavida-Portugal and Eduardo Gomes

Contributors

Alberto Pacheco, Alexander Lysyak, Chongxiao Zheng, Cláudia M. Viana, Eduardo Gomes, Ever Flores, Hirokazu Maekawa, Inês Boavida-Portugal, Jingrun Zhang, Jorge Rocha, Jose Duran, Mariano Rivera, Raul Chacon, Raymundo Torres, Regino Criado Herrero, Sergio Iglesias Pérez, Shigekazu Kusumoto, Xiaoxiang Guo, Yasuto Itoh, Zhen Li

© The Editor(s) and the Author(s) 2025

The rights of the editor(s) and the author(s) have been asserted in accordance with the Copyright, Designs and Patents Act 1988. All rights to the book as a whole are reserved by INTECHOPEN LIMITED. The book as a whole (compilation) cannot be reproduced, distributed or used for commercial or non-commercial purposes without INTECHOPEN LIMITED's written permission. Enquiries concerning the use of the book should be directed to INTECHOPEN LIMITED rights and permissions department (permissions@intechopen.com).

Violations are liable to prosecution under the governing Copyright Law.



Individual chapters of this publication are distributed under the terms of the Creative Commons Attribution 4.0 License which permits commercial use, distribution and reproduction of the individual chapters, provided the original author(s) and source publication are appropriately acknowledged. If so indicated, certain images may not be included under the Creative Commons license. In such cases users will need to obtain permission from the license holder to reproduce the material. More details and guidelines concerning content reuse and adaptation can be found at <http://www.intechopen.com/copyright-policy.html>.

Notice

Statements and opinions expressed in the chapters are those of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

First published in London, United Kingdom, 2025 by IntechOpen

IntechOpen is the global imprint of INTECHOPEN LIMITED, registered in England and Wales, registration number: 11086078, 167-169 Great Portland Street, London, W1W 5PF, United Kingdom

For EU product safety concerns: IN TECH d.o.o., Prolaz Marije Krucifikse Kozulić 3, 51000 Rijeka, Croatia, info@intechopen.com or visit our website at intechopen.com.

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

Time Series Analysis – Frontiers in Research and Practice

Edited by Jorge Rocha, Cláudia M. Viana, Inês Boavida-Portugal and Eduardo Gomes

p. cm.

Print ISBN 978-1-83635-505-2

Online ISBN 978-1-83635-504-5

eBook (PDF) ISBN 978-1-83635-506-9

If disposing of this product, please recycle the paper responsibly.

IntechOpen

intechopen.com

Built by scientists, for scientists



[Explore all IntechOpen books](#)

Meet the editors



Jorge Rocha holds an MSc in Geographic Information Systems (2003) and in Spatial Planning (2013), as well as a Ph.D. in Geographic Information Science (2012). He is currently an Associate Professor at the Institute of Geography and Spatial Planning and a member of the Modelling, Urban and Regional Planning, and Environmental Hazard and Risk Assessment and Management research groups at the Centre of Geographical Studies, all part of the University of Lisbon. His area of expertise includes Geosimulation and Geocomputation, involving Artificial Neural Networks, Graph Theory, Cellular Automata, and Multi-agent Systems. Jorge's work is quite diverse, focusing on, but not limited to, Urban Morphology, Remote Sensing, Land Use and Land Cover, Epidemiology, Health Geography, Geomarketing, Tourism, Smart Cities, and Big Data.



Cláudia M. Viana is a Geospatial Data Scientist and a Geographer. She earned her Ph.D. in Geographic Information Science (2022) and a Master's in Geographic Information Systems (2014). Currently, she is an Assistant Professor at the Centre for Geographical Studies and the Institute of Geography and Spatial Planning at the University of Lisbon, and a member of the Modelling and Spatial Planning research group. Over the past years, she has been a fellow researcher in several scientific projects related to changes in agriculture and the link between transport and land use at the Institute of Social Sciences (ULisboa) and the Faculty of Architecture (ULisboa). Her research focuses on spatiotemporal changes in agricultural land and their implications for food security and health.



Inês Boavida-Portugal is an Assistant Professor at the University of Lisbon, Institute of Geography and Spatial Planning (IGOT-ULisboa). Her recent work explores urban tourism patterns and spatial planning responses to guide cities towards future sustainable pathways. To examine these phenomena, Inês employs a mixed-method approach, including Geographical Information Systems and Information and Communication Technologies. She has particularly focused on the application of these methodologies to study how behavioral changes at the micro-level (within the governance triangle: civil society, governments, markets) can lead to critical shifts at the macro-level in urban systems, especially regarding tourism dynamics as a disruptive force.



Eduardo Gomes holds a Ph.D. in Geography from the University of Paris 1 Panthéon-Sorbonne and the University of Lisbon. His latest research focuses on the design of modeling and collaborative simulations used to analyze land-use and land-cover changes. He has been an invited lecturer at Beijing Forest University, the Institut des Systèmes Complexes de Paris, the Future Cities Laboratory, the Complexity Institute in Singapore, the

University of Copenhagen, and the Royal Melbourne Institute of Technology. Eduardo Gomes has received awards such as the Portuguese National Geography Award for the best Ph.D. thesis, the Best Scientific Article Award in Human Geography, and recognition for outstanding review contributions to scientific journals. Currently, he is an Assistant Professor at the University of Lisbon.

Contents

Preface	XI
Section 1	
Introduction	1
Chapter 1	3
Introductory Chapter: Frontiers in Research and Practice of Time Series Analysis	
<i>by Cláudia M. Viana, Inês Boavida-Portugal, Eduardo Gomes and Jorge Rocha</i>	
Section 2	
Probabilistic and Dynamical Foundations	15
Chapter 2	17
Data-Model-Driven Time Series Prediction	
<i>by Xiaoxiang Guo, Zhen Li, Jingrun Zhang and Chongxiao Zheng</i>	
Section 3	
Graphs, Dependencies, and Networked Time Series	43
Chapter 3	45
Integration of Time Series and Multiplex Networks Series in Cybersecurity and Prediction: An Innovative Approach	
<i>by Sergio Iglesias Pérez and Regino Criado Herrero</i>	
Section 4	
Efficient Learning and Domain Applications	61
Chapter 4	63
Neotectonic Landforms of Southwest Japan Restored by Numerical Modeling: Significance of Lateral Time-Progressive Motions on the Median Tectonic Line	
<i>by Yasuto Itoh, Shigekazu Kusumoto and Hirokazu Maekawa</i>	

Chapter 5	81
Neuro-Symbolic Methods for Times Series Analysis for Edge Computing	
<i>by Alberto Pacheco, Mariano Rivera, Ever Flores, Raymundo Torres,</i>	
<i>Raul Chacon and Jose Duran</i>	
Chapter 6	101
Time Series Prediction Based on Data Compression Methods	
<i>by Alexander Lysyak</i>	

Preface

Time series serve as the language describing dynamic phenomena. From climate to cybersecurity, finance, and geosciences, contemporary challenges increasingly rely on our ability to model temporal dependencies, extract insights from limited and noisy data, and implement methods tailored to real-world conditions. This book consolidates advances that extend beyond traditional approaches, integrating probabilistic modeling, network-based dependency representations, and computationally efficient learning suited for edge deployment, all emphasizing transparent and practical application.

Besides the Introduction section, the volume is organized into three other sections that reflect this arc.

Section 2, **Probabilistic and Dynamical Foundations**, lays out the methodological groundwork. In this section, the book connects data-driven and model-based methods, using Takens' embedding theorem and dynamical systems concepts to support hybrid predictors that merge mechanistic models with contemporary learning algorithms. The goal is not just to list techniques but to demonstrate how assumptions, diagnostics, and uncertainty interplay to produce robust and justifiable inferences.

Section 3, **Graphs, Dependencies, and Networked Time Series**, reinterprets temporal data as interconnected systems. The chapter uses a cybersecurity case study to make regime structures and cross-series influences clear. In this work, signals from diverse sources are rapidly fused, compressed, and processed. Throughout, the emphasis is on interpretable representation learning and forecasting that accounts for evolving dependency regimes.

Section 4, **Efficient Learning and Domain Applications**, focuses on methods designed for limited environments and scientific discovery in specific contexts. It introduces neuro-symbolic techniques suitable for real-time and edge computing, integrating spectral feature extraction, symbolization, and convolutional architectures for reliable recognition and prediction. Additionally, it discusses compression-based forecasting and universal coding as practical, transparent, and easy-to-deploy baselines that are competitive. The section concludes with a case study on restoring neotectonic landforms along the Median Tectonic Line in southwest Japan through numerical dislocation modelling, demonstrating how structured time-series analysis can support geological interpretation, especially when data are scarce and spatially interconnected.

The book targets researchers and practitioners familiar with time-dependent data who want to expand their toolkits, as well as postgraduate students seeking a clear progression from classical techniques to modern frontiers. It assumes a working knowledge of probability and linear algebra, along with basic experience in statistical computing (such as Python or R). When a chapter relies on specialised methods, we offer brief primers and direct readers to accessible references.

The sections can be read sequentially, but each chapter is independent of the others. Readers primarily interested in deployment can skip directly to Section 4 and revisit Section 2 for Bayesian calibration or Section 3 for dependency modelling as needed. Those examining networked or multi-stream data might start with Section 3, referencing Section 2 for uncertainty quantification and Section 4 for implementation templates. Throughout, we focus on reproducible workflows, transparent model specifications, and honest communication of uncertainty.

Throughout the chapters, several principles recur:

- **Purposeful Modelling:** Methods are introduced to address specific questions, such as forecasting during regime shifts, attributing influence among co-evolving series, or functioning within tight computational limits, rather than for their own sake.
- **Default uncertainty:** Forecasts and classifications are provided with intervals or probability estimates, and model critique is emphasized through posterior predictive checks, stability analyses, and sensitivity to priors or hyperparameters.
- **Transparent structure:** Whether via Bayesian priors, dynamic constraints, or graph models, we prioritise models that reveal assumptions and mechanisms, allowing review by domain experts.
- **Practical computation:** Algorithms are chosen considering memory, latency, and energy limits, in addition to accuracy, ensuring methods are applicable in real-world decision-making contexts.

We hope that *Time Series Analysis – Frontiers in Research and Practice* helps you link principled modelling with the complex realities of modern data, enabling you to create analyses that are accurate, robust, interpretable, and practical.

Jorge Rocha, Inês Boavida-Portugal, Eduardo Gomes and Cláudia M. Viana
Institute of Geography and Spatial Planning,
University of Lisbon,
Lisbon, Portugal

Section 1

Introduction

Introductory Chapter: Frontiers in Research and Practice of Time Series Analysis

Cláudia M. Viana, Inês Boavida-Portugal, Eduardo Gomes and Jorge Rocha

1. Introduction

The foundational assumption of time series analysis is the existence of a persistent, time-dependent underlying system that has shaped data in the past and is likely to continue doing so in the future. This system typically operates by establishing non-random patterns that can be detected over a time-series visual analysis or *via* some other statistical analysis. Therefore, the main objective of time series modeling is to detect such non-random patterns in the temporal behavior of a variable of interest, observing if past dynamics can allow future predictions, and thereby support decision-making processes.

In recent years, interest in time series analysis has grown considerably. This trend reflects the explosion of data availability (Big Data), the advances in computing capabilities, both in processing and storage (Data Lakes), and the emergence of sophisticated statistical and machine learning modeling techniques. In this chapter, we provide an outline of the current state of time series research, focusing on key applications, methodological developments, emerging algorithms, and ongoing challenges. We also discuss how classical statistical methods are being extended, and in some cases refined, by recent advances in artificial intelligence and, more specifically, machine learning methods.

2. Key applications

Time series analysis is widely used in various fields of knowledge, such as finance, economics, healthcare, environmental sciences, and industry, just to name a few.

In the fields of finance and economics, where these analytical methods have a long tradition, time series models are essential for understanding fluctuations in product (or asset) prices, interest and exchange rates, and market indices. They are a fundamental tool for predicting and managing risks and optimizing investment portfolios. Furthermore, time series analysis helps predict the behavior (e.g., scenario building) of important economic indicators, such as Gross Domestic Product (GDP), Gross Value Added (GVA), inflation, and unemployment, providing important data for political and economic planning [1, 2].

In this context, as we will discuss in the following section, classic models such as Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA), and Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX) continue to be widely used to capture linear trends and seasonality in data [3–5]. However, the often volatile and nonlinear nature of financial markets has led to the progressive adoption of machine learning approaches over classical methods. Hybrid approaches, such as the combination of ARIMA and Long Short-Term Memory (LSTM) artificial networks, are also becoming widely used due to their ability to leverage both statistical rigor (ARIMA) and pattern recognition in complex data (LSTM). These models have proven particularly effective in predicting stock prices and financial behaviors [1, 3, 6, 7].

In healthcare, time series analysis supports applications in fields as diverse as personalized medicine, remote monitoring, and epidemiology. From a more clinical perspective, studies demonstrate that data such as ECG signals, glucose levels, or blood pressure measurements can be used to monitor patient health in real time and anticipate adverse events, such as cardiac arrests or hypoglycemic episodes [8]. Furthermore, continuous patient monitoring based on time series allows for the early detection of critical conditions, improving clinical outcomes [9, 10]. Machine learning, particularly LSTM artificial networks and attention-based models, has shown promise in modeling complex and noisy physiological data (i.e., unwanted and meaningful data that blur existing patterns and relationships).

From a public health perspective, often even within a geographic context, epidemiological models based on historical data have been shown to aid in predicting disease outbreaks, enabling better preparedness and resource allocation [8, 11]. For example, during the COVID-19 pandemic, these models were essential in projecting infection curves and guiding government responses [12–14].

In environmental sciences, time series analysis is indispensable for identifying weather patterns, anticipating natural disasters, and monitoring ecological changes [15]. The analysis of data series such as temperature, precipitation, humidity, and wind is essential for weather prediction (short-term series) and climate studies (long-term series, e.g., 30-year climatological normals) [16]. Remote sensing data can also be an asset, as they provide high-temporal resolution information on vegetation (e.g., NDVI), temperature, and hydrology (e.g., NDWI), among other things. This information helps monitor droughts, predict floods, and inform agricultural planning decisions [11, 17–20]. Time series are also essential for predicting pollution levels and emissions, serving as a basis for environmental planning policies [2]. A clear example that demonstrates the importance of a holistic approach is the Intergovernmental Panel on Climate Change (IPCC) scenarios, which predict different scenarios for economic growth and greenhouse gas emissions.

Lastly, in industry and energy, time series analysis is primarily used for anomaly detection and process optimization. For example, by analyzing production metrics over time, companies can identify system failures and improve operational efficiency [21–23]. In the energy sector, forecasting energy consumption is essential for managing the grid itself, while forecasting the availability of renewable energies, such as solar and wind, can serve to support their integration into energy systems efficiently [20, 24].

Time series analysis also supports telecommunications, helping to predict network traffic for better bandwidth allocation [7]. In e-commerce and digital platforms, massive volumes of temporal data (e.g., clickstreams, transaction logs, and server metrics) are analyzed to forecast demand, detect irregularities, and fine-tune system

resources [25]. For instance, server load prediction helps manage infrastructure dynamically, and analysis of consumer behavior over time informs stock management and marketing strategies. Clustering and similarity analysis of time series also enhance recommender systems and user profiling.

Finally, in retail and business, sales forecasting based on time series data informs inventory and supply chain decisions, while also supporting customer analytics and targeted marketing [9, 19]. Across all these domains, time series analysis enables smarter, data-driven decision-making and improved responsiveness to real-world dynamics.

3. Methods

Time series analysis relies on a broad array of methods, ranging from traditional statistical models to modern machine learning and deep learning techniques (**Figure 1**). These techniques are designed to highlight temporal dependencies, capture underlying data structures, and predict future values based on past observations (forecasting) or vice versa (backcasting).

Despite recent developments, classical statistical models continue to be widely used, mainly due to their interpretability, simplicity, and solid theoretical foundations. Among the best known (and most widely used) are autoregressive (AR) models, which assume that current values depend linearly on past observations [11, 26]. Based on this assumption, the ARIMA model incorporates a differencing component to achieve stationarity and includes a component that uses a moving average to account for residual error [11, 27, 28]. Because time series often exhibit seasonal behavior, ARIMA evolved into SARIMA, which allows modeling periodic structures [27], and later, into SARIMAX, which allows the introduction of exogenous variables to improve predictive performance [4].

The added value that the moving average brought to ARIMA led to the development of another important group of algorithms that use moving averages to smooth out fluctuations in observations and highlight long-term trends. Notable in this

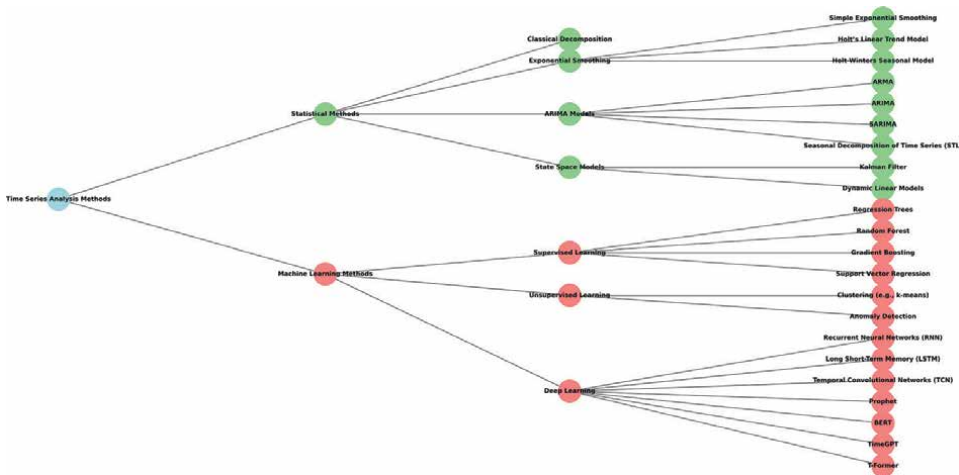


Figure 1.
Time series analysis methods.

context are the Simple Moving Average (SMA), the Weighted Moving Average (WMA), and the Exponential Moving Average (EMA). The former calculates the unweighted average using a fixed window of past values [26, 28, 29], and the second improves on the former's approach by assigning greater weight to more recent observations [28, 29]. Finally, the latter applies exponentially decreasing weights to older data, making the model more sensitive to recent changes [29].

The basic characteristics of the EMA directly relate it to the Exponential Smoothing (EMS) framework, which also places greater emphasis on more recent data to generate predictions. However, simple exponential smoothing only works well for stationary data series without trends or seasonality [11, 26], making it difficult to address both phenomena simultaneously. In contrast, the Holt-Winters method considers both, making it suitable for more complex time series [28].

Classical models tend to perform well on stable, linear data sets and are computationally efficient. However, these algorithms face limitations when faced with nonlinear dynamics, multiple variables, or long-term dependencies. With the emergence of the Internet of Things (IoT) and Big Data, these limitations have become more apparent and led to the increasing use of machine learning techniques, which bring greater flexibility to time series analysis. In the field of ML models, Random Forests (RF), Gradient Boosting Machines (XGBoost), and Support Vector Machines (SVM) have stood out, being applied not only to prediction but also to feature selection, classification, and anomaly detection [30].

More recently, there has been a resurgence in the use of artificial neural networks, which have demonstrated strong performance in modeling sequential data. This resurgence has not been primarily based on traditional structures such as multilayer perceptron (MLP), although other architectures, such as Radial Basis Function (RBF) networks, have been used to improve predictions in systems with complex and nonlinear behavior [20].

The "second life" of ANNs has been based primarily on the concept of enhanced learning (i.e., Deep Learning). Thus, models such as Recurrent Neural Networks (RNN), particularly those with LSTM architectures, showed greater prominence, as they are specially designed to retain memory for longer periods and are suitable for large, nonlinear time series [11, 20]. Temporal Convolutional Networks (TCN) have proven to be a valid alternative to RNNs, starting from the convolutional modeling paradigm and using causal filters to preserve temporal order and capture long-term patterns in the data. Similarly, the advent of Big Data, and consequently, social media data, has led to transformer-based models entering the realm of time series analysis. Originally developed for natural language processing (NLP), these models use self-attention mechanisms to model dependencies in entire sequences. Their adaptability has made them particularly effective in prediction, classification, and anomaly detection.

Finally, a promising research focus is the development of hybrid models that combine classical statistical approaches with neural networks. For example, time series can first be decomposed into trend, seasonal, and residual components, using STL (seasonal-trend decomposition based on LOESS) [11]. Then these three components can be modeled separately, frequently with distinct neural network subdivisions fitted to each signal type. In parallel, a new generation of foundation models (FMs) is emerging in the time series field. Inspired by language models like Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT), these large-scale pre-trained architectures are adapted for temporal data. Trained on massive time series datasets, models like TimeGPT [31]

and T-FORMER [32] develop a kind of “temporal intelligence” enabling zero-shot and few-shot forecasting, anomaly detection, and even domain adaptation. By leveraging transfer learning and attention mechanisms, they achieve state-of-the-art results across diverse applications [33].

All these methods depend on careful preliminary analysis. Visualization and exploratory techniques, such as plotting autocorrelation and partial autocorrelation functions, or applying smoothing filters, are essential for understanding the structure of a time series before applying complex models [34]. In addition, frequency-domain methods, such as the Fourier Transform, help detect cyclic components by transforming data from the time domain. For systems where observations are noisy or incomplete, the Kalman Filter provides a recursive method for estimating the underlying state in real time [29].

4. Challenges and future directions

Despite its many strengths, time series analysis continues to face a range of evolving challenges. These can be broadly grouped into four areas: data complexity, model limitations, computational demands, and ethical concerns. One of the most pressing difficulties lies in the increasing complexity of modern time series data. Today’s datasets are often high-volume, nonlinear, and dynamically changing, all characteristics that make accurate and generalizable forecasting particularly hard to achieve [10, 35, 36]. While classical statistical models remain valuable for many applications, they often fall short in capturing the layered, often non-stationary dependencies found in real-world data.

Another ongoing concern (or even constraint) is model generalization and computational complexity. Regarding the former, in applications where accurate predictions have strong real-world effects, several models are still unable to deliver reliable results [35, 37, 38]. Data quality—the old modeling saying about “garbage in, garbage out”—also muddles this situation. Indeed, outliers, noise, and inconsistencies can severely impact model performance, turning these issues into a current research topic [10, 39]. In high-dimensional contexts, e.g., sensor networks or genomics, dimensionality reduction becomes crucial. Techniques such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), and t-SNE help project complex datasets into lower-dimensional spaces while preserving key structures [15]. Similarly, feature selection methods, including variable importance measures from RF, prove to be effective in isolating the most relevant inputs.

As for the latter, Deep learning models, especially those built on transformer architectures, demand significant resources to train and deploy [36, 40]. In addition, model tuning and architecture selection are often time-consuming and sensitive to configuration choices, impacting performance [10, 37]. As models grow in sophistication, interpretability becomes even more crucial, especially in regulated environments such as healthcare or finance. Tools like SHapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME), originally developed for other types of data, are now being adapted to help explain time series predictions [41–43].

Anomaly detection is a particularly important application area, especially in settings where rare events carry serious consequences. In finance, anomalies might indicate fraud; in healthcare, they could signal acute medical risk [44]. Classical detection methods (based on fixed thresholds or statistical rules) often fail under

non-stationary conditions. Recent methods adopt probabilistic approaches, autoencoders, or Bayesian networks to learn normal behavior patterns and flag deviations more reliably. Newer frameworks even estimate anomalies as latent variables, eliminating the need for explicitly labeled, clean training data and improving performance in noisy environments.

Advances in representation learning, particularly *via* autoencoders and learned embeddings, also show promise. These methods enable the compression and abstraction of high-dimensional time series into interpretable formats, aiding visualization and improving anomaly diagnosis across multiple variables.

The rise of Big Data has shifted time series analysis from batch-based modeling toward real-time processing and decision-making. Continuous data streams from sensors, markets, or digital platforms require scalable and distributed computing solutions. Technologies like Apache Spark, Flink, and Kafka now underpin many real-time pipelines [45], while cloud-native and serverless architectures support the deployment of models that learn and adapt on the fly.

In response, algorithms have evolved. Scalable versions of ARIMA, Prophet, and LSTM can now be applied in real-time contexts, and online learning techniques allow models to update incrementally with each new data point, avoiding full retraining. This adaptability is crucial for responding to shifting patterns in dynamic systems.

Integrating models into production environments has also become a priority. Organizations increasingly seek end-to-end pipelines covering data ingestion, preprocessing, training, validation, deployment, and monitoring. In this context, MLOps platforms like MLflow, Kubeflow, and TensorFlow Extended (TFX) are being extended to support time series workflows, improving reproducibility, version control, and scalability [46].

As predictive models are embedded into decision-making processes, ethical considerations become unavoidable. Models must be transparent, fair, and accountable across their entire lifecycle. Future developments in time series analysis will need to incorporate fairness-aware design, privacy-preserving methods, and mechanisms for active stakeholder involvement, ensuring that models not only perform well but also align with societal values and minimize unintended consequences.

Looking forward, several promising directions are emerging. Ensemble methods, including bagging, boosting, and stacking, continue to show gains in robustness and predictive performance [10]. Deep learning architectures, especially those based on transformers, are being refined to reduce computational load while maintaining modeling capacity [36, 40]. At the same time, improved techniques for handling noise and missing data are essential for increasing model resilience [39].

Another growing area of focus is the integration of domain expertise into model architecture and training. In fields like healthcare and finance, such approaches lead to more relevant, interpretable, and actionable models [8, 47]. Finally, ongoing progress in algorithm efficiency and infrastructure scalability will be critical to meeting the demands of increasingly large and complex time series datasets [48].

5. Conclusion

Time series analysis continues to evolve at the intersection of well-established theory and cutting-edge innovation. From classical statistical models to deep learning architectures and emerging foundation models, the field has expanded both in depth and in breadth. Its growing range of applications, across domains as varied as finance,

healthcare, environmental science, and industry, reflects its central role in interpreting temporal data and supporting informed decision-making.

As new methods emerge and data sources multiply, the ability to harness time series analysis responsibly and effectively will depend not only on technical progress but also on how well researchers and practitioners respond to pressing challenges, including those related to complexity, interpretability, scalability, and ethics. Looking ahead, combining methodological rigor with ethical awareness and domain-specific insight will be key to fully realizing the potential of time series analysis in shaping a more adaptive, transparent, and data-informed future.

Acknowledgements

We acknowledge the GEOMODLAB (Laboratory for Remote Sensing, Geographical Analysis and Modeling) of the Centre of Geographical Studies/Institute of Geography and Spatial Planning for providing computational infrastructure support for this research. This work has been developed under project ML-Soil, Co-participatory Modeling of Soil Districts using Machine Learning (2024.00178. S4P24), within Science4Policy 2024 (S4P-24), an annual Science for Policy Project call, an initiative promoted by Centre for Planning and Evaluation of Public Policies in partnership with the Foundation for Science and Technology (FCT), financed by Portugal Recovery and Resilience Plan and under the framework of the CEG Research Unit UIDB/00295/2020 and UIDP/00295/2020. CMV is working under the Stimulus of Scientific Employment – Individual Support (5th edition), funded by FCT (2022.08734.CEECIND).

Conflict of interest

The authors declare no conflict of interest.

Author details

Cláudia M. Viana, Inês Boavida-Portugal, Eduardo Gomes and Jorge Rocha*
Centre of Geographical Studies, Institute of Geography and Spatial Planning,
University of Lisbon, Lisbon, Portugal and Associate Laboratory Terra, Lisbon,
Portugal

*Address all correspondence to: jorge.rocha@edu.ulisboa.pt

IntechOpen

© 2025 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Patel S, Sanghavi D, Singh A. Forecasting economic trends: Analyzing FRED data with time series methods and RNNs. In: 2023 Global Conference on Information Technologies and Communications (GCITC); Bangalore, India. 2023. pp. 1-7. DOI: 10.1109/GCITC60406.2023.10426363
- [2] Liu W, Yu X, Zhao Q, Cheng G, Hou X, He S. Time series forecasting fusion network model based on prophet and improved LSTM. *Computers, Materials and Continua*. 2023;**74**(2):3199-3219. DOI: 10.32604/cmc.2023.032595
- [3] Lohar K, Pahwa P, Asthana N, Gautam D. Stock market analysis using time series with ARIMA model. In: 2024 2nd International Conference on Computer, Communication and Control (IC4); Indore, India. 2024. pp. 1-5. DOI: 10.1109/IC457434.2024.10486268
- [4] Ram CS, Raj M, Chaturvedi RK. Boosting time-series forecasting accuracy with SARIMAX seasonal interval automation. In: *Procedia Computer Science (Seventh International Conference on Recent Trends in Image Processing and Pattern Recognition (RTIP2R-2024))*. Vol. 260. 2025. pp. 814-821. ISSN 1877-0509. DOI: 10.1016/j.procs.2025.03.262
- [5] Zhang M. Financial time series frequent pattern mining algorithm based on time series ARIMA model. In: 2023 International Conference on Networking, Informatics and Computing (ICNETIC); Palermo, Italy. 2023. pp. 244-247. DOI: 10.1109/ICNETIC59568.2023.00057
- [6] Masih B. Recurrent neural network model for time series analysis. In: *CEUR Workshop Proceedings*. Vol. 3885. CEUR-WS; 2024. pp. 485-498
- [7] Ho R, Hung K. CEEMD-based multivariate financial time series forecasting using a temporal fusion transformer. In: 2024 IEEE 14th Symposium on Computer Applications & Industrial Electronics (ISCAIE); Penang, Malaysia. 2024. pp. 209-215. DOI: 10.1109/ISCAIE61308.2024.10576340
- [8] León-Álvarez AL, Betancur-Gómez JI, Jaimes-Barragán F, Grisales-Romero H. Clinical and epidemiological rounds. *Time Series. Iatreia*. 2016;**29**(3):373-381. DOI: 10.17533/udea.iatreia.v29n3a12
- [9] Khan ABE, Kamalakannan K, Ahmed NSS. Integrating machine learning and stochastic pattern analysis for the forecasting of time-series data. *SN Computer Science*. 2023;**4**:484. DOI: 10.1007/s42979-023-01981-0
- [10] Sakib M, Mustajab S, Alam M. Ensemble deep learning techniques for time series analysis: A comprehensive review, applications, open issues, challenges, and future directions. *Cluster Computing*. 2025;**28**:73. DOI: 10.1007/s10586-024-04684-0
- [11] Abhishek Khullar V. Exploring techniques and approaches for time series analysis and its attacks. In: 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT); Kamand, India. 2024. pp. 1-6. DOI: 10.1109/ICCCNT61001.2024.10725845
- [12] Silva M, Viana CM, Betco I, Nogueira P, Roquette R, Rocha J. Spatiotemporal dynamics of epidemiology diseases: Mobility based

risk and short-term prediction modeling of COVID-19. *Frontiers in Public Health*. 2024;**12**:1359167. DOI: 10.3389/fpubh.2024.1359167. Available from: <https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2024.1359167>

[13] Barbosa B, Silva M, Capinha C, Garcia RAC, Rocha J. Spatial correlates of COVID-19 first wave across continental Portugal. *Geospatial Health*. 23 Jun 2022;**17**(s1). DOI: 10.4081/gh.2022.1073

[14] Silva M, Betco I, Capinha C, Roquette R, Viana CM, Rocha J. Spatiotemporal dynamics of COVID-19 infections in mainland Portugal. *Sustainability*. 2022;**14**:10370. DOI: 10.3390/su141610370

[15] Rekha MP, Perumal K. A hybrid ANN random forest regression method of feature extraction in time series data. *SN Computer Science*. 2025;**6**:23. DOI: 10.1007/s42979-024-03530-9

[16] Trindade J, Rocha J, Santos PP, et al. Methodology for sea level rise hazard assessment in the exposed coastal zone of the Portuguese mainland. *Geogaceta*. 2023;**74**:59-62. DOI: 10.55407/geogaceta98169

[17] Viana CM, Girão I, Rocha J. Long-term satellite image time-series for land use/land cover change detection using refined open source data in a rural region. *Remote Sensing*. 2019;**11**:1104. DOI: 10.3390/rs11091104

[18] Viana CM, Freire D, Abrantes P, Rocha J. Evolution of agricultural production in Portugal during 1850-2018: A geographical and historical perspective. *Land*. 2021;**10**:776. DOI: 10.3390/land10080776

[19] Sonata I, Heryadi Y. Comparison of LSTM and transformer for time

series data forecasting. In: 2024 7th International Conference on Informatics and Computational Sciences (ICICoS); Semarang, Indonesia. 2024. pp. 491-495. DOI: 10.1109/ICICoS62600.2024.10636892

[20] Zheng K, Wang J, Chen Y, Jiang R, Wang W. DDTCN: Decomposed dimension time-domain convolutional neural network along spatial dimensions for multiple long-term series forecasting. *Applied Intelligence*. 2024;**54**(8):6606-6623. DOI: 10.1007/s10489-024-05526-7

[21] Moon J, Oh S, Kum S. An introduction to the architecture of CLUST platform supporting multiple time series data integration and analysis. In: 2022 13th International Conference on Information and Communication Technology Convergence (ICTC); Jeju Island, Korea, Republic of. 2022. pp. 507-509. DOI: 10.1109/ICTC55196.2022.9952914

[22] Kugiumtzis D, Tsimpiris A. Measures of analysis of time series (MATS): A MATLAB toolkit for computation of multiple measures on time series data bases. *Journal of Statistical Software*. 2010;**33**(5):1-30. DOI: 10.18637/jss.v033.i05

[23] Struckov A, Yufa S, Visheratin AA, Nasonov D. Evaluation of modern tools and techniques for storing time-series data. *Procedia Computer Science*. 2019;**156**:19-28. DOI: 10.1016/j.procs.2019.08.125

[24] Arslan S. A hybrid forecasting model using LSTM and prophet for energy consumption with decomposition of time series data. *PeerJ Computer Science*. 2022;**8**:e1001. DOI: 10.7717/PEERJ-CS.1001

[25] Jin M, Jin W, Xue H, Salim F, Liang Y. Time series analysis in the web:

- Recent advances and future trends. In: Companion Proceedings of the ACM on Web Conference 2025 (WWW '25). New York, NY, USA: Association for Computing Machinery; 2025. pp. 29-32. DOI: 10.1145/3701716.3715853
- [26] Conklin JD. When your process has runs trends and cycles. *Quality Progress*. 2005;38(3):64-69
- [27] Benghalia A, Benkenzoua H, Hamraras C, Boufenar C. Long-term forecasting of hydrocarbon products sales. In: 2024 1st International Conference on Electrical, Computer, Telecommunication and Energy Technologies (ECTE-Tech); Oum El Bouaghi, Algeria. 2024. pp. 1-6. DOI: 10.1109/ECTE-TECH62477.2024.10851084
- [28] Zhao Y. Analysis and prediction of fuel oil in a terminal. In: Tiwari S, Trivedi M, Mishra K, Misra A, Kumar K, Suryani E, editors. *Smart Innovations in Communication and Computational Sciences*. Advances in Intelligent Systems and Computing. Vol. 1168. Singapore: Springer; 2021. DOI: 10.1007/978-981-15-5345-5_20
- [29] Lotysh V, Gumeniuk L, Humeniuk P. Comparison of the effectiveness of time series analysis methods: SMA, WMA, EMA, EWMA, and Kalman filter for data analysis. *Informatyka, Automatyka, Pomiary W Gospodarce I Ochronie Środowiska*. 2023;13(3):71-74. DOI: 10.35784/iapgos.3652
- [30] Hansen JV, McDonald JB, Nelson RD. Some evidence on forecasting time-series with support vector machines. *The Journal of the Operational Research Society*. 2006;57(9):1053-1063. DOI: 10.1057/palgrave.jors.2602073
- [31] Liao W, Wang S, Yang D, et al. TimeGPT in load forecasting: A large time series model perspective. *Applied Energy*. 2025;379:124973. DOI: 10.1016/j.apenergy.2024.124973
- [32] Lim B, Arık SÖ, Loeff N, Pfister T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*. 2021;37(4):1748-1764. DOI: 10.1016/j.ijforecast.2021.03.012
- [33] Liang Y, Wen H, Nie Y, et al. Foundation models for time series analysis: A tutorial and survey. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24). New York, NY, USA: Association for Computing Machinery; 2024. pp. 6555-6565. DOI: 10.1145/3637528.3671451
- [34] Chandler RE, Scott EM. Exploratory analysis. In: Chandler RE, Scott EM, editors. *Statistical Methods for Trend Detection and Analysis in the Environmental Sciences*. 2011. DOI: 10.1002/9781119991571.ch2
- [35] Liu Z, Zhu Z, Gao J, Xu C. Forecast methods for time series data: A survey. *IEEE Access*. 2021;9:91896-91912. DOI: 10.1109/ACCESS.2021.3091162
- [36] Wang D. A comprehensive study on transformer-based time series forecasting. In: Mzili T, Arya AK, editors. *Innovations in Optimization and Machine Learning*. Hershey, PA: IGI Global Scientific Publishing; 2025. pp. 159-180. DOI: 10.4018/979-8-3693-5231-1.ch007
- [37] Abbas JK, Al-Azzawi W, Hassan SI, Sharif SFA, Manee MJ. Deep learning techniques for the evaluation of a financial time series forecasting model. In: 2024 International Conference on Emerging Research in Computational Science (ICERCS); Coimbatore,

India. 2024. pp. 1-6. DOI: 10.1109/ICERCS63125.2024.10895447

[38] Yousaf H, Munir K, Hassan A, Haris M, Saleem Bokhari SA, Ullah K. Time series and machine learning methods for short-term load forecasting in modern power systems. In: 2024 International Conference on Electrical, Communication and Computer Engineering (ICECCE); Kuala Lumpur, Malaysia. 2024. pp. 1-6. DOI: 10.1109/ICECCE63537.2024.10823556

[39] Zhang L. Research on the application of functional data regression analysis in time series forecasting. In: 2024 IEEE 2nd International Conference on Image Processing and Computer Applications (ICIPCA); Shenyang, China. 2024. pp. 600-605. DOI: 10.1109/ICIPCA61593.2024.10709041

[40] Ang JS, Ng KW, Chua FF. Modeling time series data with deep learning: A review, analysis, evaluation and future trend. In: 2020 8th International Conference on Information Technology and Multimedia (ICIMU); Selangor, Malaysia. 2020. pp. 32-37. DOI: 10.1109/ICIMU49871.2020.9243546

[41] Viana CM, Santos M, Freire D, Abrantes P, Rocha J. Evaluation of the factors explaining the use of agricultural land: A machine learning and model-agnostic approach. *Ecological Indicators*. 2021;**131**:108200. DOI: 10.1016/j.ecolind.2021.108200

[42] Betco I, Ribeiro AI, Vale DS, Encalada-Abarca L, Viana CM, Rocha J. Sentiment analysis using a lexicon-based approach in Lisbon, Portugal. *Geospatial Health*. 2025;**20**(1SE-Gnosis(GIS) Symposium):1344. DOI: 10.4081/gh.2025.1344

[43] Tong Y, Liu J, Yu L, et al. Technology investigation on time

series classification and prediction. *PeerJ Computer Science*. 2022;**8**:e982. DOI: 10.7717/PEERJ-CS.982

[44] Choudhary S, Hiranandani G, Kumar SS. Sparse decomposition for time series forecasting and anomaly detection. In: Proceedings of the 2018 SIAM International Conference on Data Mining (SDM). Society for Industrial and Applied Mathematics; 2018. pp. 522-530. DOI: 10.1137/1.9781611975321.59

[45] Sharma G, Bandhu KC. A survey on big data and time-series analysis. In: 15th International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT 2024. Vol. 2. Grenze International Journal of Engineering and Technology; 2024. pp. 3536-3544

[46] Harika A, Aravinda K, Nagpal A, Singh DV, Praveen, Abdulaali HS. Temporal data mining: Uncovering patterns in time-series data streams with machine learning. In: 2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON); Gautam Buddha Nagar, India. 2023. pp. 1539-1544. DOI: 10.1109/UPCON59197.2023.10434603

[47] Bleikh HY, Young WL. *Time Series Analysis and Adjustment: Measuring, Modelling and Forecasting for Business and Economics*. 1st ed. Routledge; 2016. DOI: 10.4324/9781315550954

[48] Palpanas T, Beckmann V. Report on the first and second interdisciplinary time series analysis workshop (ITISA). *SIGMOD Record*. 2019;**48**(3):36-40. DOI: 10.1145/3377391.3377400

Section 2

Probabilistic and Dynamical
Foundations

Data-Model-Driven Time Series Prediction

Xiaoxiang Guo, Zhen Li, Jingrun Zhang and Chongxiao Zheng

Abstract

In complex systems science, time series prediction serves as a core methodology for uncovering the dynamical laws of natural and social phenomena, which demonstrates significant applications across numerous disciplines such as finance, meteorology, and medicine. Current research in time series prediction primarily follows two paradigms: data-driven machine learning methods, while excelling in non-linear fitting capabilities, inherently suffer from limited interpretability; model-driven mechanistic analysis methods, despite offering strong theoretical interpretability, often struggle to achieve satisfactory data-fitting performance. To overcome this dilemma, a novel data-model hybrid-driven forecasting approach has emerged. The organic integration of neural networks' data representation advantages with mechanistic models' physical interpretability is catalyzing a paradigm shift in time series prediction research. This chapter will systematically review advances in time series prediction methods, with a focus on three cutting-edge directions: deep learning-based prediction algorithms, dynamics-inspired forecasting frameworks, and physics-informed neural network (PINN) approaches. The aim is to provide researchers with comprehensive and in-depth methodological guidance, facilitating breakthroughs in complex system modeling and prediction studies.

Keywords: time series prediction, data-model driven, machine learning, mathematical model, dynamical evolution

1. Introduction

Time series data plays a pivotal role in scientific research and engineering applications, serving as a fundamental medium for uncovering the dynamic evolution of complex systems [1, 2]. It is ubiquitous across diverse domains, including finance, meteorology, biomedicine, and industrial control, where accurate forecasting is essential for both understanding system dynamics and optimizing decision-making processes [3–9]. However, the underlying dynamical patterns are often obscured by seemingly chaotic observations, posing significant challenges for prediction tasks [10].

Traditional statistical models, such as autoregressive integrated moving average (ARIMA), exhibit strong performance in linear and stationary scenarios but struggle to capture the intricate non-linearities inherent in real-world systems [11]. Recent advancements in artificial intelligence and computational power have propelled data-driven approaches, particularly neural networks, to the forefront of time series

forecasting due to their exceptional capacity for modeling non-linear relationships [12–15]. Nevertheless, purely data-driven methods often lack interpretability and may fail to generalize under unseen conditions, as they do not explicitly account for the governing physical laws of the system [16, 17]. Consequently, a critical research focus has emerged: how to effectively integrate data-driven models with domain-specific physical principles to enhance prediction reliability, interpretability, and generalizability. This paradigm shift, bridging machine learning with mathematical modeling, holds the potential to unlock more robust and physically consistent forecasting frameworks, paving the way for breakthroughs in complex system analysis and control.

In recent years, deep neural network algorithms based on the universal approximation theorem, such as backpropagation neural networks (BPNNs), long short-term memory (LSTM) neural networks, and attention mechanism neural networks, have demonstrated exceptional performance in non-linear fitting tasks [12–15, 18–20]. Meanwhile, time series prediction methods derived from Takens embedding theorem and generalized embedding theorems, including multi-view embedding (MVE), randomly distributed embedding (RDE), delay parameterized method (DPM), and auto-reservoir neural network (ARNN), introduce the dynamical evolution information into prediction models, offering innovative solutions for predicting low-dimensional chaotic and high-dimensional short-range time series [21–24]. To address the limitations of the “black-box” nature, physics-informed neural networks have significantly enhanced model interpretability by integrating physical prior knowledge with neural networks, providing novel research perspectives for the modeling and prediction of complex systems [25, 26].

This chapter will systematically review the advances in time series prediction methods in three cutting-edge directions: deep learning-based prediction algorithms, dynamics-inspired forecasting frameworks, and physics-informed neural network approaches. This enables researchers to gain a deeper understanding of state-of-the-art time series forecasting methods. In Section 2, we will introduce deep learning-based prediction algorithms, including BPNN, LSTM, and reservoir computing (RC). In Section 3, we discuss the application of embedding theories in time series prediction and present state-of-the-art forecasting methods such as MVE, RDE, DPM, and ARNN. In Section 4, we elaborate on data-model-driven prediction approaches, including physics-informed neural networks (PINNs) and transmission dynamics-informed neural networks (TDINNs) [27]. Section 5 ends with a discussion on future research directions.

2. Deep learning-based prediction algorithms

Deep neural networks have demonstrated remarkable performance in time series analysis tasks, such as prediction, classification, and anomaly detection, owing to their powerful non-linear fitting capabilities. This section systematically introduces three representative models, BP neural networks, LSTM networks, and reservoir computing, with an in-depth discussion of their core principles and application scenarios. The presentation aims to provide readers with cutting-edge research insights and practical paradigms in time series analysis.

2.1 BP neural network

The BP neural network was formally proposed by D. Rumelhart, G. Hinton, and R. Williams in 1986 [28]. As a multi-layer feed-forward neural network, the core

principle of the BP network lies in optimizing the network weights and biases *via* back-propagation to minimize the mean squared error (MSE) between the predicted outputs and the actual outputs. The network architecture generally comprises an input layer, hidden layers, and an output layer, where neurons across layers are interconnected *via* weighted connections. The number of neurons in the input layer aligns with the dimensions of historical observations within the time series, while that in the output layer corresponds to the dimensions of the prediction target.

In a four-layer BP neural network (see the framework in **Figure 1**), let the input signal be $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$. Then, the i^{th} neuron in 1^{st} hidden layer satisfies,

$$h_i^1 = \sigma^1 \left(\sum_{p=1}^n \omega_{ip}^1 x_p + b_i^1 \right), \quad (1)$$

and the j^{th} neuron in 2^{nd} hidden layer obeys,

$$h_j^2 = \sigma^2 \left(\sum_{i=1}^{m_1} \omega_{ji}^2 h_i^1 + b_j^2 \right), \quad (2)$$

The output signal,

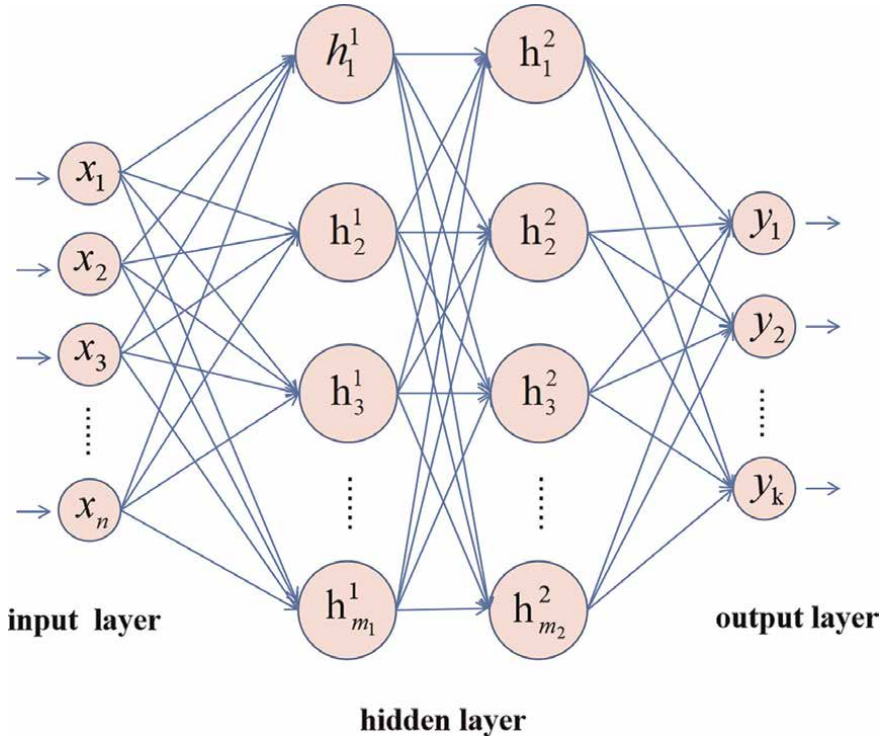


Figure 1.

The brief framework of a four-layer BP neural network. Input: $\{x_1, \dots, x_n\}$, 1^{st} hidden layer: $\{h_1^1, \dots, h_{m_1}^1\}$, 2^{nd} hidden layer: $\{h_1^2, \dots, h_{m_2}^2\}$, and output: $\{y_1, \dots, y_k\}$.

$$\begin{aligned}
 y_q &= \sigma^3 \left(\sum_{j=1}^{m_2} \omega_{qj}^3 h_j^2 + b_q^3 \right) \\
 &= \sigma^3 \left(\sum_{j=1}^{m_2} \omega_{qj}^3 \left(\sigma^2 \left(\sum_{i=1}^{m_1} \omega_{ji}^2 h_i^1 + b_j^2 \right) \right) + b_q^3 \right) \\
 &= \sigma^3 \left(\sum_{j=1}^{m_2} \omega_{qj}^3 \left(\sigma^2 \left(\sum_{i=1}^{m_1} \omega_{ji}^2 \left(\sigma^1 \left(\sum_{p=1}^n \omega_{ip}^1 x_p + b_i^1 \right) \right) + b_j^2 \right) \right) + b_q^3 \right).
 \end{aligned} \tag{3}$$

Here, σ_1 , σ_2 , and σ_3 are the activation functions; ω_{ip}^1 , ω_{ji}^2 , and ω_{qj}^3 are weight parameters; and b_i^1 , b_j^2 , and b_q^3 are the bias parameters. The loss function of the network is typically defined as,

$$\mathcal{L} = \frac{1}{2} \sum_{q=1}^k (y_q - \hat{y}_q)^2, \tag{4}$$

where $\{\hat{y}_q, q = 1, 2, \dots, k\}$ is the real output signal. To minimize the prediction error, the gradient descent algorithm is employed, which updates the network weights and biases by computing the derivative of the loss function with respect to the parameters, thereby optimizing the model. This is normally done using back-propagation. The iteration of the parameter v in the network is,

$$v \leftarrow v + \Delta v. \tag{5}$$

Taking the update of ω_{qj}^3 as an example, the sigmoid function is chosen as the activation function.

$$\begin{aligned}
 \Delta \omega_{qj}^3 &= -\eta \frac{\partial \mathcal{L}}{\partial \omega_{qj}^3} \\
 &= -\eta \left(\frac{\partial \mathcal{L}}{\partial y_q} \cdot \frac{\partial y_q}{\partial \omega_{qj}^3} \right) \\
 &= -\eta \left((y_q - \hat{y}_q) \cdot (1 - y_q) \cdot y_q \cdot h_j^2 \right)
 \end{aligned} \tag{6}$$

Similarly, the update formula of other parameters can be obtained.

2.2 LSTM neural network

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem in traditional RNNs, enabling better learning of long-term dependencies in sequential data [12]. The core architecture of LSTM comprises a memory cell and three gating mechanisms: the input gate, forget gate, and output gate. (See a brief framework in **Figure 2**.)

The forget gate decides what portion of the previous cell state C_{t-1} needs to be removed. This decision is computed using the current input x_t and the previous hidden state h_{t-1} , followed by a sigmoid activation to produce a forget gate output f_t ,

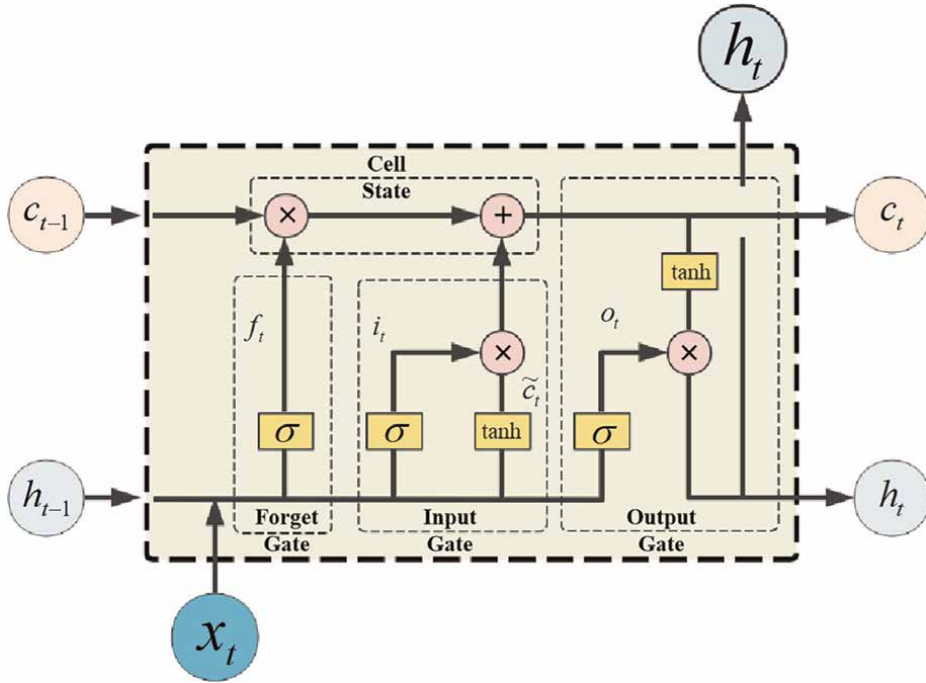


Figure 2.
 The brief framework of the LSTM neural network.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (7)$$

where σ denotes the sigmoid function and W_f and b_f represent the weight matrix and bias term of the forget gate, respectively. A value of f_t close to 0 means completely forget, while a value close to 1 means fully retain.

The input gate determines how much of the candidate information (derived from the current input x_t and previous hidden state h_{t-1}) should be added to the memory cell and produces a candidate memory cell state $\tilde{C}_t$ to propose new potential values. Mathematically, these operations are defined as,

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C). \end{aligned} \quad (8)$$

The final cell state update combines these terms with the output of the forget gate,

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (9)$$

The output gate regulates the quantity of information from the current memory cell state C_t that is relayed to the hidden state h_t at the current time step. This involves two steps: A sigmoid layer decides which parts of C_t to output, based on the current input x_t and previous hidden state h_{t-1} ; then the gate modulates the tanh-squashed version of C_t to produce h_t [29].

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t &= o_t \odot \tanh(C_t). \end{aligned} \quad (10)$$

In an LSTM, the sigmoid function acts as a switch in each gate, regulating the flow of information by outputting values between 0 and 1. The $\tanh$ function, which scales values to the range $[-1, 1]$, helps control the memory cell's state, ensuring balanced and stable information flow, and contributes to generating the hidden state.

Example: Predict the next number in a time series.

Given a one-dimensional sequence $X = \{x_1, x_2, x_3, \dots, x_n\}$, the goal is to train an LSTM model for sequence extrapolation. Specifically, the model should predict the $(n + 1)$ -th element given the preceding n consecutive elements. For $n = 3$, the model should output x_4 when the input sequence is $[x_1, x_2, x_3]$.

Assume the LSTM unit contains a single neuron, with an initial memory cell $C_0 = 0$ and an initial hidden state $h_0 = 0$. The detailed calculations for each time step are as follows (notation: σ symbolizes the sigmoid activation mapping, while $\tanh$ represents the hyperbolic tangent activation transformation).

- Time step $t = 1$: Input x_1

If,

$$\begin{aligned} f_1 &= \sigma(W_f \cdot [h_0, x_1] + b_f) = 0.8, \\ i_1 &= \sigma(W_i \cdot [h_0, x_1] + b_i) = 0.9, \\ \tilde{C}_1 &= \tanh(W_C \cdot [h_0, x_1] + b_C) = 1.0, \end{aligned}$$

which means 80% of the previous memory is retained and 90% of new information is written into the memory cell. So, the memory cell is updated,

$$C_1 = f_1 \cdot C_0 + i_1 \cdot \tilde{C}_1 = 0.8 \times 0 + 0.9 \times 1.0 = 0.9.$$

Suppose output gate $o_1 = \sigma(W_o \cdot [h_0, x_1] + b_o) = 0.7$, then the hidden state,

$$h_1 = o_1 \cdot \tanh(C_1) = 0.7 \times \tanh(0.9) \approx 0.5014.$$

- Time step $t = 2$: Input x_2

Suppose,

$$\begin{aligned} f_2 &= \sigma(W_f \cdot [h_1, x_2] + b_f) = 0.7, \\ i_2 &= \sigma(W_i \cdot [h_1, x_2] + b_i) = 0.8, \\ \tilde{C}_2 &= \tanh(W_C \cdot [h_1, x_2] + b_C) = 2.0, \\ o_2 &= \sigma(W_o \cdot [h_1, x_2] + b_o) = 0.6, \end{aligned}$$

then, the memory cell C_2 and the hidden state h_2 satisfy,

$$\begin{aligned} C_2 &= f_2 \cdot C_1 + i_2 \cdot \tilde{C}_2 = 0.7 \times 0.9 + 0.8 \times 2.0 = 2.23, \\ h_2 &= o_2 \cdot \tanh(C_2) = 0.6 \times \tanh(2.23) \approx 0.5863. \end{aligned}$$

- Time step $t = 3$: Input x_3

If,

$$\begin{aligned} f_3 &= \sigma(W_f \cdot [h_2, x_3] + b_f) = 0.75, \\ i_3 &= \sigma(W_i \cdot [h_2, x_3] + b_i) = 0.85, \\ \tilde{C}_3 &= \tanh(W_C \cdot [h_2, x_3] + b_C) = 3.0, \\ o_3 &= \sigma(W_o \cdot [h_2, x_3] + b_o) = 0.65, \end{aligned}$$

then, the memory cell C_3 and the hidden state h_3 satisfy,

$$\begin{aligned} C_3 &= f_3 \cdot C_2 + i_3 \cdot \tilde{C}_3 = 0.75 \times 2.23 + 0.85 \times 3.0 = 4.2225, \\ h_3 &= o_3 \cdot \tanh(C_3) = 0.65 \times \tanh(4.2225) \approx 0.6497. \end{aligned}$$

The final hidden state h_3 , which represents an understanding of x_1, x_2 , and x_3 , is mapped to predict x_4 via a fully connected layer, $\hat{x}_4 = W_{out}h_3 + b_{out}$.

This case demonstrates the application of LSTM in simple sequence prediction, highlighting how gating mechanisms dynamically manage temporal information flow. In practice, attention should be paid to hyper-parameter tuning (e.g., hidden layer size, learning rate), data preprocessing (e.g., normalization), and over-fitting control (e.g., adding drop-out layers) to enhance model generalization.

2.3 Reservoir computing

Reservoir computing (RC), also known as echo state network and liquid state machine, is a variant of recurrent neural networks [10, 30]. Traditional standard RNNs face the challenge of gradient vanishing or exploding when processing long sequences. Although improved models like LSTM and GRU alleviate this issue through gating mechanisms, they require extensive parameter training, leading to high computational costs. The core technical feature of RC lies in using a sparsely and randomly connected recurrent hidden layer (termed as a reservoir) for efficient time series processing. By fixing the random connection weights of the reservoir and only training the output layer weights, RC significantly reduces computational complexity while fully preserving the ability to model temporal dependencies.

An RC network consists of an input layer, a reservoir (hidden layer), and an output layer, with the reservoir serving as the core component [31]. See the brief framework of reservoir computing in **Figure 3**. The reservoir contains far more neurons than the input dimension ($N \gg M$), and the connections between neurons are random and sparse. Besides, the spectral radius of the reservoir weight matrix $\rho(W_{res}) < 1$ ensures the network operates in stable dynamics. The update process of RC can be decomposed into the following steps:

1. *From the input layer to the reservoir.* The input signals are mapped to the reservoir space via a fixed weight matrix.

$$\vec{u}(t) = W_{in}\vec{x}(t), \vec{x}(t) \in R^{M \times 1}, W_{in} \in R^{N \times M}, \vec{u}(t) \in R^{N \times 1}. \quad (11)$$

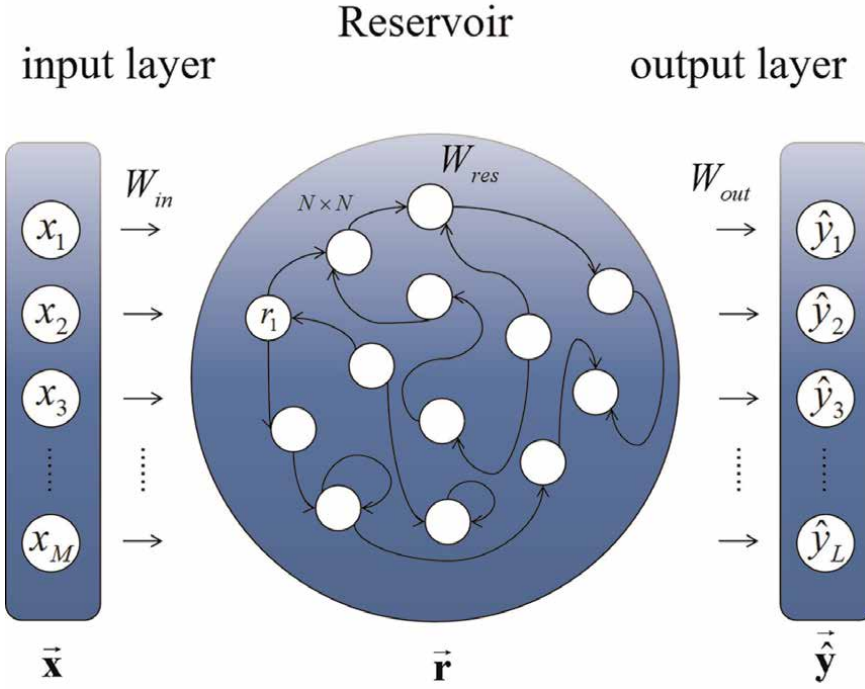


Figure 3.
The brief framework of reservoir computing.

2. *The update of the reservoir state.* The reservoir state is recursively updated using a leaky integrator model to balance the influence of historical states and current inputs.

$$\vec{r}(t) = (1 - \alpha) \vec{r}(t - 1) + \alpha \sigma(W_{res} \vec{r}(t - 1) + \vec{u}(t)). \quad (12)$$

Here, $W_{res} \in R^{N \times N}$ is the internal weight matrix of the reservoir, which is randomly generated and satisfies $\rho(W_{res}) < 1$. $\alpha \in (0, 1]$ is the leak rate that controls the state update speed [32]. $\sigma(\cdot)$ is a non-linear activation function enabling the reservoir to capture non-linear dynamics.

3. *From the input layer to the output layer.* The output signals are obtained by a linear map.

$$\vec{\hat{\mathbf{y}}}(t) = W_{out} \vec{r}(t), W_{out} \in R^{L \times N}, \vec{\hat{\mathbf{y}}}(t) \in R^{L \times 1}. \quad (13)$$

The loss function is $\mathcal{L} = \sum_t \left\| \vec{\hat{\mathbf{y}}}(t) - W_{out} \vec{r}(t) \right\|^2$..

Due to space constraints, for more excellent RC algorithms, please see the references [33–37].

3. Dynamics-inspired forecasting frameworks

Despite their high predictive accuracy, deep learning-based prediction methods are still plagued by the intrinsic interpretability issues of machine learning. Crucially, they overlook the dynamical evolution of the system, failing to exploit its inherent temporal/spatio-temporal dynamics. Embedding theory and phase space reconstruction methods enable the extraction of dynamical properties from indirect, time-varying observational data, offering a novel framework for time series prediction. This section begins with the mathematical foundations of Takens' theorem, explores phase space reconstruction techniques for time series, and highlights recent advances in noise-resistant and high-dimensional modeling to enhance the forecasting techniques in complex systems.

3.1 Phase space reconstruction

Phase space reconstruction is a process of reconstructing the system's phase space from a time series through the delay coordinates technique [38, 39]. For a time series $X = \{x(1), x(2), x(3), \dots, x(N)\}$, N is the length of the time series, its reconstructed phase space R is defined as:

$$R = \begin{pmatrix} x(1) & x(1 + \tau) & \cdots & x(1 + (m - 1)\tau) \\ x(2) & x(2 + \tau) & \cdots & x(2 + (m - 1)\tau) \\ \vdots & \vdots & \ddots & \vdots \\ x(N - (m - 1)\tau) & x(N - (m - 2)\tau) & \cdots & x(N) \end{pmatrix}, \quad (14)$$

where τ is the time delay and m is the embedding dimension. Each row of R is a m -dimensional delay coordinate vector.

The time delay (τ) and embedding dimension (m) are critical parameters in phase space reconstruction. If τ is too small, the delay coordinate vectors become nearly indistinguishable, obscuring the orbital evolution characteristics in phase space and complicating feature extraction. Conversely, an excessively large τ causes the vectors to be overly separated, weakening their correlation and leading to the loss of dynamical information. Similarly, for the embedding dimension m , a value that is too small fails to fully unfold the phase space, causing attractor overlap. On the other hand, an unnecessarily large m introduces computational redundancy, prolonging runtime and consuming excessive memory resources. Traditional methods for estimating time delay include the autocorrelation function method, the average displacement method, and the mutual information method [40–42]. Key techniques for determining the embedding dimension encompass the geometric invariant method, the false nearest neighbors method, and the improved false nearest neighbors method (Cao's method) [43–45].

3.2 Embedding theorem

Theorem 1.1 (*Takens' Embedding Theorem* [46]): if M is a d -dimensional compact manifold, $\varphi : M \rightarrow M$, φ is a smooth differential homomorphism, $y : M \rightarrow R$, and y has second-order successive derivatives, $\varphi(\varphi, y) : M \rightarrow R^{2d+1}$, where:

$$\varphi(\varphi, y) = (y(x), y(\varphi(x)), y(\varphi^2(x)), \dots, y(\varphi^{2d}(x))), x \in M \quad (15)$$

Then $\phi(\varphi, y)$ is an embedding of M into R^{2d+1} .

Takens' embedding theorem establishes that a suitable time delay (τ) and embedding dimension (m) yield a reconstructed phase space trajectory topologically equivalent to the original dynamics. This theorem offers a theoretical basis for time series prediction *via* reconstructed phase space methodologies. The predictor $f_T : R(t + T) = f_T(R(t))$ can be approximated by the global or local neighbor information of past data in the reconstructed phase space [39, 47].

Theorem 1.2 (*Generalized Embedding Theorem* [48]): M is a d -dimensional compact manifold with the set of observation functions $\langle y_k \rangle, k = 1, 2, \dots, 2d + 1$. Here, $y_k : M \rightarrow R$ and y_k has at least C^2 smoothness, then $\Phi_{\langle y_k \rangle} : M \rightarrow R^{2d+1}$,

$$\Phi_{\langle y_k \rangle}(x) = (y_1(x), y_2(x), y_3(x), \dots, y_{2d+1}(x)), x \in M \quad (16)$$

is an embedding of M into R^{2d+1} .

The generalized embedding theorem guarantees topological equivalence between the dynamics of the original system and its reconstructed phase space $\Phi_{\langle y_k \rangle}$, defined by distinct observation functions y_k (Eq. (16)). Notice that the component y_k could be either an unlagged observation variable or a lagged observation variable.

Based on the Takens' embedding theorem and generalized embedding theorem, many excellent prediction algorithms, including multi-view embedding, randomly distributed embedding, delay parameterized method (DPM), and auto-reservoir neural network, are proposed to forecast the low-dimensional chaotic or high-dimensional short-range time series [21–24]. In the following subsections, we will introduce those advanced frontier prediction algorithms in detail.

3.3 Multi-view embedding method

The prediction of complex dynamic systems has long faced two major challenges: the curse of dimensionality and short-lived, noisy data. The curse of dimensionality arises from the exponential growth of model parameters due to high-dimensional variable interactions, while real-world observations are often limited to brief time spans and contaminated by significant noise. These issues make it difficult to balance accuracy and feasibility in traditional approaches, such as linear models (e.g., multivariate auto-regression) or equation-based frameworks. To address these limitations, multi-view embedding (MVE) has emerged. Rooted in Takens' theorem, MVE offers an innovative non-parametric solution by converting system complexity into an informational advantage through attractor reconstruction of multivariate time series.

For the high-dimensional observation variables $\{x_1(t), x_2(t), \dots, x_n(t)\}$, the coordinates in the reconstructed phase space can be,

$$\begin{aligned} &\{x_1(t), x_1(t - \tau), x_1(t - 2\tau), \dots, x_1(t - (l - 1)\tau)\}, \\ &\{x_2(t), x_2(t - \tau), x_2(t - 2\tau), \dots, x_2(t - (l - 1)\tau)\}, \\ &\vdots \\ &\{x_n(t), x_n(t - \tau), x_n(t - 2\tau), \dots, x_n(t - (l - 1)\tau)\}. \end{aligned} \quad (17)$$

Here, l is the possible delays for each variable. The E -dimensional view constructed in the MVE method is,

$$V = \left(x_k(t), x_{s_k^1}(t - \delta_k^1 \tau), x_{s_k^2}(t - \delta_k^2 \tau), \dots, x_{s_k^{E-1}}(t - \delta_k^{E-1} \tau) \right), \quad (18)$$

where $s_k^1, s_k^2, \dots, s_k^{E-1}$ are selected from Eq. (17) except $x_k(t)$ without replacement and $\delta_k^1, \delta_k^2, \dots, \delta_k^{E-1} \in [0, l-1]$, $1 \leq k \leq n$. In fact, each reconstruction created by a specific combination of variables can be regarded as a subspace of a system containing unique information. It should be noted that there is at least one coordinate that had no delay in the reconstruction of V . Thus, the total number of views equals to,

$$m := \binom{nl}{E} - \binom{n(l-1)}{E}. \quad (19)$$

Here, $\binom{nl}{E}$ quantifies attractors formed by selecting E coordinates from nl possibilities, while $\binom{n(l-1)}{E}$ counts invalid attractors where all E coordinates originate exclusively from the $n(l-1)$ coordinates exhibiting positive (non-zero) delay.

The predictions are produced from each univariate view, and then the top $K = \sqrt{m}$ optimal views are filtered by in-sample prediction accuracy. Subsequently, the MVE forecast (e.g., for variable y) is then defined as a simple average,

$$\hat{y}_{t+1} = \frac{1}{k} \sum_{i=1}^k y_{nn^i(t)+1}, \quad (20)$$

where $nn^i(t)$ is the time index of the nearest neighbor in the i th attractor. The sketch map of the MVE method is shown in **Figure 4**.

In conclusion, the MVE method is particularly effective in overcoming the limitations of short time series and noisy time series. However, if the dimension is relatively high, the computational load of this method will be relatively large. The randomly distributed embedding framework proposed by Ma *et al* partially addressed the computational complexity challenge.

3.4 Randomly distributed embedding method

In the era of big data, it is challenging to accurately predict the temporal sample data of a few high-dimensional variables. Ma *et al.* proposed a frameless model, namely randomly distributed embedding (RDE), which demonstrated excellent predictive capabilities for short-term high-dimensional time series [22]. Due to the existence of a large number of interacting variables in high-dimensional non-linear systems, multiple low-dimensional non-delayed embeddings containing historical dynamic information can be generated from high-dimensional observational data according to the delayed embedding theorem, and each attractor is predicted in one step through the corresponding predictor. The distribution of low-dimensional embeddings actually patches all pieces of association information into the whole dynamics of the system. Ultimately, the results of all individual predictors are aggregated to form a probability distribution of the predicted values. The final predicted value is obtained through weighted averaging based on the selected randomly embedded predicted values.

For a short-term high-dimensional time series $\{x_1(t), x_2(t), \dots, x_n(t)\}$, the l th low-dimensional non-delay embedding can be,

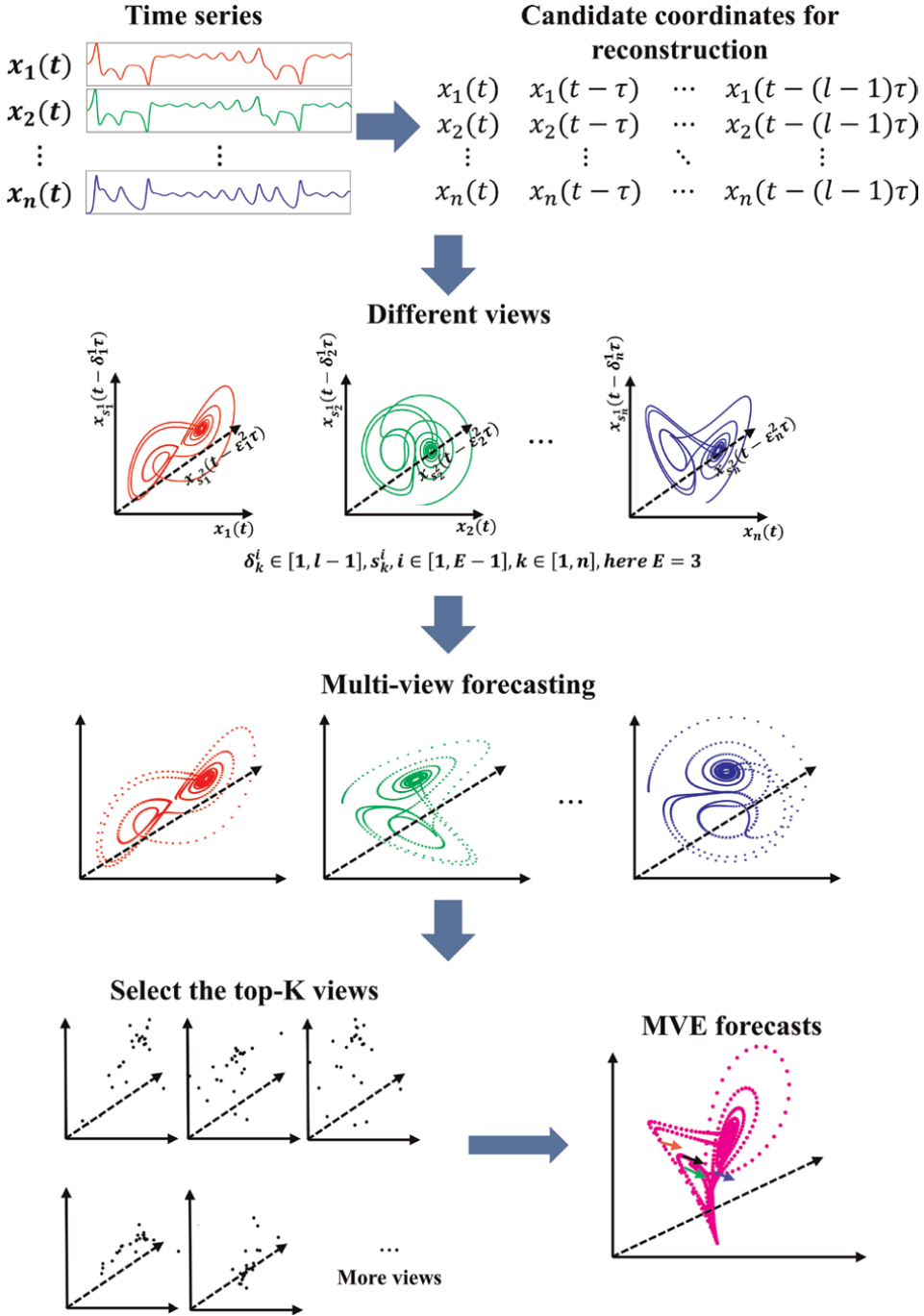


Figure 4.
The sketch map of the MVE method.

$$S^l = \begin{pmatrix} x_{s_1^l}^l(1) & x_{s_1^l}^l(2), & \cdots & x_{s_1^l}^l(T) \\ x_{s_2^l}^l(1) & x_{s_2^l}^l(2), & \cdots & x_{s_2^l}^l(T) \\ \vdots & \vdots & \ddots & \vdots \\ x_{s_E^l}^l(1) & x_{s_E^l}^l(2) & \cdots & x_{s_E^l}^l(T) \end{pmatrix}, \quad (21)$$

where $s_1^l, s_2^l, \dots, s_E^l \in [1, n]$ and E is the embedding dimension. The delay embedding is the reconstruction phase space R_k related to the target variable x_k .

$$R_k = \begin{pmatrix} x_k(1) & x_k(2), & \cdots & x_k(T - (E - 1)\tau) \\ x_k(1 + \tau) & x_k(2 + \tau), & \cdots & x_k(T - (E - 2)\tau) \\ \vdots & \vdots & \ddots & \vdots \\ x_k(1 + (E - 1)\tau) & x_k(2 + (E - 1)\tau) & \cdots & x_k(T) \end{pmatrix}. \quad (22)$$

Embedding theory indicates that all the aforementioned reconstructed attractors are topologically equivalent to the original attractor, due to a diffeomorphic mapping. Thus, for each index tuple, $l = (s_1^l, s_2^l, \dots, s_E^l)$ a component of such a mapping, denoted by ψ^l , can be obtained as a predictor for the target variable $x_k(t)$ in the form of,

$$x_k(t + \tau) = \psi^l(x_{s_1^l}^l(t), x_{s_2^l}^l(t), \dots, x_{s_E^l}^l(t)). \quad (23)$$

Thus, the one-step prediction $\hat{x}_k^l(t^* + \tau)$ can be obtained with different index tuples l . The multiple prediction values $\hat{x}_k^l(t^* + \tau)$ actually form a probability distribution. So the better prediction can be estimated by,

$$\hat{x}_k(t^* + \tau) = \varepsilon[\hat{x}_k^l(t^* + \tau)], \quad (24)$$

where ε represents an estimation based on the available probability information of the random variable $\hat{x}_k^l$. The sketch map of the RDE method is shown in **Figure 5**. In the RDE framework, the application of probability distribution information enhances the noise resistance of the prediction algorithm. Besides, the random selection of low-dimensional subspaces for attractor reconstruction reduces the computational complexity of the MVE method.

From a theoretical perspective, RDE transforms high-dimensional characteristics into information advantages through the exchange of spatial information (high-dimensional variable associations) and temporal distribution (predicted value probabilities), providing a universal framework for the prediction of short-term high-dimensional data.

3.5 Delay parameterized method

As described in the above MVE and RDE methods, the high-dimensional short-term time series can be predicted by using the dynamical evolution information of the

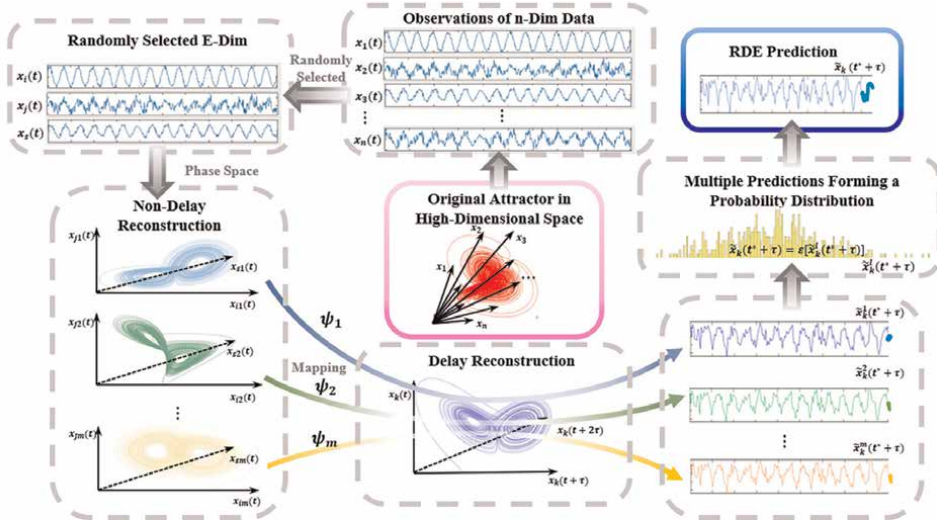


Figure 5.
The sketch map of the RDE method.

reconstructed phase space. However, it is a more complex problem to predict the low-dimensional mid-term chaotic time series with insufficient information. To address this issue, Guo *et al.* designed a delay parameterized method (DPM) for the prediction of low-dimensional medium-term time series [23]. The time delay is defined as an unknown parameter to be estimated, and the parameter-dependent reconstruction can incorporate richer dynamical information into the prediction algorithm.

Figure 6 briefly presents the main idea of DPM. The low-dimensional mid-term observations, denoted as $\{x_1(t), x_2(t), \dots, x_m(t)\}$, are one subspace of the original system. Then, the corresponding phase space of the same dimension is reconstructed from the observations. Between the initial phase space and the reconstructed phase space, a correlation function is proposed to achieve the prediction of multi-dimensional coupled chaotic time series. When reconstructing the phase space, the embedding dimension is fixed equal to the dimension of the state space. The time delay is defined as an unknown parameter, and the appropriate time delay is selected by using optimization algorithms such as the traversal algorithm, the genetic algorithm, and the particle swarm optimization algorithm.

The non-delay subspace in matrix form is,

$$\mathcal{X} = \begin{pmatrix} x_1(1) & x_2(1) & \cdots & x_m(1) \\ x_1(2) & x_2(2) & \cdots & x_m(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(N) & x_2(N) & \cdots & x_m(N) \end{pmatrix}, \quad (25)$$

and the delay reconstructed phase space is,

$$X_k = \begin{pmatrix} x_k(1) & x_k(1+\tau) & \cdots & x_k(1+(m-1)\tau) \\ x_k(2) & x_k(2+\tau) & \cdots & x_k(2+(m-1)\tau) \\ \vdots & \vdots & \ddots & \vdots \\ x_k(N) & x_k(N+\tau) & \cdots & x_k(N+(m-1)\tau) \end{pmatrix}. \quad (26)$$

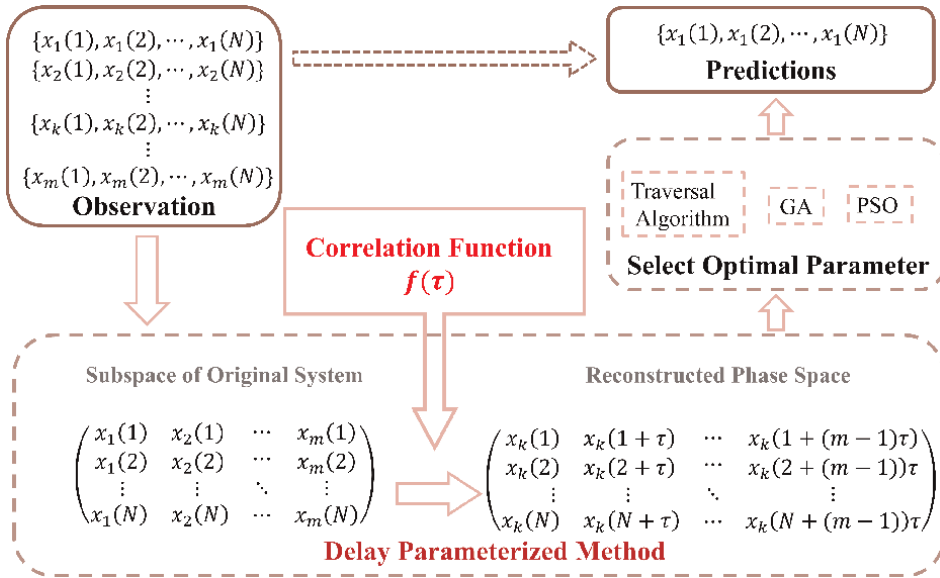


Figure 6. The main idea of DPM. The correlation function f_τ bridges the relation between the original space and the reconstructed space, making the low-dimensional mid-term observations predictable.

Note that the values $\{x_k(N+\tau), x_k(N+2\tau), \dots, x_k(N+(m-1)\tau)\}$ in Eq. (26) are unknown data that need to be predicted. The predictor f_τ satisfies,

$$X_k = f_\tau(\mathcal{X}) \quad (27)$$

A linear approximation of f_τ can be obtained by the first $N - (m-1)\tau$ lines of well-known data in X_k and $\mathcal{X}$.

If the linear correlation function supposes $\bar{X}_k = \bar{\mathcal{X}}P$, where $\bar{X}_k$ and $\bar{\mathcal{X}}$ are the first $N - (m-1)\tau$ lines of X_k and $\mathcal{X}$. Then, the parameter matrix P can be solved by the least squares method. That is,

$$P = (\bar{\mathcal{X}}^T \bar{\mathcal{X}})^{-1} \bar{\mathcal{X}}^T \bar{X}_k. \quad (28)$$

Once the parameter matrix P is obtained, the prediction of $\{x_k(N+k), k = 1, 2, \dots, (m-1)\tau\}$ can be achieved by Eq. (27). The fitness function F is defined as,

$$F = \frac{1}{d} \sum_{j=1}^d \frac{\text{Corr}(x_j, \hat{x}_j)}{\text{Error}(x_j)} \quad (29)$$

where x_j is the real value of j^{th} observed signals, $\hat{x}_j$ is the predicted value, and d is the dimension of observations. Therefore, the algorithm, including the traversal algorithm, GA, and PSO algorithm, aims to find the parameters $\tau = (\tau_1, \tau_2, \dots, \tau_d)$ to maximize the function F .

Similarly, the non-linear correlation function between X_k and $\mathcal{X}$ can be approximately expressed by a non-linear polynomial function. The state space is refined $\mathcal{X} =$

$\{x_i, x_i x_j, x_i x_j x_k, \dots\}$ to include the high-order non-linear relationships. Here, the Hadamard product $x_i x_j$ represents components x_i coupling a non-linear relationship with x_j . The DPM method, based on the embedding theorem and topological theory, can effectively predict chaotic time series, including low-dimensional intermediate time series. The feasibility and validity of the method are verified by predicting the Lorentz chaotic system, material plastic deformation, and stock price fluctuation. Due to a severe lack of information, it is difficult to predict short-term time series, which is an unresolved issue that requires further investigation.

4. Physics-informed neural network approaches

To address the limitations of the “black-box” nature of neural networks, physics-informed neural networks have significantly enhanced model interpretability by integrating physical prior knowledge with neural networks, providing novel research perspectives for the modeling and prediction of complex systems. In this section, we introduce the application of PINNs and TDINNs in scientific computing and the prediction of complex dynamic systems. We first elaborate on the advantages of PINNs in solving both forward and inverse problems of partial differential equations, then explore the TDINN algorithm, which embeds transmission dynamics into neural networks based on the PINNs framework, providing an effective tool for the predictions of time series in complex dynamical systems.

4.1 Physics-informed neural networks

Physics-informed neural networks address the interpretability challenge through innovative integration of differential equation constraints with deep learning architectures, establishing a new physics-guided modeling paradigm. PINNs extend conventional neural network frameworks by integrating governing physical equations into the learning process. This hybrid approach enables simultaneous optimization by integrating empirical data with fundamental physical laws, thereby improving model accuracy while maintaining interpretability.

A brief outline of the PINNs framework is shown in **Figure 7**. Suppose the constrained equation satisfies an ODE or PDE model $\frac{du}{dt} = F\left(u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right)$. The input signals of PINNs are x and t , and the output signal is u . Besides, the loss function is consistent of data loss MSE_u and model loss MSE_f , i.e.,

$$LOSS = MSE_u + \lambda MSE_f, \quad (30)$$

where λ is a weight parameter, and

$$MSE_u = \frac{1}{n} \sum_{i=1}^n (u_i - \hat{u}_i)^2, MSE_f = \frac{1}{n} \sum_{i=1}^n \left(\frac{\partial u_i}{\partial t} - F\left(u_i, \frac{\partial u_i}{\partial x}, \frac{\partial^2 u_i}{\partial x^2}\right) \right)^2, \quad (31)$$

$\hat{u}_i$ represents the real value. Then, the model parameters are solved through the back-propagation method.

Generally, the application of the PINNs framework can be divided into two categories: forward problems for solution prediction and inverse problems for parameter or model identification. A forward problem is to solve the solution of a physical model

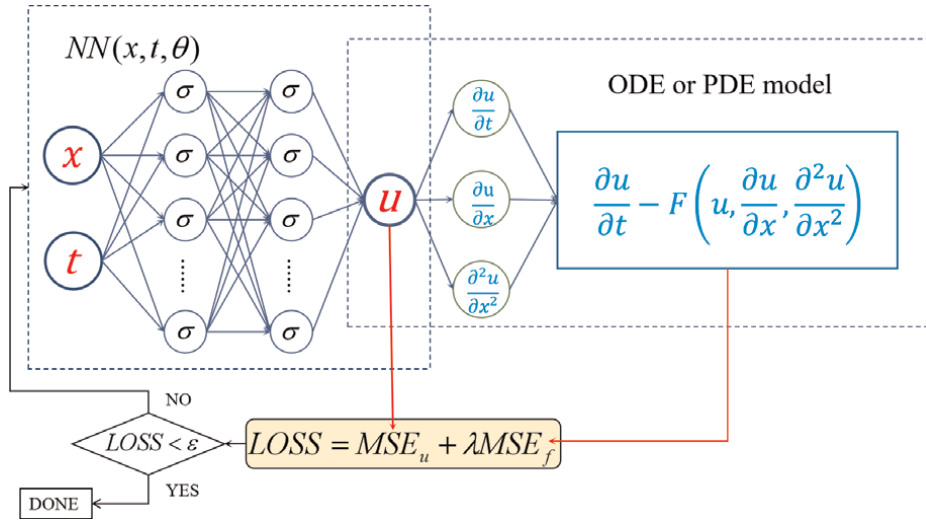


Figure 7.
 A brief outline of the PINNs framework. The input signals are x and t , and the output signal is u . The loss function consists of data loss MSE_u and model loss MSE_f .

F with known initial/boundary conditions and explicit governing equations, while an inverse problem aims to recover unknown model parameters or governing equations from partial observational data of the physical system. Next, we take a damped oscillator model as a case study to elaborate on the application of the PINNs framework for both forward and inverse problems. The damped oscillator model follows,

$$\begin{cases} m \frac{d^2 u}{dt^2} + \mu \frac{du}{dt} + ku = 0, \\ u(t = 0) = 1, \frac{du}{dt}(t = 0) = 0. \end{cases} \quad (32)$$

Here, $u(t)$ is the displacement, m is the oscillator mass, μ is the damping coefficient, and k is the spring constant. Let $\delta = \frac{\mu}{2m}$, $\omega_0 = \sqrt{\frac{k}{m}}$, when $\delta < \omega_0$, the Eq. (32) has an exact solution,

$$u(t) = e^{-\delta t} (2A \cos(\phi + \omega t)), \quad \omega = \sqrt{\omega_0^2 - \delta^2}. \quad (33)$$

Forward problem: We have the initial condition $u(t = 0) = 1$, boundary condition $\frac{du}{dt}(t = 0) = 0$, and the parameters $m = 1$, $\delta = 2$, $\omega_0 = 20$ for $t \in [0, 1]$. The PINNs are used to train the neural network $NN(t; \theta)$ to approximate the solution of Eq. (32). The corresponding loss function is defined as,

$$\mathcal{L}(\theta) = \underbrace{(NN(0; \theta) - 1)^2}_{\text{Initial condition}} + \underbrace{\lambda_1 \left(\frac{dNN}{dt}(0; \theta) - 0 \right)^2}_{\text{Boundary condition}} + \underbrace{\lambda_2 \sum_{i=1}^N \left(\left[m \frac{d^2}{dt^2} + \mu \frac{d}{dt} + k \right] NN(t_i; \theta) \right)^2}_{\text{Model residual}}, \quad (34)$$

where λ_1 and λ_2 are weighting parameters that balance the contributions of different constraint terms in the loss function.

During the training process, the neural network's parameters are iteratively optimized to minimize the loss function. **Figure 8** illustrates the predicted solutions of the PINNs model, compared with the exact solution, at the training steps 0,5000,10000, and 15000. It reveals that the PINNs can efficiently approximate the solution of PDE models.

Inverse problem: We have the noisy observational data, u_{obs} , of oscillator displacement, and the form of a constrained equation. The objective is to estimate the damping coefficient μ . Thus, the neural network $NN(t; \theta)$ is designed to approximate the solution $u(t)$ while simultaneously learning the unknown parameter μ . The corresponding loss function is,

$$\mathcal{L}(\theta, \mu) = \underbrace{\frac{1}{N} \sum_{i=1}^N \left(\left[m \frac{d^2}{dt^2} + \mu \frac{d}{dt} + k \right] NN(t_i; \theta) \right)^2}_{\text{Model residual}} + \underbrace{\frac{\lambda}{M} \sum_{j=1}^M (NN(t_j; \theta) - u_{obs}(t_j))^2}_{\text{Data loss}} \quad (35)$$

Figure 9 shows the estimation performance of the exact solution and the real damping coefficient at the training steps 5000 and 15000. This demonstrates that the PINNs framework can effectively identify unknown model parameters and/or governing equations from partial observational data.

4.2 Transmission dynamics-informed neural networks

Based on the PINNs framework, He *et al* proposed a transmission dynamics-informed neural network (TDINN) by encoding the Susceptible-Exposed-Infected-Recovered-Dead (SEIRD) compartment model into deep neural networks to fit multi-wave COVID-19 epidemic data. [26]. The SEIRD model equations are given as follows:

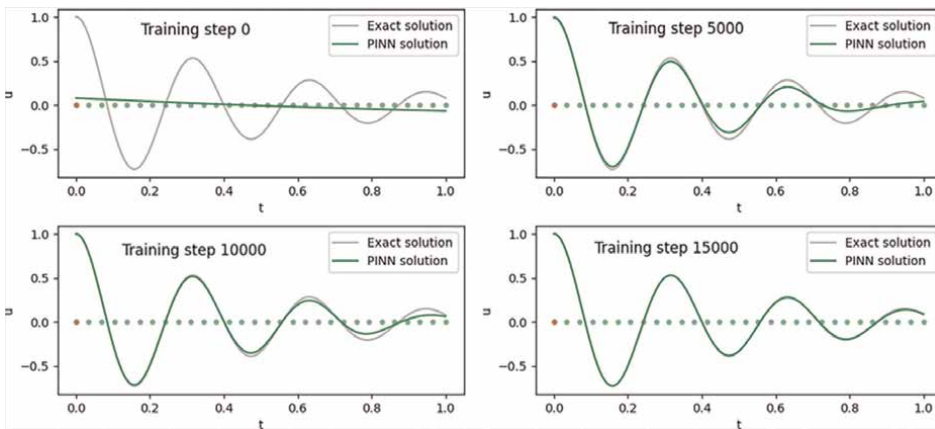


Figure 8.
The predicted solutions from the PINNs model, compared with the exact solution, at training steps 0,5000,10000, and 15000.

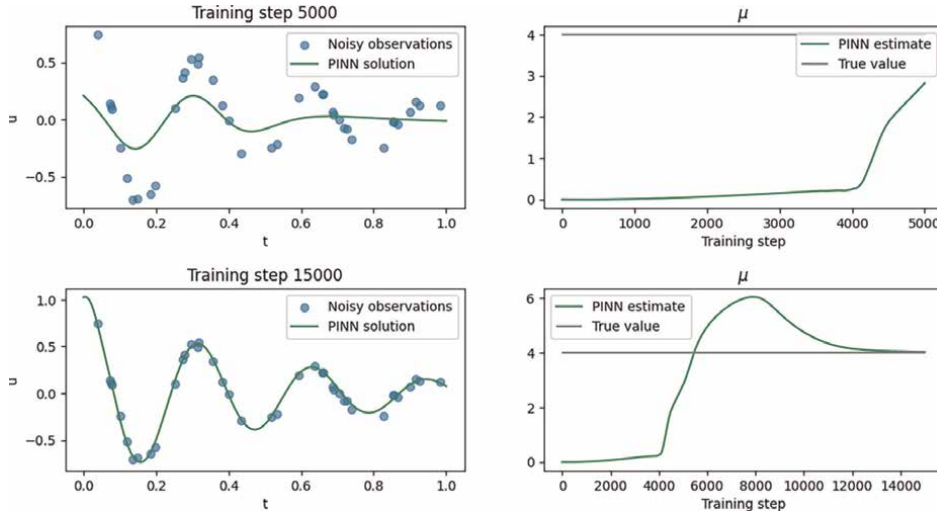


Figure 9.
 The estimation performance of the exact solution $u(t)$ and the real damping coefficient μ at the training steps 5000 and 15,000.

$$\begin{cases} \frac{dS(t)}{dt} = -\beta(t) \frac{S(t)I(t)}{N}, \\ \frac{dE(t)}{dt} = \beta(t) \frac{S(t)I(t)}{N} - \alpha E(t), \\ \frac{dI(t)}{dt} = \alpha E(t) - \gamma I(t) - \delta I(t), \\ \frac{dR(t)}{dt} = \gamma I(t), \\ \frac{dD(t)}{dt} = \delta I(t). \end{cases} \quad (36)$$

The total population N is partitioned into compartments such that $N(t) = S(t) + E(t) + I(t) + R(t) + D(t)$. The transmission rate $\beta(t)$ is modelled as a time-dependent function to account for dynamic factors such as behavioural adaptations, vaccination campaigns, and the emergence of new viral variants. In contrast, the transition rate α (from exposed to infectious states), along with the recovery rate γ and mortality rate δ , are treated as constants.

TDINN incorporates two independent neural networks DNN_β and DNN_M , as shown in **Figure 10**. The DNN_β is used to learn the transmission rate function $\beta(t)$, while the DNN_M is applied to learn the evolution of the SEIRD model. The optimization objective is characterized by the following loss function: $\mathcal{L} = Loss_{data} + Loss_{ode}$.

It should be noted that, in compartment models, not all epidemic variables are observable. We only have daily reported cases I_{new} , cumulative reported cases I_{cum} , daily death cases D_{new} , and cumulative death cases D_{cum} . Therefore, the data loss $Loss_{data}$ is generated by the only available data information. Since the $D(t)$ compartment has no outflow terms, its output directly represents cumulative deaths (D_{cum}). The cumulative reported cases can be characterized by I_{cum} , and

$$\frac{dI_{cum}(t)}{dt} = \alpha E(t). \quad (37)$$

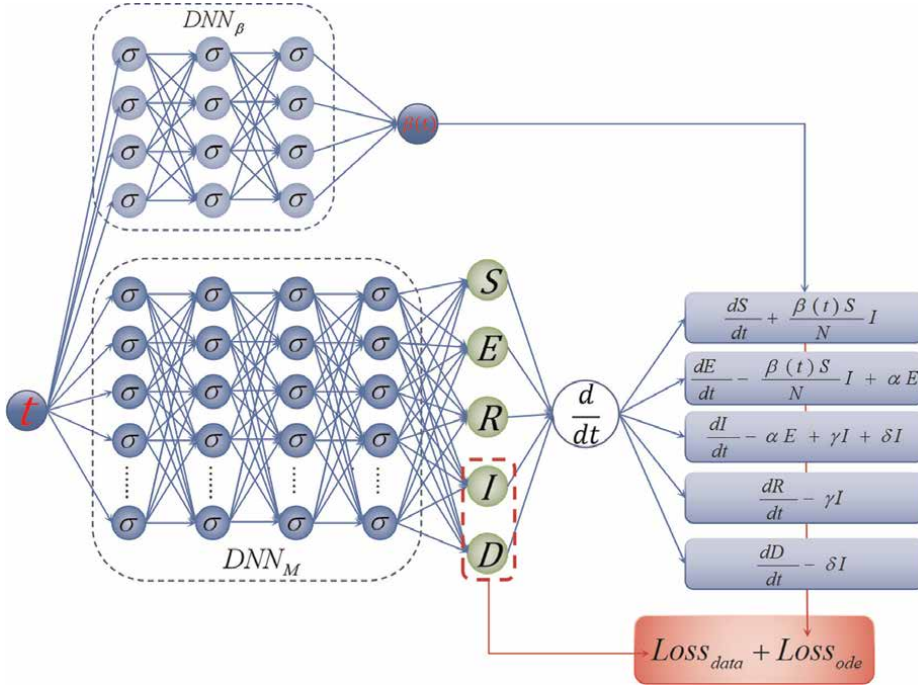


Figure 10.
The outline of the TDINN method.

Thus, the data loss $Loss_{data}$ is defined as,

$$\begin{aligned}
 Loss_{data} = & \frac{1}{N_d} \sum_{i=1}^{N_d} |I_{new}^{NN}(t_i) - I_{new}(t_i)|^2 + \frac{1}{N_d} \sum_{i=1}^{N_d} |D_{new}^{NN}(t_i) - D_{new}(t_i)|^2 \\
 & + \frac{1}{N_d} \sum_{i=1}^{N_d} |I_{cum}^{NN}(t_i) - I_{cum}(t_i)|^2 + \frac{1}{N_d} \sum_{i=1}^{N_d} |D_{cum}^{NN}(t_i) - D_{cum}(t_i)|^2
 \end{aligned} \quad (38)$$

where $I_{new}^{NN}(t) = I_{cum}^{NN}(t) - I_{cum}^{NN}(t-1)$, $D_{new}^{NN}(t) = D_{cum}^{NN}(t) - D_{cum}^{NN}(t-1)$, and N_d represent the size of the training set. Based on the model equations, the residuals of each component are,

$$\begin{aligned}
 L_1(t_i) &= \frac{d}{dt} S^{NN}(t_i) + \frac{1}{N} \beta^{NN}(t_i) S^{NN}(t_i) I^{NN}(t_i) \\
 L_2(t_i) &= \frac{d}{dt} E^{NN}(t_i) - \frac{1}{N} \beta^{NN}(t_i) S^{NN}(t_i) I^{NN}(t_i) + \alpha E^{NN}(t_i) \\
 L_3(t_i) &= \frac{d}{dt} I^{NN}(t_i) - \alpha E^{NN}(t_i) + \gamma I^{NN}(t_i) + \delta I^{NN}(t_i) \\
 L_4(t_i) &= \frac{d}{dt} R^{NN}(t_i) - \gamma I^{NN}(t_i) \\
 L_5(t_i) &= \frac{d}{dt} D^{NN}(t_i) - \delta I^{NN}(t_i) \\
 L_6(t_i) &= \frac{d}{dt} I_{cum}^{NN}(t_i) - \alpha E^{NN}(t_i)
 \end{aligned} \quad (39)$$

Therefore,

$$Loss_{ode} = \frac{1}{N_e} \sum_{j=1}^6 \sum_{i=1}^{N_e} |L_j(t_i)|^2 \quad (40)$$

where N_e represents the number of residual points. The weights and biases of DNN_M and DNN_β are randomly initialized at first, then an Adam optimizer is employed with a fixed learning rate of 0.001. During back-propagation, gradients of residuals are computed *via* automatic differentiation to update network parameters and minimize the total loss.

The proposed TDINN algorithm based on physics-informed neural networks demonstrates unique advantages in time series forecasting by effectively capturing non-linear relationships and dynamic variations in complex systems. This method not only achieves high accuracy in predicting epidemic trends and peak timing of infectious diseases but also robustly handles time series data with noise and non-linear characteristics. By integrating observational data with embedded epidemiological dynamics equations, TDINN optimizes parameter estimation while strictly adhering to bio-physical constraints, thereby significantly enhancing the prediction model's accuracy and reliability. The data-model coupling framework of TDINN not only improves predictive capability but also ensures consistency with real-world system dynamics, establishing itself as a powerful tool for time series forecasting applications.

5. Conclusions

The chapter systematically explores the advancements in time series prediction, focusing on the integration of data-driven and model-driven approaches to address the limitations of traditional methods. Data-driven techniques, such as BP neural networks, LSTM, and reservoir computing, excel in capturing non-linear relationships but often lack interpretability. Conversely, model-driven methods provide theoretical insights but may underperform in complex, real-world scenarios.

To bridge this gap, hybrid approaches like physics-informed neural networks (PINNs) and transmission dynamics-informed neural networks (TDINNs) have emerged. These methods combine the predictive power of neural networks with domain-specific physical laws, enhancing accuracy, interpretability, and generalizability. Additionally, dynamics-inspired frameworks, such as MVE and RDE, leverage phase space reconstruction and embedding theorems to improve predictions for chaotic and high-dimensional systems.

Despite these advancements, challenges remain, particularly in predicting low-dimensional short-term time series and optimizing computational efficiency. Future research should focus on refining hybrid models, improving noise resistance, and developing scalable algorithms for broader applications. By continuing to integrate data-driven and model-driven paradigms, the field can achieve more robust and interpretable solutions for complex time series forecasting across diverse domains.

Acknowledgements

This work was supported by the China Postdoctoral Science Foundation Funded Project (Grant No. 2022 M712902), the Natural Science Foundation of Henan (Grant

No. 232300421345), and the Zhongyuan Youth Top Talent Plan (Zhongyuan Youth Postdoctoral Innovative Talents).

Conflict of interest

The authors declare no conflict of interest.

Author details

Xiaoxiang Guo*, Zhen Li, Jingrun Zhang and Chongxiao Zheng
School of Mathematics and Statistics, Henan Academy of Big Data, Zhengzhou University, Zhengzhou, China

*Address all correspondence to: guoxiaoxiang0379@163.com

IntechOpen

© 2025 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Clauset A, Larremore DB, Sinatra R. Data-driven predictions in the science of science. *Science*. 2017;**355**:477-480
- [2] Subrahmanian VS, Kumar S. Predicting human behavior: The next frontiers. *Science*. 2017;**355**:489-489
- [3] Ginsberg J, Mohebbi MH, Patel RS, Brammer L, Smolinski MS, Brilliant L. Detecting influenza epidemics using search engine query data. *Nature*. 2009;**457**:1012-1014
- [4] Liu Y, Liu Q, Wang W, Zhao J, Leung H. Data-driven based model for flow prediction of steam system in steel industry. *Information Sciences*. 2012;**193**:104-114
- [5] Acharya UR, Sree SV, Swapna G, Martis RJ, Suri JS. Automated EEG analysis of epilepsy: A review. *Knowledge-Based Systems*. 2013;**45**: 147-165
- [6] Lee RST. Chaotic type-2 transient-fuzzy deep neuro-oscillatory network (CT2TFDNN) for worldwide financial prediction. *IEEE Transactions on Fuzzy Systems*. 2020;**28**(4):731-745
- [7] Li K, Xie X, Xue W, Chen X. Hybrid teaching—Learning artificial neural network for city-level electrical load prediction. *Science China Information Sciences*. 2020;**63**:159204
- [8] Li J, Shi Y, Li H, Yang B. TC-GATN: Temporal causal graph attention networks with nonlinear paradigm for multivariate time-series forecasting in industrial processes. *IEEE Transactions on Industrial Informatics*. 2023;**19**(6): 7592-7601
- [9] Cattaneo MD, Fan YY, Li RZ, Song R. Data science in economics and finance: Introduction. *Journal of Econometrics*. 2024;**239**(2):105627
- [10] Jaeger H, Haas H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*. 2004;**304**: 78-80
- [11] Box GEP, Jenkins GM. *Time Series Analysis, Forecasting and Control*. San Francisco, CA: Holden-Day; 1970
- [12] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*. 1997;**9**(8):1735-1780
- [13] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;**521**:436-444
- [14] Zhu XL, Fu BF, Yang YD, Ma Y, Hao JY, Chen SQ, et al. Attention-based recurrent neural network for influenza epidemic prediction. *BMC Bioinformatics*. 2019;**20**:575
- [15] Zheng HF, Lin F, Feng XX, Chen YJ. A hybrid deep learning model with attention-based Conv-LSTM networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*. 2021;**22**(11): 6910-6920
- [16] Yan H, Tian C. A novel two-stage forecasting model based on error factor and ensemble method for multi-step wind power forecasting. *Applied Energy*. 2019;**238**:368-383
- [17] Wen T, Chen H, Cheong KH. Visibility graph for time series prediction and image classification: A review. *Nonlinear Dynamics*. 2022;**110**: 2979-2999
- [18] Griffith A, Pomerance A, Gauthier DJ. Forecasting chaotic systems

with very low connectivity reservoir computers. *Chaos*. 2019;**29**(12): 123108

[19] Barbosa WAS, Gauthier DJ. Learning spatiotemporal chaos using next-generation reservoir computing. *Chaos*. 2022;**32**(9):093137

[20] Liu JM, Xu X, Guo WJ, Li E. Learning hidden dynamics using echo state network. *Nonlinear Dynamics*. 2025;**113**:14181-14199

[21] Ye H, Sugihara G. Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality. *Science*. 2016;**353**:922-925

[22] Ma H, Leng S, Aihara K, Lin W, Chen L. Randomly distributed embedding making short-term high-dimensional data predictable. *Proceedings of the National Academy of Sciences of the United States of America*. 2018;**115**(43):E9994-E10002

[23] Guo XX, Sun YT, Ren JL. Low dimensional mid-term chaotic time series prediction by delay parameterized method. *Information Sciences*. 2020;**516**: 1-19

[24] Liu R, Aihara K, Chen LN. Autoreervoir computing for multistep ahead prediction based on the spatiotemporal information transformation. *Nature Communications*. 2020;**11**:4568

[25] Chen Z, Liu Y, Sun H. Physics-informed learning of governing equations from scarce data. *Nature Communications*. 2021;**12**:6136

[26] He MQ, Tang B, Xiao YN, Tang SY. Transmission dynamics informed neural network with application to COVID-19 infections. *Computers in Biology and Medicine*. 2023;**165**:107431

[27] Li H, Li J, Wu Z, Wang M, Liu G, Sun R, et al. Dynamics-based predictions of infinite-dimensional complex systems using dynamical system deep learning method. *Machine Learning: Science and Technology*. 2025; **6**(2):025008

[28] Rumelhart D, Hinton G, Williams R. Learning representations by back-propagating errors. *Nature*. 1986;**323**: 533-536

[29] Lu X, Poon J, Khushi M. Bridging the gap between machine and human in stock prediction: Addressing heterogeneity in stock market. *IEEE Access*. 2024;**12**:186171-186185

[30] Maass W, Markram H. On the computational power of circuits of spiking neurons. *Journal of Computer and System Sciences*. 2004;**69**(4): 593-616

[31] Na X, Ren W, Liu M, Han M. Hierarchical echo state network with sparse learning: A method for multidimensional chaotic time series prediction. *IEEE Transactions on Neural Networks and Learning Systems*. 2023; **34**(11):9302-9313

[32] Ye ZW, Wei DQ. Predicting chaos in permanent magnet synchronous motor using dual reservoir interaction with high-order state reservoir computing. *Physica Scripta*. 2025;**100**(3):036010

[33] Bollt E. On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD featured. *Chaos*. 2021;**31**:013108

[34] Gauthier DJ, Bollt E, Griffith A, Barbosa WAS. Next generation reservoir computing. *Nature Communications*. 2021;**12**:5564

- [35] Yan M, Huang C, Bienstman P, Tino P, Lin W, Sun J. Emerging opportunities and challenges for the future of reservoir computing. *Nature Communications*. 2024;**15**:2056
- [36] Liang XP, Tang JS, Zhong YN, Gao B, Qian H, Wu HQ. Physical reservoir computing with emerging electronics. *Nature Electronics*. 2024;**7**: 193-206
- [37] Ehlers PJ, Nurdin HI, Soh D. Stochastic reservoir computers. *Nature Communications*. 2025;**16**:3070
- [38] Packard NH, Crutchfield JP, Farmer JD, Shaw RS. Geometry from a time series. *Physical Review Letters*. 1980;**45**(9):712-716
- [39] Farmer JD, Sidorowich JJ. Predicting chaotic time series. *Physical Review Letters*. 1987;**59**:845-848
- [40] Rosenstein MT, Collins JJ, De LC. Reconstruction expansion as a geometry-based framework for choosing proper delay times. *Physica D: Nonlinear Phenomena*. 1994;**73**(1-2):82-98
- [41] Rosenstein MT, Collins JJ, De LC. A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*. 1993;**65**(1-2):117-134
- [42] Fraser AM, Swinney HL. Independent coordinates for strange attractors from mutual information. *Physical Review A*. 1986;**33**(2): 1134-1140
- [43] Grassberger P, Procaccia I. Characterization of strange attractors. *Physical Review Letters*. 1983;**50**(5): 346-349
- [44] Kennel MB, Brown R, Abarbanel HD. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*. 1992; **45**(6):3403-3411
- [45] Cao LY. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D: Nonlinear Phenomena*. 1997;**110**(1-2): 43-50
- [46] Takens F. Detecting strange attractors in turbulence. *Lecture Notes in Mathematics*. 1981;**898**(1):366-381
- [47] Sugihara G, May R. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*. 1990;**344**(6268):734-741
- [48] Deyle ER, Sugihara G. Generalized theorems for nonlinear state space reconstruction. *PLoS One*. 2011;**6**(3):1-8

Section 3

Graphs, Dependencies, and Networked Time Series

Integration of Time Series and Multiplex Networks Series in Cybersecurity and Prediction: An Innovative Approach

Sergio Iglesias Pérez and Regino Criado Herrero

Abstract

This review covers a recent and innovative methodological framework that uses novel tools developed within the field of network science to address some of the limitations of traditional time series and network analysis techniques. It summarizes and complements a set of studies that introduce and validate a novel approach based on the transformation of interaction-based time series into multiplex networks. This is achieved by first converting time series into visibility graphs (VGs), extracting their topological characteristics, grouping these characteristics to identify distinct behavior patterns, and finally structuring these patterns as layers within a multiplex network. At the core of this framework is the temporal metagraph, a mathematical structure in which temporal dynamics between entities are intrinsically encoded. The effectiveness of this framework is demonstrated by its application in two disparate and complex domains: cybersecurity, where it improves network intrusion detection systems (NIDSs) by providing more accurate and efficient identification of attackers; and predictive analytics, where it leverages urban mobility data to forecast real estate market trends with significant improvements over state-of-the-art models. The synthesis reveals that this integrated approach offers a powerful, generalizable, and computationally efficient paradigm for extracting useful intelligence from vast streams of temporal interaction data.

Keywords: complex networks, multiplex networks, time series, visibility graph, machine learning, cybersecurity, temporal metagraph, computational efficiency

1. Introduction

The analysis of complex systems, based on the development of network science, has stood out for capturing the static topology of relationships between entities, providing deep insight into the structure and organization of systems. On the other hand, time series analysis has specialized in modeling the evolution of individual variables over time, identifying trends, periodicities, and anomalies. While both fields have been immensely fruitful in their respective domains, this conceptual

separation often fails to capture the essence of a wide range of real-world phenomena in which the nature of interactions is inherently dynamic and co-evolutionary. The topology of a system influences its temporal dynamics, which in turn reshapes the topology. A framework that treats these two dimensions in isolation is fundamentally limited. To bridge this gap, several temporal network models have been developed. The most common approaches, such as snapshot-based or time-segment-based representations, conceptualize a temporal network as a sequence of static graphs, where each graph represents the state of the system at a specific instant or within a discrete time window. While these models offer a sequential view of network evolution, they have significant limitations. They often struggle to capture the continuous nature of interactions, as they treat time as a series of disconnected states rather than as a fluid dimension. This can obscure the emergent properties and holistic dynamics that unfold throughout the entire lifecycle of an interaction. This review details a methodological framework that transcends these limitations by proposing not only a new algorithm, but a fundamentally different philosophy of data representation [1–5]. This philosophy is based on a hierarchical abstraction process that systematically transforms massive, high-speed streams of raw event data into a compact, feature-rich, and highly predictive final representation. This process and its representation are designed to systematically reduce the volume of data while increasing its informational value at each stage of the process. The central thesis of this approach is based on a clear and effective concatenation of stages. The behavior over time of an interaction—for example, the rhythmic pattern of probing and testing by a cyber attacker or the daily routine of an urban traveler—directly causes a specific and unique topological structure when transformed into a visibility graph. In turn, the topology of this graph causes it to be classified into a specific behavior group. These groups form the layers of a final multiplex network that encompasses the entire process, a structure that encapsulates and classifies the types of interactions of the agents and/or actors involved in the system. It is this final, distilled representation that enables accurate, high-fidelity prediction.

This review provides an exhaustive exploration of this innovative framework and methodology. Thus, after this Introduction, Section 2 begins by laying the theoretical groundwork, introducing the concepts of the temporal metagraph and the visibility graph transformation, which are used to encode dynamics (time series) into a static, analyzable structure. Section 3 is devoted then to detail the k-visibility algorithm for quantifying and clustering these structures into behavioral archetypes or patterns. Following this, in Section 4 the construction of the final analytical model, the Temporal Behavior Multiplex Network, is described. Then, we present two extensive application sections, demonstrating the framework's remarkable predictive power and versatility in the disparate domains of socioeconomic forecasting and advanced cybersecurity. Finally, as conclusions, the review synthesizes these findings, positions the framework within the current state of the art of data science, and outlines promising trajectories for future research and development.

2. The temporal metagraph and visibility graph transformation: Encoding dynamics into structure

At the heart of this methodology lie a novel mathematical foundation and a robust transformation process that converts the fluid, one-dimensional nature of time into a static structure that can be analyzed using the powerful tools of graph theory.

This section details the model for constructing the visibility graph (VG) associated with a time series and the latest theoretical concept known as the temporal metagraph. Broadly speaking, the metagraph structure is based on a graph, in which its edges are themselves visibility graphs. As an example, consider the time series corresponding to communications between different Internet Protocol (IP) addresses. Each time series corresponds to the communication between two IPs during a certain period of time, and the metagraph groups and integrates all communications, giving rise to a new graph structure (metagraph) in which the nodes are the IP addresses of the system, and the edges are the visibility graphs associated with the time series corresponding to the communications in that period of time between each two IP addresses. In other words, each edge of the metagraph structure represents the visibility graph corresponding to the time series of communications between the two nodes in question. Thus, the process begins with the aggregation of raw event data. For any pair of interactive nodes (e.g., two IP addresses, two urban areas), a univariate time series is generated that quantifies their interaction (e.g., bytes sent, number of taxi trips) at a specific frequency (e.g., every hour, every day) during a defined time interval. This time series is then transformed into a graph using one of the two visibility graph construction methods.

2.1 Natural visibility graph (NVG)

The natural visibility graph (NVG) algorithm is based on the principle of direct, line-of-sight visibility [6]. Each data point in the time series is treated as a node in the graph. Two nodes, corresponding to data points (t_a, y_a) and (t_b, y_b) in the time series, are connected by an edge if the straight line segment joining them is not obstructed by any intermediate data point (t_c, y_c) [6]. Formally, an edge exists between nodes a and b (with $t_a < t_b$) if for any intermediate time t_c ($t_a < t_c < t_b$) the corresponding value y_c satisfies the condition $y_c < y_b + (y_a - y_b)(t_b - t_a)t_b - t_a t_b - t_c$.

As we can see in **Figure 1**, the NVG is particularly adept at capturing the “global” landscape and long-range correlations within the time series. A very high peak in the series can be “seen” from many other points, creating long-range connections (hubs) in the resulting graph that correspond to major events or trends.

2.2 Horizontal visibility graph (HVG)

The horizontal visibility graph (HVG) is a more constrained variant of the NVG that captures more localized information. In an HVG, two data points are connected

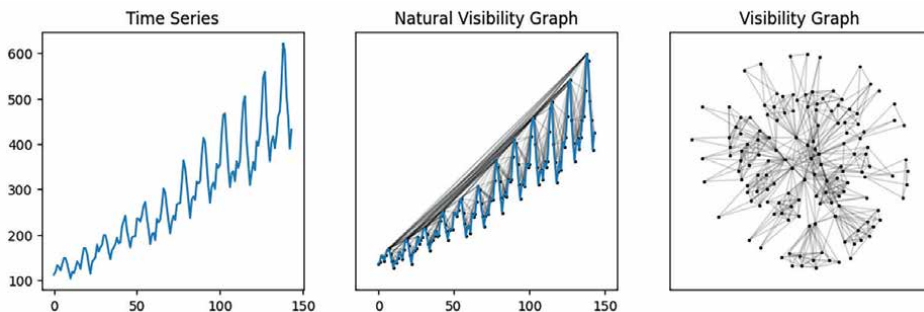


Figure 1.
 Natural visibility graph.

only if a direct horizontal line of sight exists between them, without any intermediate data point having a value greater than or equal to the heights of the two points being connected [6]. Formally, an edge exists between nodes i and j (with $t_i < t_j$) if for any intermediate time t_k ($t_i < t_k < t_j$) the value y_k satisfies the condition $y_k < \min\{y_i, y_j\}$. Due to its stricter definition, the HVG is highly sensitive to the “local” dynamics and short-term volatility of the time series. A data point can be easily hidden by a slightly taller neighbor, even if a much larger peak exists further away in the series.

Due to its stricter definition, the HVG is very sensitive to “local” dynamics and short-term volatility in time series. A data point can easily be hidden by another slightly higher one, even if there is a much larger peak further along in the series. The decision to generate both an NVG and an HVG for each time series is a powerful feature engineering strategy designed to capture information at multiple scales simultaneously. The NVG provides a macro-level view of the main events of the interaction and long-term trends, while the HVG provides a micro-level view of its local volatility and neighborhood-level patterns. By extracting topological features from both graphs for each time series, the framework creates a richer, more nuanced, and more descriptive feature vector than either graph could provide on its own. This dual feature set is a key factor contributing to the framework’s ability to achieve accurate and meaningful separation of behaviors in the subsequent clustering stage, which directly improves its predictive accuracy.

This transformation from a time series to a graph is theoretically sound, as it has been rigorously demonstrated that the mapping preserves essential properties [6]. For instance, periodic time series generate regular or recurrent graphs, random series generate random graphs (with exponential degree distributions), and fractal series generate scale-free networks [6]. This provides a firm theoretical foundation for the subsequent analysis, ensuring that the topological features extracted are meaningful reflections of the underlying dynamics.

2.3 The temporal metagraph: A new mathematical foundation

The theoretical basis for this framework is a new mathematical structure called a “temporal metagraph.” This concept offers a much richer representation of dynamic interactions in complex systems. Formally, a *temporal metagraph* $M = (X, E)$ is defined as a finite set of nodes $X = \{1, \dots, N\}$ connected by a set of edges $E = \{g_1, \dots, g_m\}$ [1]. The revolutionary feature of this structure is that each edge $g_k \in E$ is not a simple weighted link. Instead, each edge is itself a graph, $g_k = (Y_k, \mathcal{E}_k)$, representing the complete time series of interactions between the two nodes it connects [1]. The key innovation lies in how the time parameter is incorporated. Unlike conventional temporal network models, in which time is an external parameter used to segment or divide the network into snapshots, the temporal metagraph incorporates time “implicitly and intrinsically” into the structure of its edges [1]. The entire history of an interaction, with all its fluctuations, periodicities, peaks, and periods of inactivity, is encapsulated in the topology of the graph that constitutes the edge. This represents a significant paradigm shift. Instead of considering time as a sequence of network states, it is conceived as a static structure that can be analyzed holistically. This approach transforms the conventional analytical question “What was the network like at times $t_1, t_2, t_3, \dots$?” into a deeper and more holistic question: “What is the complete structural nature of the interaction between node A and node B during the entire observation period?” The answer to this question is a unique and well-defined

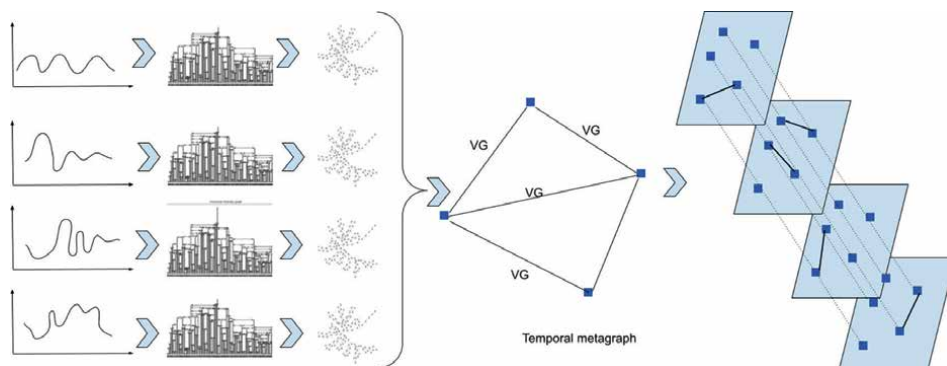


Figure 2.
 Temporal metagraph.

mathematical object: the corresponding edge of the temporal metagraph (i.e., the corresponding “graph-edge”). All process overview can be described in **Figure 2**.

2.4 From theory to practice: Some applications of the visibility graph transformation

In the practical implementation of this framework, the abstract “graph-edges” of the temporal metagraph are realized as visibility graphs. The process begins with the aggregation of raw event data. For any given pair of interacting nodes (e.g., two IP addresses, two city zones), a univariate time series is generated that quantifies their interaction (e.g., bytes sent, number of taxi trips) at a specific frequency (e.g., hourly, daily) over a defined time window. This time series is then transformed into a graph using one of these two methods: The natural visibility graph (NVG) or the horizontal visibility graph (HVG).

3. The k -visibility algorithm: Quantifying and clustering behavioral archetypes

Once the dynamic interactions within a system have been transformed from one-dimensional time series into the topological space of visibility graphs, the next critical step is to quantify these topologies and use them to classify the interactions into distinct behavioral archetypes. This process, formalized as the k -visibility algorithm, combines graph-based feature engineering with unsupervised clustering to achieve an efficient and highly accurate classification of temporal behaviors.

3.1 Feature engineering from visibility graph topology

The basis of the k -visibility algorithm is the creation of a quantitative signature for each interaction. For each time series, the corresponding natural visibility graph (NVG) and horizontal visibility graph (HVG) are generated. From each of these two graphs, a set of theoretical graph metrics is calculated. These metrics are then concatenated to form a single feature vector that numerically describes the dynamics of the original interaction. The various examples developed show that the richness of this feature vector is directly proportional to the quality of the final prediction. In this

regard, it should be noted that the set of features used has evolved through research. The initial applications of the methodology were based on a set of fundamental metrics that captured basic topological properties [1]:

- Maximum degree: The highest number of connections that a single node has in the graph. This metric reflects the most “visible” or influential event in the original time series, such as a major traffic spike or a significant burst of data transfer.
- Link density: The ratio of existing edges to the maximum possible number of edges. It indicates the overall interconnectedness and complexity of the temporal dynamics.

Subsequent research, in particular the study on the Helsinki real estate market, demonstrated that enriching this feature vector with more sophisticated graph metrics leads to substantial improvements in predictive performance. These additional features provide a more nuanced description of the graph structure and, by extension, of the time series dynamics.

3.2 The k-visibility algorithm

The process of clustering is formalized into a straightforward and efficient procedure named the k-visibility algorithm, as we can see in **Figure 3**. It comprises three distinct steps:

1. *Transformation*: Every time series in the dataset is converted into its corresponding pair of visibility graphs (one NVG and one HVG).

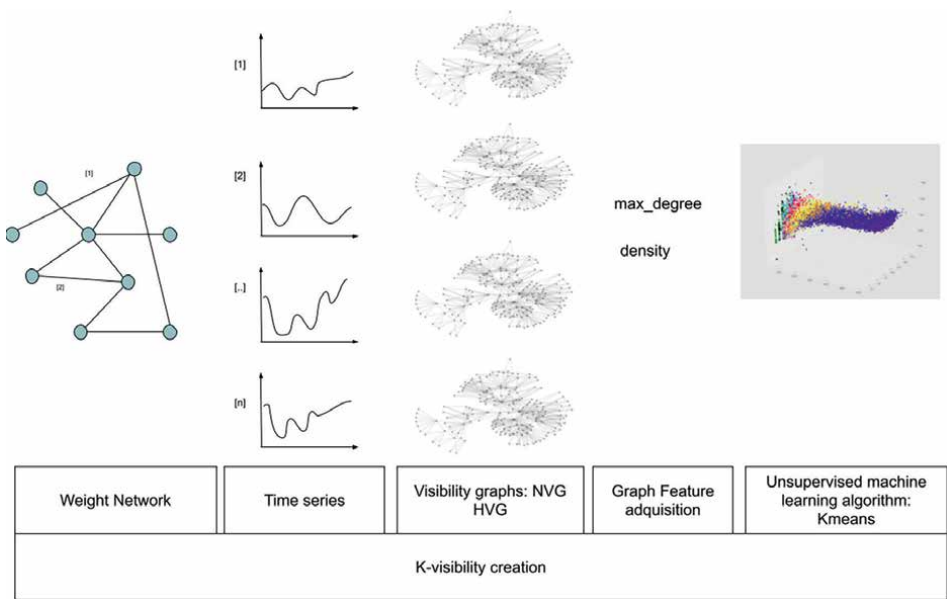


Figure 3.
k-Visibility algorithm.

2. *Feature extraction*: For each time series, its feature vector is computed by calculating the selected graph metrics for both its NVG and HVG and combining them. For example, using five metrics on both graphs would result in a 10-dimensional feature vector for each time series.
3. *Clustering*: A standard unsupervised clustering algorithm, specifically K-Means, is applied to the set of all feature vectors. This partitions the time series into a predefined number, $\mathcal{N}$, of distinct clusters, where each cluster represents a behavioral archetype.

3.3 Performance comparison and architectural advantages

Compared to traditional time series clustering algorithms, the k-visibility approach demonstrates exceptional performance. In a comparative analysis using the prestigious UCR time series archive, the k-visibility algorithm proves to be highly competitive in terms of accuracy, as measured by the Adjusted Rand Index (ARI). In these tests, it achieved the best ARI score on most of the evaluated datasets, outperforming established methods such as K-Shape and those based on Dynamic Time Warping (DTW). However, the most significant advantage of the k-visibility algorithm lies in its computational efficiency. In all comparative tests, it was consistently and demonstrably the fastest algorithm, often by a significant margin. This remarkable speed is not just an implementation detail, but an inherent architectural advantage. Traditional methods such as DTW, which are often used in algorithms such as TimeSeriesKMeans, have a time complexity that is quadratic in the length of the series, denoted as $O(m^2)$, since they must calculate a cost matrix by comparing each pair of data points between two series. The k-visibility framework fundamentally decouples the computational cost of clustering from the length of the time series. The initial transformation to visibility graphs is computationally efficient. The subsequent calculation of the graph metrics is relatively fast. The crucial step is the final K-Means clustering, which operates on a small fixed-size feature vector (e.g., a vector of 4 or 10 features). The size of this vector is independent of the length of the original series, which means that the cost of clustering remains low, regardless of whether the input series has a 100 or a 100,000 data points. This architectural decoupling is the key innovation that explains the significant performance improvements and makes the entire framework scalable to the Big Data challenges for which it was designed.

4. The Temporal Behavior Multiplex Network: A functional system decomposition

The culmination of the data processing and abstraction pipeline is the construction of a final analytical structure: the *Temporal Behavior Multiplex Network*. This network is not a temporal representation in the conventional sense but rather a powerful abstraction that organizes the system's interactions according to their underlying dynamic archetypes. It serves as the platform upon which the final, high-accuracy predictions are made.

4.1 From clusters to layers: Constructing the multiplex network

The construction of the multiplex network, a direct consequence of the results of the clustering phase, is defined as follows [1]:

- *Nodes*: The nodes of the multiplex network are the original and fundamental entities of the system being analyzed. They can be IP addresses in a computer network, taxi zones in a city, or postal codes in a real estate market.
- *Layers*: Several different layers are created, equal in number to the number of clusters, N , identified by the k -visibility algorithm in the previous step. If the algorithm grouped the interaction time series into six different behavioral clusters, the resulting multiplex network will have six layers.
- *Edge placement*: The basic principle of construction lies in the distribution of the edges. An edge representing the interaction between two nodes is placed in the specific layer corresponding to the behavioral cluster to which its associated time series has been assigned. For example, all interactions showing a pattern of “weekday work activity” will reside in one layer, while all interactions characterized by a pattern of “weekend leisure” will be placed in another.

4.2 A functional, not temporal, decomposition

It is essential to highlight that this structure, the Temporal Behavior Multiplex Network, is fundamentally different from conventional temporal or time-interval networks. Its layers do not represent snapshots or time intervals. Instead, each layer represents a *temporal behavior archetype*. The network provides a functional decomposition of system interactions, categorized according to their inherent dynamic patterns. This allows the analyst to go beyond the simple question of who interacts with whom and instead investigate how interactions have occurred in terms of their rhythm, periodicity, and explosiveness over time. It is a structural representation of the functional dynamics of the system.

4.3 Final feature extraction: The behavioral fingerprint

The last step before final predictive modeling is to extract a summarized but powerful set of features for each node in this rich multilayered structure. The main and most effective feature used in the documented studies is the degree of a node in each of the N layers. This process generates a feature vector for each node, where each element of the vector quantifies the participation of that node in a specific type of temporal behavior. In a multiple network with six layers, each node is described by a six-dimensional feature vector, where the k th element is the number of connections (degree) that the node has in layer k . The predictive power of this degree vector, which appears at first sight simple, derives from the fact that the multiple network acts as a sophisticated behavior fingerprinting engine. A single node, such as a specific taxi zone or IP address, participates in numerous interactions, each with its own unique time series.

The k -visibility clustering process classifies each of these individual interactions into a behavioral archetype, which corresponds to a layer of the network. By calculating the degree of the node in each layer, we essentially count how many of its relationships fall into each distinct behavioral category. The resulting vector ($degree_{layer1}, degree_{layer2}, \dots, degree_{layerN}$), serves as a compact, quantitative fingerprint of that node's entire interaction profile. For example, a taxi zone in a city's financial district might show a high degree in the “weekday work activity” layer and a low degree in the “weekend leisure” layer. Conversely, an area in a recreational district would show the

opposite profile. In a cybersecurity context, an attacker's IP address might have a high degree in a layer corresponding to anomalous, low, and slow scanning patterns, while it would have a low degree in layers representing normal user traffic.

This “digital fingerprint” is a highly compressed but information-rich description of the role and function of the node within the complex system. It works as a predictive feature because it has effectively abstracted millions of individual, noisy events into a simple, meaningful, and powerful behavior profile for each actor in the system.

5. Application I: High-fidelity prediction in complex socioeconomic systems

The effectiveness and versatility of this methodological framework are clearly demonstrated by its application to complex socioeconomic systems. By analyzing temporal patterns of human mobility, the model demonstrates its ability to capture latent social and economic dynamics, which it then leverages to make predictions of economic outcomes with remarkable accuracy. This section details two key case studies that validate the framework's performance in this area. In this regard, it is important to note that when analyzed on a large scale, human mobility patterns are not random. They are a direct reflection of the underlying social and economic fabric of a city or region. The daily and weekly rhythms of how, when, and where people move are a powerful indicator of economic activity, social cohesion, and urban land use. The framework presented here is particularly well suited to decoding these signals, transforming raw mobility data into predictive insights.

5.1 Case study 1: Forecasting the Helsinki real estate market

In a pioneering study, the framework was applied to predict real estate price trends in Helsinki, Finland, using a dataset of shared-bicycle trips—a source of data seemingly unrelated to property values [1]. The specific details are as follows:

- *Problem and data:* The objective was to predict the trend of real estate prices for the year 2020. The input data consisted of 3.7 million records of shared-bicycle trips between different city zones (nodes) from April to October 2019.
- *Methodology and results:* The complete five-stage pipeline was implemented. Daily time series of bicycle trips were created for each pair of interacting zones. These series were transformed into NVG and HVG pairs, from which topological features were extracted. The k-visibility algorithm was used to cluster these interactions into behavioral archetypes, which then formed the layers of a Temporal Behavior Multiplex Network. The degree of each city zone in each layer was used as the input feature for a regression model. An initial model, using a basic set of two features per graph and six clusters (layers), achieved a respectable coefficient of determination (R^2) of 0.705.
- *The impact of granularity:* A crucial finding of the study was the dramatic improvement in performance achieved by refining the model's granularity. By increasing the number of topological features to five per graph and expanding the number of behavioral clusters to eight, the model's predictive power surged. The final model achieved a remarkable R^2 of 0.927 and reduced the Mean

Absolute Percentage Error (MAPE) by 57%. This result empirically proves that a more detailed and nuanced behavioral decomposition provides a significantly stronger predictive signal.

The continuous success of these models across different cities, data sources, and scales cannot be attributed to a simple statistical correlation. The framework is successful because it captures latent economic indicators implicit in mobility data. The aggregation of travel data effectively separates different socioeconomic functions. For example, it distinguishes “business district interactions,” characterized by heavy weekday traffic, from “recreational activities,” which predominate on weekends. The economic health and attractiveness of a neighborhood are directly related to the types of activities it hosts and attracts. A high score in the “business” layer suggests a strong commercial center, while a high score in the “leisure” layer suggests strong cultural or recreational vitality. Changes in these mobility patterns can serve as an early indicator of changes in a neighborhood’s economic evolution. The predictive power of the model derives from the fact that the layered structure of the Temporal Behavior Multiplex Network effectively quantifies these different socioeconomic functions for each neighborhood, providing a rich set of predictive features not found in traditional models that rely solely on historical price data.

6. Application II: Advanced behavioral threat detection in cybersecurity

The versatility of the defined framework is clearly demonstrated when applied to a completely different domain with a different set of challenges: cybersecurity. In this context, the goal is to create a new generation of intrusion detection systems (IDSs) that overcomes the critical limitations of traditional approaches and provides a more intelligent, actor-centric view of network threats.

6.1 A rethink of intrusion detection

Traditional IDS platforms have long been a staple of enterprise security, but they face increasing challenges in today’s threat landscape. Most of them are based on static rules and signs, making them ineffective against new 0-day attacks for which no signs yet exist. In addition, they are known to generate a high volume of alerts, a significant portion of which are false positives. This “alert fatigue” overwhelms security analysts, making it difficult to identify real threats among all the noise. Finally, the volume and speed of modern network traffic can overload the processing capacity of these event-by-event analysis systems. The framework presented here facilitates a paradigm shift in threat detection. It moves away from granular, and often noisy, analysis of individual malicious events to holistic identification of malicious actors (e.g., IP addresses) based on their complete behavior profile over time [2, 4, 6].

6.2 Case study: A next-generation IDS

To validate its applicability in this domain, the new framework was used to build a proof-of-concept IDS classifier. Specifically:

- *Problem and data:* The task was to build a classifier capable of identifying attacker IP addresses within a network. The study utilized the public

UNSW-NB15 dataset, a comprehensive collection of 2.5 million labeled network flow records. The IP addresses served as the nodes of the system. To quantify the interaction between any pair of communicating IPs, time series were created using the sbytes (source bytes sent) feature from the flow records.

- *Methodology and results:* The five-stage framework was applied directly. The time series of sbytes for each communicating IP pair were transformed into visibility graphs. The k-visibility algorithm was then used to cluster these interactions into distinct behavioral archetypes. These clusters formed the layers of a Temporal Behavior Multiplex Network, where each layer represented a specific type of communication behavior. The feature vector for each IP address was formed by its degree in each of the multiplex layers. This vector, the “behavioral fingerprint,” was then used to train a Random Forest classifier to predict whether an IP address was an “attacker” or “benign.”
- *The power of granularity in anomaly detection:* The approach proved highly effective, successfully transforming an overwhelming problem of analyzing 2.5 million individual events into a much more manageable classification task for just 49 unique IP addresses. A key finding was the direct and positive relationship between the granularity of the behavioral clustering and the accuracy of the final detection. As the number of clusters (and thus, the number of layers in the multiplex network) was increased from 2 to 12, the accuracy of the model in identifying attacker IPs rose steadily, exceeding 90% with 12 layers. This empirically demonstrates that a more fine-grained behavioral decomposition provides a more powerful and discriminating predictive signal for anomaly detection.

The success of the framework in the cybersecurity domain stems from its inherent ability to operationalize the core principles of User and Entity Behavior Analytics (UEBA). The methodology works by first establishing a robust, data-driven baseline of normal behavior at a systemic level and then identifying significant deviations from that baseline. The vast majority of network traffic is benign and follows predictable temporal patterns (e.g., standard web browsing, periodic database queries, and regular backup operations). The k-visibility clustering algorithm automatically groups these common, high-volume behaviors into large, well-defined “normal” clusters, which form the dominant, high-density layers of the Temporal Behavior Multiplex Network.

In contrast, the activities of malicious actors—such as reconnaissance scanning, command-and-control communication, or data exfiltration—have fundamentally different temporal dynamics. They are often characterized by being low-intensity and slow, highly periodic, or exhibiting unusual bursts of activity that differ starkly from the rhythm of normal traffic. The clustering algorithm effectively isolates these anomalous behaviors into their own, much smaller and sparser clusters. These clusters become the “attack” or “anomalous” layers in the multiplex network. An IP address that has a high degree in one of these anomalous layers is, by definition, exhibiting a suspicious behavioral fingerprint. Its pattern of interaction is a clear outlier when compared to the established norms of the network. This is why the model is so effective: it does not require a preexisting signature for a specific type of attack. It only needs to identify that an IP’s pattern of interaction is fundamentally different from the baseline. This provides a robust and adaptable defense, even against unknown, 0-day attacks that have no predefined signature.

7. Conclusions

This report has detailed a unified methodological framework that innovatively integrates time series analysis and multiplex network theory to model, analyze, and predict the behavior of complex dynamic systems. The process, which begins with the theoretical foundation of the temporal metagraph, uses the visibility graph transformation to encode temporal dynamics into a static topological structure. Subsequently, the k-visibility algorithm performs efficient and accurate behavioral clustering, the results of which are used to construct a Temporal Behavior Multiplex Network. This final structure serves as a powerful analytical and predictive tool, providing a “behavioral fingerprint” for each actor in the system. Therefore, the fundamental contributions and main advantages of this approach, demonstrated through rigorous comparisons and various real-world applications, are diverse:

- *Deep dimensional reduction*: The framework has a proven ability to condense massive event data streams, which can reach several million records, into small, dense, and highly predictive feature vectors, often comprising only a few features per node.
- *Architectural and computational efficiency*: The methodology is demonstrably faster than traditional time series analysis methods, such as those based on Dynamic Time Warping (DTW). This efficiency is not an implementation trick, but an architectural feature that makes the framework viable for Big Data applications where computational cost is a critical constraint.
- *High predictive accuracy*: The framework achieves state-of-the-art predictive performance in real-world, high-stakes settings. It has demonstrated coefficients of determination (R^2) above 0.9 in complex socioeconomic forecasting tasks and classification accuracy above 90% in cybersecurity threat detection.
- *Versatility and generalization*: The successful application of the same basic methodology to domains as diverse as urban mobility, real estate economics, and network security validates its robustness and generalization as a tool for analyzing complex dynamic systems.
- *Behavior-centric analysis*: The framework enables a crucial paradigm shift from event-level analysis to behavior analysis. This provides a more holistic and deeper understanding of system dynamics by focusing on the behavior profiles of entities within the system rather than their isolated actions.

Last but not least, it's worth noting that the field of time series analysis and graph-based machine learning is currently dominated by deep learning models, particularly Graph Neural Networks (GNNs) for spatio-temporal forecasting and Transformers for processing sequential data. These architectures have achieved state-of-the-art results on numerous benchmarks and represent the cutting edge of research. The framework presented in this report should be positioned not as a direct replacement for these models across all tasks, but as a compelling alternative with a distinct and powerful value proposition. While deep learning models offer exceptional predictive

power, they often come with significant drawbacks: they are computationally expensive, require massive labeled datasets for training, and frequently operate as “black boxes,” making their internal decision-making processes difficult to interpret.

On the other hand, the multiplex network framework and k-visibility stand out in areas where deep learning models can fail. Their main strengths are:

1. *Interpretability*: The layers of the multiplex network correspond to understandable behavioral archetypes (e.g., “weekday business traffic,” “anomalous network scanning”), providing a “white box” view of the model’s reasoning.
2. *Computational efficiency*: Thanks to its architectural design, which decouples the cost of clustering from the length of time series, it is significantly faster and less resource-intensive than deep learning models, making it ideal for enterprise environments where efficiency and speed of deployment are critical.
3. *Scalability*: It has been shown to scale effectively to real-world massive event log datasets. In the specific context of cybersecurity, the current trend is toward integrated, artificial intelligence (AI)-driven platforms, such as Palo Alto Networks’ Cortex XSIAM (extended security intelligence and automation management) and CrowdStrike’s Charlotte AI, as well as the widespread adoption of Zero Trust architectures. While the use of multiplex networks for general robustness analysis is an emerging area of research [7], the specific application detailed here—the use of multiplex structure to create actor-level “behavioral fingerprints” for threat detection—remains a novel and highly relevant approach that fits well with UEBA principles.

7.1 Future research

The research documented in this report opens several promising lines for future work, building upon the solid foundation of the existing framework.

- *Methodological refinements*: Future research could explore the use of supervised or semi-supervised clustering techniques to create more domain-specific and finely tuned behavioral layers. Investigating new and more exotic graph-theoretic metrics could also further enrich the feature vectors and potentially increase predictive power.
- *Algorithmic optimization*: While the k-visibility algorithm is already fast, further optimizations in the construction of visibility graphs or the exploration of alternative clustering algorithms to K-Means could enhance performance on petabyte-scale datasets.
- *New applications*: The framework’s demonstrated versatility invites its application to other domains rich in temporal interaction data. Promising areas include financial market analysis (modeling stock-to-stock interactions), climate science (analyzing interactions between different climate monitoring stations), and social media dynamics. The recent evolution of the core geometric concepts into “inverse k-visibility” for robotics and mapping further highlights the broad potential of the underlying theory.

- *Advanced network analysis*: A significant opportunity lies in moving beyond the simple node degree as the final extracted feature. Future work could apply more sophisticated algorithms, such as community detection, node classification, or graph embedding techniques, directly on the final Temporal Behavior Multiplex Network. This could unlock even richer and more powerful features for prediction, providing a deeper understanding of the system's meso-scale structure and dynamics.

In conclusion, the integration of time series and multiplex networks via this novel framework represents a significant and practical advancement in the field of temporal data analysis. It provides a potent, efficient, and versatile methodology for modeling, analyzing, and understanding complex phenomena, opening new avenues for research and application across a wide spectrum of scientific and technological domains.

Acknowledgements

This work was supported by INCIBE/URJC Agreement M3386/2024/0031/001 within the framework of the Recovery, Transformation and Resilience Plan funds of the European Union (Next Generation EU). The authors acknowledge the use of ChatGPT for language editing of the chapter.

Author details

Sergio Iglesias Pérez^{1,2,3} and Regino Criado Herrero^{1,4*}

1 Data, Complex Networks and Cybersecurity Center (DCNC), Rey Juan Carlos University, Madrid, Spain

2 Science, Computing, and Technology Department, School of Architecture, Engineering and Design, Universidad Europea de Madrid, Madrid, Spain

3 Department of Computer Science and Technology, Universidad Internacional de La Rioja, Logroño, Spain

4 Departamento de Matematica Aplicada Ciencia e Ingenieria de los Materiales y Tecnologia Electronica ESCET, Universidad Rey Juan Carlos, Madrid, Spain

*Address all correspondence to: regino.criado@urjc.es

IntechOpen

© 2025 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Iglesias-Perez S, Criado R. Temporal metagraph: A new mathematical approach to capture temporal dependencies and interactions between different entities over time. *Chaos, Solitons & Fractals*. 2023;**175**:113940. DOI: 10.1016/j.chaos.2023.113940
- [2] Iglesias-Perez S, Moral-Rubio S, Criado R. A new approach to combine multiplex networks and time series attributes: Building intrusion detection systems (IDS) in cybersecurity. *Chaos, Solitons and Fractals*. 2021;**150**:111143. DOI: 10.1016/j.chaos.2021.111143
- [3] Iglesias-Perez S, Moral-Rubio S, Criado R. Combining multiplex networks and time series: A new way to optimize real estate forecasting in New York using cab rides. *Physica A*. 2023;**609**:128306. DOI: 10.1016/j.physa.2022.128306
- [4] Iglesias-Perez S, Partida A, Criado R. The advantages of k-visibility: A comparative analysis of several time series clustering algorithms. *AIMS Mathematics*. 2024;**9**(12):35551-35569. DOI: 10.3934/math.20241687
- [5] Iglesias-Perez S, Criado R. Increasing the Effectiveness of Network Intrusion Detection Systems (NIDSs) by Using Multiplex Networks and Visibility Graphs. *Mathematics*. 2023;**11**(1):107. DOI: 10.3390/math11010107
- [6] Lacasa L, Luque B, Ballesteros F, Luque J, Nuno JC. From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences of the United States of America*. 2008;**105**:4972-4975. DOI: 10.1073/pnas.0709247105
- [7] Nana H, Wanga S, Ouyanga C, Zhoua Y, Gua W. Assessing the Robustness and Reducibility of Multiplex Networks with Embedding-Aided Interlayer Similarities. Available from: <https://arxiv.org/pdf/2505.06998>

Section 4

Efficient Learning and Domain Applications

Neotectonic Landforms of Southwest Japan Restored by Numerical Modeling: Significance of Lateral Time-Progressive Motions on the Median Tectonic Line

Yasuto Itoh, Shigekazu Kusumoto and Hirokazu Maekawa

Abstract

Geomorphic development of a convergent margin can be described with numerical simulation. The Median Tectonic Line (MTL), an arc-bisecting fault of southwest Japan, has been right-laterally activated under the recent oblique subduction of the Philippine Sea Plate. A series of dislocation models of the vertical deformation field indicated that the complicated arrangement of uplifting and subsiding areas of the island arc is related to the bent morphology of the regional fault. Our approach successfully restored the neotectonic landforms on this active plate margin. In return, the burial history of sedimentary basins along the MTL constrained the temporal propagation pattern of active segments on the modeled fault, which is probably related to the surface roughness of the converging oceanic plate.

Keywords: dislocation modeling, active landform, transcurrent fault, Philippine Sea Plate, southwest Japan, Median Tectonic Line, Sanbagawa metamorphic rocks

1. Introduction

It is accepted knowledge that oblique subduction of an oceanic plate often results in prevalent simple shear, which provokes regional wrench deformation (e.g., [1]) and eventually cuts off a forearc sliver [2] of the overriding continental plate. The northwestern Pacific margin has been a site of such deformation modes since the late Mesozoic (e.g., [3]). Among the long convergent zones, here we concentrate on the southwestern Japan arc, which is bisected by the Median Tectonic Line (MTL). Previous studies (e.g., [4]) dealt with the formation processes of specific basins on the regional fault and successfully restored tectonic landforms. In this paper, the authors aim to assess more regional crustal strain by means of dislocation modeling. It is a

challenge to decipher a causal relationship between simple fault motions and complicated deformation fields in a mobile belt.

2. Background

2.1 Recent plate tectonic context

Figure 1 presents the tectonic context of southwest Japan. This region is an island arc facing the Philippine Sea Plate, which has shifted its direction of travel counter-clockwise since the late Quaternary [6]. As a result of the transient convergence of the oceanic plate, the arc was bisected by the MTL, which has been activated with dextral slips. The area north of the tectonic line (Inner Zone) is composed of a pre-Cenozoic accretionary complex intruded by voluminous granitic plutons and studded with numerous active faults under E-W compression, whereas the area south of the MTL (Outer Zone) comprises a growing accretionary prism suffering shear deformation [7].

2.2 Long-term deformation modes of the southwestern Japan arc

The MTL is characterized by an exceptionally prolonged history of activity dating back to the Cretaceous era, during which the northwestern Pacific margin was under strong sinistral shear provoked by vigorous oblique convergence of oceanic plates

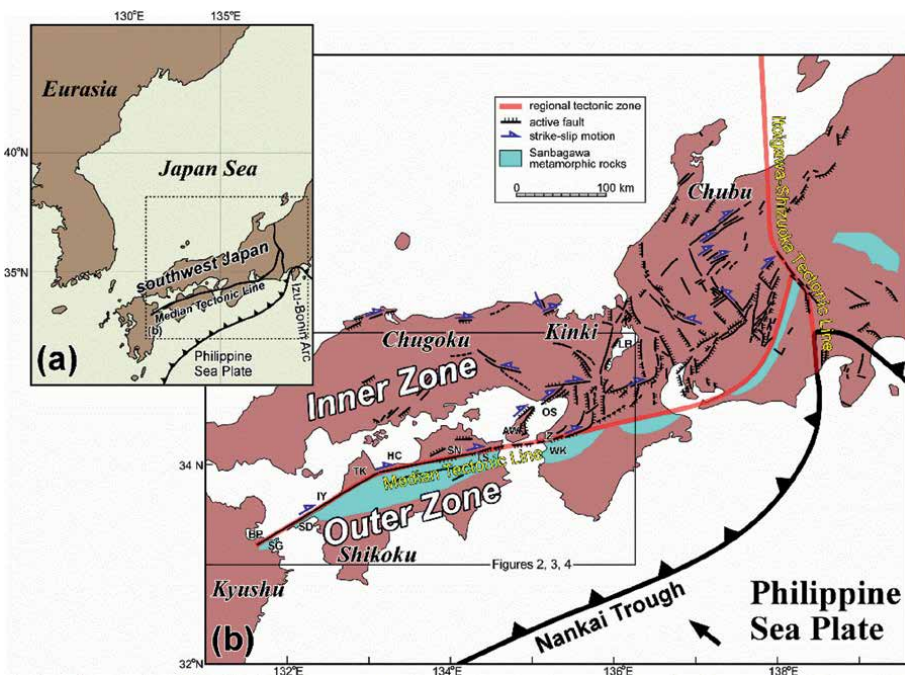


Figure 1. Regional setting of the southwestern Japan arc (a) and significant tectonic elements in the study area (b). Active faults are after Huzita [5]. Abbreviations: BP, Beppu Bay; SG, Saganoseki Peninsula; SD, Sadamisaki Peninsula; IY, Iyonada Sea; TK, Takanawa Peninsula; HC, Hiuchinada Sea; SN, Sanuki Mountains; TS, Tokushima Plain; AW, Awaji Island; OS, Osaka Basin (bay and plain); IZ, Izumi Mountains; WK, Wakayama Plain; LB, Lake Biwa.

(e.g., [3]). Some argue that spreading ridges existed in the Mesozoic Pacific realm and were subducted beneath southwest Japan, which resulted in the rapid exhumation of its forearc region. Wallis and Okudaira [8] stated that the prograde metamorphism of the Sanbagawa belt (see **Figure 1b** for location) is indicative of the involvement of a ridge and/or young slab in orogenesis. Southwest Japan was rifted and drifted from the mother continent as the Japan Sea backarc spread during the Neogene (e.g., [9]). This swiftly migrating continental fragment collided against the Izu-Bonin Arc on the Philippine Sea Plate (see **Figure 1a**), which caused confined indentation around the eastern part (Chubu) of southwest Japan [10]. Deformation raised by the tectonic event is expressed as the curvature of the MTL and the northward cusped shape of the Sanbagawa metamorphic rocks in the Outer Zone. Thus, the metamorphic rocks are regarded as the best indicator of the long-term deformation modes of the study area.

The Sanbagawa metamorphic belt extends east to west for about 800 km from the Kanto Mountains through the Chubu, the Kii Peninsula, and Shikoku to the Saganoseki area in Kyushu. The belt is one of the most extensively studied high-pressure/low-temperature (high P/T) metamorphic belts in Japan. The metamorphic protoliths are dominated by Jurassic to Cretaceous mudstone and volcanoclastic rocks with minor chert, limestone, sandstone, basaltic lava, metagabbro, and ultramafic rocks. The Sanbagawa belt is bounded on the north by a major strike-slip fault, the MTL, which separates it from the Cretaceous high T/P Ryoke belt. The main deformation in the Sanbagawa belt is characterized by ductile flow in an E-W direction sub-parallel to the length of the belt, probably due to the oblique subduction. Kinematic studies suggest that the main ductile deformation caused a 35–65% thinning of the region, which played an essential role in causing exhumation of the region [11]. The belt is in contact with the Triassic to Jurassic Chichibu belt to the south.

After a post-rifting dormant period in the middle Miocene, the southwestern Japan arc suffered strong deformation [12] related to resumed normal subduction of the Philippine Sea Plate. At the same time, the MTL was activated as a reverse fault (e.g., [13]). Thus, the island arc was a site of N-S contractional features during the Pliocene. The old tectonic phases have been studied based on their remarkably cumulative crustal strain, whereas the most recent simple shear phase since the Pleistocene is not yet sufficiently understood. Therefore, we first describe the ongoing deformation using gravity anomaly data in the next section.

2.3 Gravity anomaly

Figure 2a is the Bouguer anomaly map in the study area, based on the gravity database by Komazawa [14]. The Bouguer density is assumed to be 2670 kg/m³. Our study area has low relative gravity because the Pacific Ocean area overall has high relative gravity. The Seto Inland Sea, especially from Beppu Bay (BP in **Figure 1b**) to the Iyonada Sea (IY in **Figure 1b**), is conspicuously a gravity low. In addition, the area reaching from Osaka Bay (OS in **Figure 1b**) to Lake Biwa (LB in **Figure 1b**), the so-called “Kinki-Triangle Area” [15, 16], is also a low-gravity area.

It is known that these low-gravity areas, excluding Lake Biwa, are caused by tectonic sedimentary basins (e.g., [17–21]), and these basins are interpreted as having been formed by activities of the MTL and tectonic lines and faults around the MTL in the Quaternary (e.g., [22, 23]). In fact, in and around these low-gravity areas are distributed tectonic lines and faults such as the MTL, Arima-Takatsuki Tectonic Line (ATTL), Oita-Kumamoto Tectonic Line (OKTL), Nojima Fault (NF), Rokko-Awaji Fault System (RAFS), Uemachi Fault (UF), Ikoma Fault (IF), and Nara Tohen Fault (NTF).

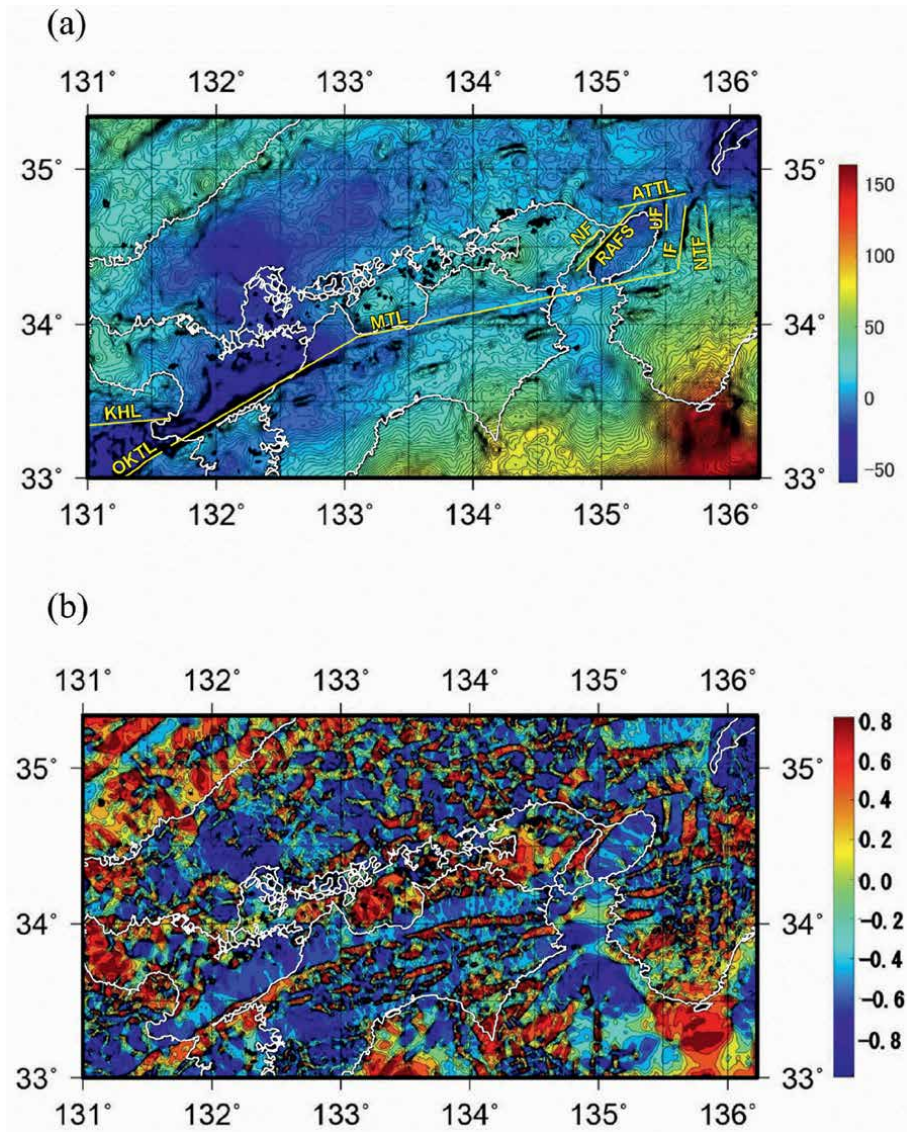


Figure 2. Bouguer anomaly map (a) and Shape Index (SI) image map (b) of the study area. The Bouguer anomaly map is based on the gravity database by Komazawa [14]. The Bouguer density is assumed to be 2670 kg/m^3 . The unit of the Bouguer anomaly is mGal ($1 \text{ mGal} = 1 \times 10^{-5} \text{ m/s}^2$). SI is a shape indicator of the gravitational potential surface defined by Eq. (2) and classifies the potential surface shape as dome ($SI = 1$), ridge ($SI = 0.5$), flat ($SI = 0$), valley ($SI = -0.5$), and bowl ($SI = -1$). Yellow lines in (a) are significant faults and tectonic lines identified with the following abbreviations: KHL, Kurume-Hiji Line; OKTL, Oita-Kumamoto Tectonic Line; MTL, Median Tectonic Line; NF, Nojima Fault; RAFS, Rokko-Awaji Fault System; ATTL, Arima-Takatsuki Tectonic Line; UF, Uemachi Fault; IF, Ikoma Fault; NTF, Nara Tohen Fault.

These are depicted in **Figure 2a**. Furthermore, we can find steep horizontal gravity gradient zones caused by these structures. We can find these steep horizontal gravity gradient zones not only in the areas along faults and tectonic lines but also at boundaries such as metamorphic rock-sedimentary layers and/or basement rock-sedimentary layers in the southern area of the MTL in the Kii Peninsula and Shikoku district (**Figure 1**).

In recent years, the gravity gradient tensor has been applied to estimate subsurface density structures or to extract characteristics of subsurface structures such as the strike direction of a dike, fault dip, and other features (e.g., [24–26]). The gravity gradient tensor Γ is defined by spatial differentiation of gravity anomaly vectors Δg_x , Δg_y , and Δg_z and is described as follows (e.g., [27, 28]):

$$\Gamma = \begin{pmatrix} \frac{\partial \Delta g_x}{\partial x} & \frac{\partial \Delta g_x}{\partial y} & \frac{\partial \Delta g_x}{\partial z} \\ \frac{\partial \Delta g_y}{\partial x} & \frac{\partial \Delta g_y}{\partial y} & \frac{\partial \Delta g_y}{\partial z} \\ \frac{\partial \Delta g_z}{\partial x} & \frac{\partial \Delta g_z}{\partial y} & \frac{\partial \Delta g_z}{\partial z} \end{pmatrix} = \begin{pmatrix} \Delta g_{xx} & \Delta g_{xy} & \Delta g_{xz} \\ \Delta g_{yx} & \Delta g_{yy} & \Delta g_{yz} \\ \Delta g_{zx} & \Delta g_{zy} & \Delta g_{zz} \end{pmatrix} \quad (1)$$

The gravity gradient tensor is symmetric, and the sum of its diagonal components is zero. Thus, the tensor has nine components but only five independent components. Each component of the gravity gradient tensor has been observed by a gravity gradiometer, but we can estimate them by calculation from the gravity anomaly Δg_z (e.g., [29]). This study employed the gravity gradient tensor data calculated from gravity anomaly data, Δg_z , shown in **Figure 2a**, by the technique of Mickus and Hinojosa [29] because gravity gradient measurements covering our study area have not yet been carried out.

Figure 2b is the Shape Index (*SI*) map, which is a helpful indicator for understanding the shape of the gravity potential field. *SI* is defined by four gravity gradient tensor components and is described as follows [30]:

$$SI = \frac{2}{\pi} \arctan \left[\frac{\Delta g_{zz}}{\sqrt{(\Delta g_{xx} - \Delta g_{yy})^2 + 4\Delta g_{xy}^2}} \right] \quad (2)$$

The values of *SI* are distributed over the range -1 to 1 . The *SI* value is used to classify the gravity potential surface into five shapes: dome ($SI = 1$), ridge ($SI = 0.5$), flat ($SI = 0$), valley ($SI = -0.5$), and bowl ($SI = -1$). Although the gravitational potential surface shape and subsurface structure do not necessarily coincide, indicating the need for caution when applying this indicator to actual field data (e.g., [31]), this indicator is helpful for knowing information on the structural shape in the absence of other information because the subsurface structure affects the curvature of the gravitational potential surface.

Figure 2b shows that BP, IY, Takanawa Peninsula (TK), Sanuki Mountains (SN), Tokushima Plain (TS), Izumi Mountains (IZ), Wakayama Plain (WK), and OS in **Figure 1b** have potential surface shapes that are classified from valley to bowl. These potential surface shapes indicate the existence of low-density materials and/or concave structures and suggest that sedimentary layers are continuously distributed in this area. In fact, it has been inferred that the alluvium layer and sedimentary rocks are distributed at the surface in this area and are thickly distributed in the subsurface (e.g., [19, 32, 33]). However, it is not easy to interpret the *SI* value in the TK in general in terms of the existence of low-density material because granitic rocks are exposed at the surface in and around this peninsula.

In contrast, there are E-W direction signals suggesting potential surface shapes classified as ridge to dome in the southern region of the MTL. Such potential surface shapes indicate the existence of high-density materials and/or convex structures. The zonal signals stretching from the Saganoseki Peninsula (SG in **Figure 1b**) to the southern margin of the WK are especially prominent. Similar E-W signals are distributed in regions further south. These are consistent with the area of the Sanbagawa metamorphic belt exposed at the surface, which experienced regional exhumation in the late Mesozoic. In addition, signals classified as ridge to dome are recognized around the eastern side of the Hiuchinada Sea (HC in **Figure 1b**). Granitic and andesitic rocks are exposed at the surface of the islands distributed in this area. The influence of these rocks would appear as the *SI* values.

2.4 Previous studies

2.4.1 Quaternary basin configuration along the MTL

Volumetric analysis of neotectonic sedimentary basins along the MTL was performed by Itoh et al. [22]. On the northwestern coast of Shikoku Island, the Iyonada Sea (IY in **Figure 1b**) is characterized by a remarkably negative Bouguer gravity anomaly (see **Figure 2a**) with a local minimum of approximately -50 mGal. After applying Talwani's method [34] to estimate 2D subsurface structures, they regarded the Iyonada Basin as a half-graben having its deepest axis (ca. 4 km) along the MTL. The total volume of recent sediments burying the basin was calculated to be 7.2×10^3 km³ based on numerical integration.

As for the eastern portion of the MTL, Itoh et al. [22] gave a rough estimate of the volume of the Osaka Basin (OS in **Figure 1b**) as 9.1×10^2 km³ by applying Gauss's theorem [35] and assuming a density contrast of 400 kg/m³ between sediments and basement rocks. Based on a comprehensive interpretation of reflection seismic data, Itoh and Takemura [36] described the 3D architecture of the offshore part (Osaka Bay) as an asymmetric basin with its depocenter (ca. 3 km) along the eastern coast of Awaji Island (AW in **Figure 1b**). The total volume of the basin-filling unconsolidated sediments was determined to be 1.3×10^3 km³ using the Gauss-Legendre numerical integration (e.g., [37]).

2.4.2 Numerical modeling of active landforms

Numerical simulations to reconstruct the patterns of topography or subsurface structures at the Quaternary active terminations of the MTL have been attempted by Kusumoto et al. [38] and Kusumoto et al. [39]. In both numerical simulations, tectonic lines and faults were approximated by a dislocation plane in an elastic isotropic half-space, and surface deformation due to fault motion was compared with basic topography or subsurface structure patterns. Analytical solutions by Okada [40] were employed in the present study. This modeling approach does not consider the phenomena necessary for discussing structure on a geologic time scale, such as weathering, erosion, sedimentation, and viscoelastic effects. Therefore, it focuses on the basic patterns of deformation fields created by fault motions and discusses whether a combination of fault motions can form the structure under examination. In these simulations, the displacement field is often normalized by the absolute value of the maximum displacement to focus on the pattern of the displacement field.

Kusumoto et al. [38] showed that Beppu Bay (BP in **Figure 1b**) was formed as a pull-apart basin due to the right-lateral motion of the MTL and the Kurume-Hiji Line (KHL in **Figure 2a**) after 1.5 Ma, and that the tectonic processes of this region suggested by Itoh et al. [41] from geological viewpoints were also correct from geophysical viewpoints. The MTL and KHL arrangement is right-lateral right stepping. In addition, Kusumoto et al. [42] assumed a two-layer model of basement rock and sedimentary layers. They estimated the gravity anomaly pattern caused by the basement structure reconstructed by Kusumoto et al. [38]. The results also showed that gravity anomalies caused by tectonic structures are generally explained, except for the gravity anomaly caused by the Shishimuta caldera [43].

At the eastern termination of the Quaternary activity of the MTL, the MTL and ATTL have a right-lateral, left-stepping fault arrangement. Although these fault arrangements generally form uplift areas between both faults, Osaka Bay (OS in **Figure 1b**), a tectonic basin, was formed in this area. Kusumoto et al. [39] considered the IF, NTF, and RAFS shown in **Figure 2a** as secondary faults of the MTL and ATTL. They suggested that reverse faulting formed basins in areas that would generally be uplifted. As a result of numerical simulation, it was indicated that the distribution of basins and uplift zones in and around Osaka Bay could be explained by the right-lateral motion of the MTL and ATTL and the reverse motions of these secondary faults.

3. Numerical modeling

3.1 Method

In this study, we employed the dislocation solutions of Okada [40] for reconstructing the topographic pattern along the Quaternary active areas of the MTL. Okada's solutions have often been employed to quantitatively interpret surface or subsurface deformation fields caused by fault motion, geothermal activities, and volcanism (e.g., [44–46]). They focused on topographic differences before and after these events in each case. In the quantitative interpretations, fault parameters are estimated from displacement and/or strain data observed at the surface or subsurface by inversion analysis.

When we focus only on the basic topography pattern, we can use this dislocation modeling to discuss qualitatively whether a combination of some fault motions can form the structures or topography being examined. In addition to the previously mentioned studies (e.g., [38, 47, 48]), this method has also been applied to reconstruct patterns of topography or subsurface structures and validate tectonics or forming processes suggested by geological viewpoints. In this case, they also focused on topographic difference patterns accumulated by repeated fault motions.

In this modeling, the number of faults and their fault parameters are estimated by trial and error, namely forward modeling. Dislocation planes are set up at the locations of existing faults and tectonic lines, and displacement fields are calculated by applying the amount of dislocation found in the literature to the plane. This modeling procedure is useful and practical when the modeling area is not overly large and the faults or tectonic lines responsible for the structure of concern are clear (e.g., [38, 47–49]). However, as in this study, when the modeling area is vast, the number of faults and their fault parameters are set roughly based on the fault distribution and/or the characteristic distribution of the target structures and are corrected by trial and error (e.g., [50]).

As is also well known, faults move repeatedly on geologic time scales. Itoh et al. [47] introduced this effect into their model by superpositioning dislocation solutions. They attempted to reconstruct the basin structure pattern using this method and showed that the basin of concern is a tectonic basin due to the activities of several faults. This method has been shown to explain well both the topography and the distribution of rigid rotation of the crust as inferred from paleomagnetic studies. This method has also been used to verify formation processes for tectonic basins in Hokkaido, Japan, and estimate the amount of fault displacement [48, 50, 51].

In this study, to reconstruct the topography along the MTL, we employed the method proposed by Kusumoto et al. [50] for selecting effective faults and their fault parameters and the method of Itoh et al. [47] for modeling the accumulation of fault movement.

3.2 Modeling parameters

As is well known, the MTL consists of many active faults, and each fault's activity history and characteristics have been studied in detail (e.g., [52]). It is ideal to know the historical fault parameters of each active fault to understand the MTL activity and its activity history. However, the overall motion and segmentation of the fault are more important for modeling regional topographic reconstruction.

During the Quaternary, the MTL is known to have been dominated by right-lateral strike-slip motion. On the other hand, the subduction of the Philippine Sea Plate creates uplift and subsidence areas, namely regional topographic patterns on the continental plate side (e.g., [16]). In a strict discussion, we should include this effect in our modeling. However, because we focused on the topographic pattern along the MTL, we treated fault motions of the MTL and its related faults and tectonic lines. Right-lateral faulting generally forms a characteristic displacement field with two subsidence-uplifted pairs along the fault. In this study, we visually extracted the combination of topographic features that would result in this displacement pattern from topographic maps along the MTL and divided the MTL into four segments.

The structures on which we focused in this study are BP, IY, Sadamisaki Peninsula (SD in **Figure 1b**), TK, HC, SN, TS, Awaji Island (AW in **Figure 1b**), OS, IZ, and WK. These structures are all along the MTL (see **Figure 1b**). We segmented the MTL into four parts to model or reconstruct these structures (**Figure 3**).

MTL1 (red) is a segment that requires the formation of two subsidence-uplifted pairs [i.e., BP-SG and the valley area of Shikoku Mountains (SH in **Figure 3**)-TK]. In addition, the eastern end of MTL1 is also the bend of the MTL. MTL2 (yellow) is a segment that requires the formation of two subsidence-uplifted pairs (i.e., HC-central SH and IZ-WK). The western end of MTL2 is located at the bend of the MTL. MTL3 (yellowish green) is a segment that requires the formation of two subsidence-uplifted pairs (i.e., IY-SD and the valley area of SH-TK). MTL4 (pink) is a segment that requires the formation of two subsidence-uplifted pairs (i.e., HC-central SH and TS-SN). Because the KHL (blue), IF (black), ATTL (gray), and RAFS (brown) have been introduced in previous modeling for the restoration of BP, AW, and OS [38, 39], we also introduced them in this modeling (**Figure 3**).

In this study, we assumed that all fault tops reach the surface, and their bottom depth is 15 km. In addition, we estimated the fault dip and cumulative dislocations of each fault and tectonic line, which are required to reconstruct the target structures by trial and error.

Although it has been pointed out that the MTL has a low dip at depth by seismic surveys (e.g., [32, 54, 55]), lateral strike-slip faults generally can only move at high

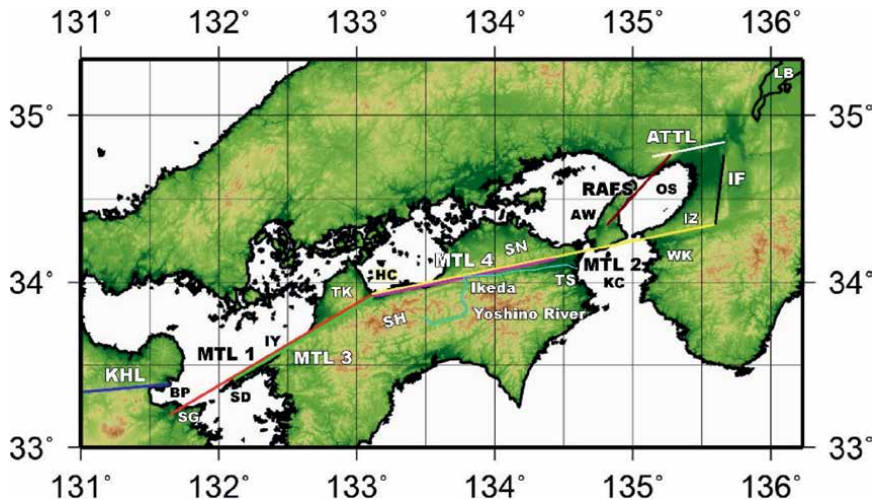


Figure 3.
Location map of the faults and tectonic lines assumed in this study. MTL1, MTL2, MTL3, and MTL4 are the four assumed segments of the MTL. Significant tectonic structures and their abbreviations are as follows: KHL, Kurume-Hiji Line; RAES, Rokko-Awaji Fault System; ATTL, Arima-Takatsuki Tectonic Line; IF, Ikoma Fault; BP, Beppu Bay; SG, Saganoseki Peninsula; SD, Sadamisaki Peninsula; IY, Iyonada Sea; TK, Takanawa Peninsula; SH, Shikoku Mountains; HC, Hiuchinada Sea; SN, Sanuki Mountains; TS, Tokushima Plain; AW, Awaji Island; KC, Kii Channel; OS, Osaka Basin (bay and plain); IZ, Izumi Mountains; WK, Wakayama Plain; LB, Lake Biwa. The topographic map was adapted from the Digital Map 250 m Grid (elevation) by the Geospatial Information Authority of Japan [53].

angles. Also, it is known that there is no subsidence area at the termination of the lateral fault if the lateral fault has a dip lower than 75° (e.g., [56]). Considering these factors, we set the inclination angle of the MTL to 80° based on Kusumoto et al. [38] and Kusumoto [26] (Table 1). In addition, we assumed that the fault dip of the IF and RAES is 80° based on Kusumoto et al. [39]. As is well known, this fault dip is not reasonable as a reverse fault, as described in [39], but it is necessary to restore the structures.

In the modeling of topography or subsurface structures (e.g., sedimentary basins, pull-apart basins, and others) as assumed for an elastic medium, in some cases, a medium with a high Poisson's ratio is assumed (e.g., [57]). This approach covers the

Fault zone	Fault type	Initial length (km)	Total slip (km)	Dip
MTL 1	Right lateral	123.2	50.5	80° N
MTL 2	Right lateral	219.5	54.6	80° N
MTL 3	Right lateral	73.9	49.62	80° N
MTL 4	Right lateral	91.8	54.6	80° N
KHL	Right lateral	100.0	171	80° S
ATTL	Right lateral	44.6	12.8	80° S
IF	Reverse	39.9	2.0	80° E
RAFS	Reverse	61.7	2.0	80° NW

Table 1.
Fault parameters for each fault and tectonic line in the model.

cumulative fault motions or some geological processes at the geological time scale (e.g., weathering and viscoelastic effects) by elastic constants. In this study, it is not necessary to consider the high Poisson's ratio of the medium, and we assumed 0.25 (average for rocks) because the cumulative displacement is reproduced by the superposition of the analytical solutions. In addition, if we set Poisson's ratio to 0.25, it is unnecessary to consider the effect of Young's modulus from the relationship between Young's modulus and Poisson's ratio via Lamé's constants.

3.3 Results

Figure 4a shows the surface vertical displacement pattern obtained with the parameters shown in **Table 1**. The surface vertical displacement pattern shown here is normalized by the maximum absolute value of the cumulative vertical displacement. This normalization was done to reduce the effects, which deformations in the geological time scale were simulated and evaluated by numerical modeling based on elasticity as much as possible. The uplifted areas are shown by warm colors ranging from yellow to red, and the subsided areas are shown by cold colors ranging from green to blue.

Because we simplified the segmentation of the MTL and the segmented fault geometries, restoration of the structure was limited. However, as shown in **Figure 4a**, we find that BP, IY, TK, HC, SN, TS, AW, OS, IZ, and WK have been generally reconstructed by our simplified models. The SD is a subsidence area, but it is an uplift area relative to the IY. In addition, the SG, where the Sanbagawa metamorphic rocks are exposed, is also in an uplift zone.

In this study, from the topography along the MTL, we visually extracted the topographic features explainable by combining four-quadrant-type displacement field patterns formed by lateral faulting, and we assumed active areas (segments) of the MTL to reconstruct these structures. As a result, the MTL was divided into four segments, and large-scale structures along the MTL were generally restored. Of the MTL segments, MTL3 and MTL4 overlap with MTL1 and MTL2. To confirm the importance of these overlaps, we calculated the vertical displacement field when MTL3 and MTL4 were not included in the model.

Figure 4b shows the vertical displacement pattern when the MTL3 and MTL4 segments are removed from the simulation shown in **Figure 4a**. This figure is also normalized by the maximum absolute value of cumulative vertical displacement. The uplifted areas are shown by warm colors, and subsided areas are shown by cold colors. From this figure, we find that the basic structures of the study area (BP, IY, TK, HC, AW, OS, IZ, and WK) can be restored without MTL3 and MTL4 activity. However, if the motion of the MTL3 segment is not considered, the depth of the IY is shallower than that of the BP. Also, if the motion of the MTL4 segment is not considered, then the SN and TS are not formed.

The gravity anomaly map (**Figure 2a**) shows that the gravity low around the IY is of the same magnitude as that of BP. Gravity analysis has shown that BP is a sedimentary basin filled with sediments about 3 km thick (e.g., [20, 32]), and it has been suggested that the sedimentary basin of the IY is also of the same scale [19]. It is possible to assume the normal faulting to the MTL along the IY only from the viewpoint of shape reconstruction [19]. However, the MTL had actually moved as a right-lateral fault during the Quaternary. Therefore, we assigned right-lateral motion to the MTL3 segment, and the sedimentary basin was restored by adding the MTL3 segment to the MTL1 segment.

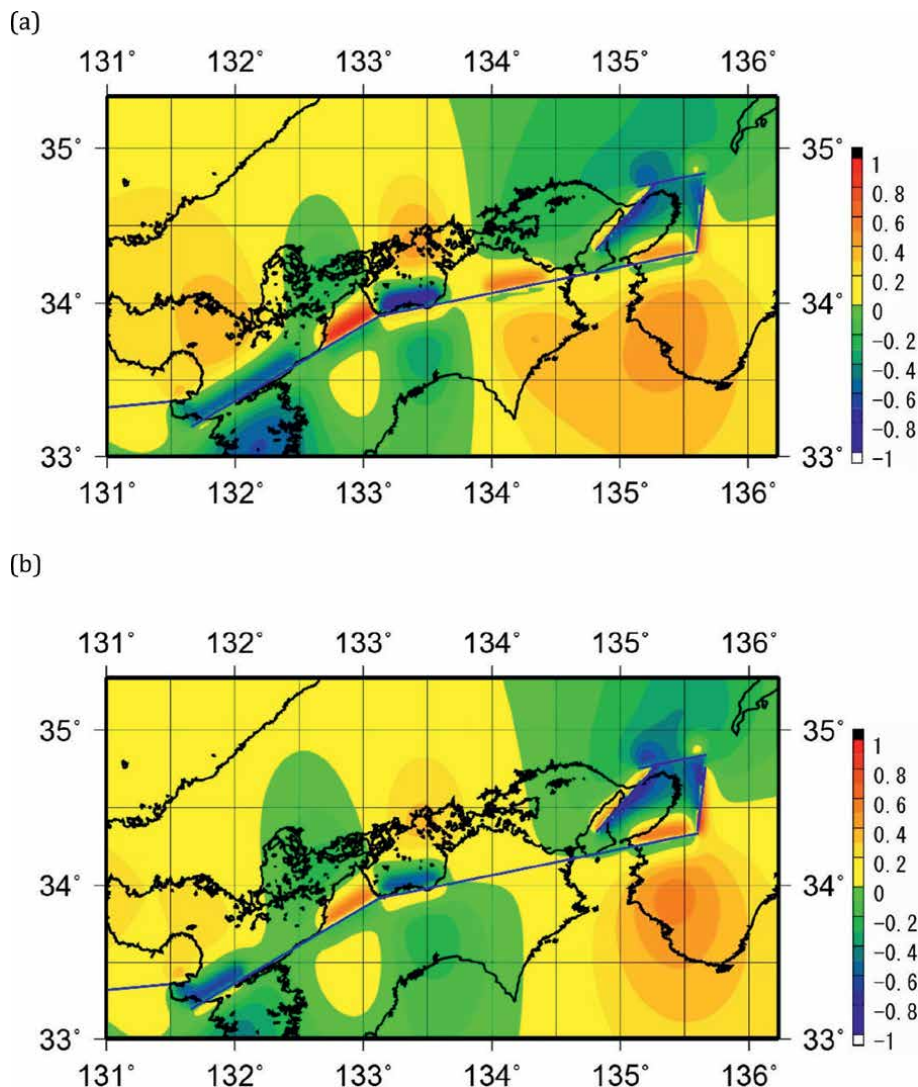


Figure 4. Surface vertical displacement field. The vertical displacement field was normalized by the maximum absolute value of the cumulative vertical displacement. Uplifted areas are shown by warm colors (yellow to red). Subsided areas are shown by cold colors (green to blue). (a) The vertical displacement field is calculated by assuming all fault motions shown in **Table 1**. (b) The vertical displacement field is calculated by excluding MTL3 and MTL4 fault motions from all faults shown in **Table 1**.

In addition, as shown in **Figure 3**, the Yoshino River bends at Ikeda near the MTL (the west end of the Tokushima Plain (TS)) and shifts its direction clockwise from N-S to W-E. The bend of the river is clearly caused by the activity of the MTL. This indicates that the SN and TS are tectonic structures along the MTL as a continuity of their motions. These structures were reconstructed by adding the right-lateral motion of the MTL4 segment to the right-lateral motion of the MTL2 segment.

These results show that the right-lateral motions of the MTL3 and MTL4 segments played an essential role in forming structures along the MTL.

4. Discussion: Quaternary motion history of the MTL

The previous section described how the MTL was divided into four overlapping segments (see **Figure 3**), which was necessary to obtain the best match between the modeled deformation field and actual geomorphological features. From the geological point of view, this assumption corresponds to the fact that the fault segments around Shikoku Island have dextral slips with a longer period. The progressive motion history is probably related to the topographic relief of the subducting plate. Most of southwest Japan is obliquely underthrust by the Shikoku Basin, which is the youngest portion of the Philippine Sea Plate between the Kyushu-Palau Ridge (remnant arc rifted by seafloor spreading) and the Izu-Bonin Arc (**Figure 5**).

As depicted in **Figure 5**, the Shikoku Basin is studded with many bumps, such as the Kinan Seamount Chain. Yamazaki and Okamura [58] pointed out that a strange forearc sag cut by a fault scarp is an expression of wedge deformation linked to W-NW convergence of the Kyushu-Palau Ridge since the counterclockwise shift of relative motion of the oceanic plate in the late Pleistocene [6]. They also argued that the Tosabae Bank (A in **Figure 5**) around the trench slope break along the Nankai Trough is underlain by a subducting seamount based on the forward modeling of geomagnetic anomalies. The presence of the concealed seamount was shown as a seismic velocity image by the later seismological study of Kodaira et al. [59]. Another deformed seamount under the accretionary prism (B in **Figure 5**) was delineated through a multichannel seismic reflection survey by Park et al. [60]. On the basis of these previous findings, we propose that the MTL began its right-lateral motion on Shikoku Island, which has been under intensive shear provoked by strong coupling between the rugged surface of the converging plate and forearc sliver, and then expanded its active segment by means of fault propagation on both terminations.

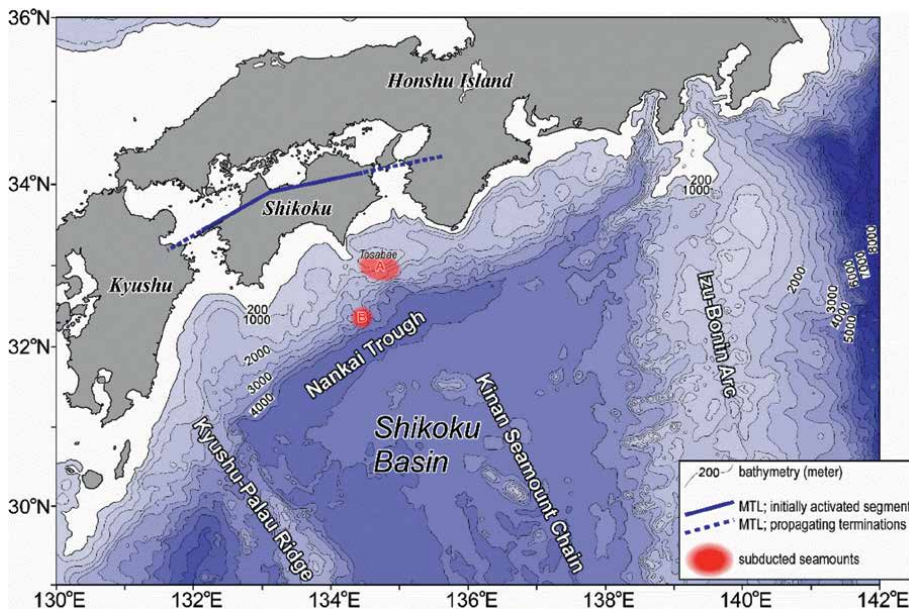


Figure 5. Ocean floor topography around the study area. Positions of subducting seamounts on the forearc region of Shikoku Island are after A = Yamazaki and Okamura [58] and Kodaira et al. [59] and B = Park et al. [60].

5. Conclusions

In this study, from the topography along the MTL, we extracted the topographic patterns explainable by the combination of four-quadrant-type displacement field patterns due to lateral faulting and divided the MTL into four segments. Although the reconstructed displacement (topographic) patterns have shortcomings because the segmentation of the MTL and the fault geometries were simplified in the model, the main structures along the MTL (such as BP, IY, TK, HC, SN, IZ, and WK) were successfully reconstructed by the simplified model shown in this study. In addition, it was shown that the bend area of the MTL had more significant right-lateral motion than the other areas and played an essential role in forming structures.

The point is that reliable numerical simulation stands on precise volumetric analysis of neotectonic basins concealed by recent sediments. For this purpose, 3D architecture around the fault under investigation should be assessed by geophysical measures such as gravity anomaly modeling and/or seismic interpretation. Our integrated approach has elucidated the spatiotemporal evolutionary processes of an arc-bisecting fault, the MTL. Strong forearc coupling on the oblique convergent interface of the east Eurasian margin resulted in strike-slip movements around Shikoku Island in southwest Japan, and stress buildup on both ends of the MTL has given an impetus to the propagation of the terminations of active segments. Thus, the activity of the fault is in a pronounced transient state governed by the changing motion of the Philippine Sea Plate.

Conflict of interest

The authors declare no conflict of interest.

Author details

Yasuto Itoh^{1*}, Shigekazu Kusumoto² and Hirokazu Maekawa¹

¹ Osaka Metropolitan University, Osaka, Japan

² Institute for Geothermal Sciences, Kyoto University, Oita, Japan

*Address all correspondence to: yasutokov@yahoo.co.jp

IntechOpen

© 2025 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Itoh Y, Takemura K. Quaternary geomorphic trends within Southwest Japan: Extensive wrench deformation related to transcurrent motions of the Median Tectonic Line. *Tectonophysics*. 1993;227:95-104
- [2] Fitch TJ. Plate convergence, transcurrent faults, and internal deformation adjacent to Southeast Asia and the western Pacific. *Journal of Geophysical Research*. 1972;77:4432-4460. DOI: 10.1029/JB077i023p04432
- [3] Itoh Y, Takano O, Takashima R. Tectonic synthesis—A plate reconstruction model of the NW Pacific region since 100 Ma. In: Itoh Y, editor. *Dynamics of Arc Migration and Amalgamation—Architectural Examples from the NW Pacific Margin*. Rijeka: InTech; 2017. pp. 93-111. DOI: 10.5772/67358
- [4] Itoh Y, Kusumoto S, Takemura K. Evolutionary process of Beppu Bay in Central Kyushu, Japan: A quantitative study of the basin-forming process controlled by plate convergence modes. *Earth, Planets and Space*. 2014;66:74. DOI: 10.1186/1880-5981-66-74
- [5] Huzita K. Role of the Median Tectonic Line in the Quaternary tectonics of the Japanese islands. *Memoirs of Geological Society of Japan*. 1980;18:129-153
- [6] Nakamura K, Renard V, Angelier J, Azema J, Bourgois J, Deplus C, et al. Oblique and near collision subduction, Sagami and Suruga troughs—Preliminary results of the French-Japanese 1984 Kaiko cruise, Leg 2. *Earth and Planetary Science Letters*. 1987;83:229-242. DOI: 10.1016/0012-821X(87)90068-9
- [7] Taira A. Tectonic evolution of the Japanese island arc system. *Annual Review of Earth and Planetary Sciences*. 2001;29:109-134. DOI: 10.1146/annurev.earth.29.1.109
- [8] Wallis SR, Okudaira T. Paired metamorphic belts of SW Japan: The geology of the Sanbagawa and Ryoke metamorphic belts and the Median Tectonic Line. In: Moreno T, Wallis S, Kojima T, Gibbons W, editors. *The Geology of Japan*. London: Geological Society; 2016. pp. 101-124. DOI: 10.1016/0191-8141(95)00005-X
- [9] Otofujii Y, Matsuda T. Amount of clockwise rotation of Southwest Japan—Fan shape opening of the southwestern part of the Japan Sea. *Earth and Planetary Science Letters*. 1987;85:289-301. DOI: 10.1016/0012-821X(87)90039-2
- [10] Itoh Y. Differential rotation of the eastern part of Southwest Japan inferred from paleomagnetism of Cretaceous and Neogene rocks. *Journal of Geophysical Research*. 1988;93:3401-3411. DOI: 10.1029/JB093iB04p03401
- [11] Wallis S. Vorticity analysis and recognition of ductile extension in the Sanbagawa belt, SW Japan. *Journal of Structural Geology*. 1995;17:1077-1093
- [12] Itoh Y, Nagasaki Y. Crustal shortening of Southwest Japan in the Late Miocene. *Island Arc*. 1996;5:337-353
- [13] Itoh Y, Iwata T, Takemura K. Three-dimensional architecture of the Median Tectonic Line in Southwest Japan based on detailed reflection seismic and drilling surveys. In: Itoh Y, editor. *Evolutionary Models of Convergent Margins—Origin of Their Diversity*. Rijeka: InTech; 2017. pp. 55-75. DOI: 10.5772/67434

- [14] Komazawa M. Gravity Grid Database of Japan. DVD Edition, Digital Geoscience Map P-2. Geological Survey of Japan, AIST: Tsukuba; 2013
- [15] Huzita K. Tectonic development of the median zone (Setouti) of Southwest Japan, since the Miocene. *Journal of Geosciences*. Osaka City University. 1962;**6**:103-144
- [16] Huzita K. Tectonic development of Southwest Japan in the Quaternary period. *Journal of Geosciences*. Osaka City University. 1969;**12**:53-70
- [17] Nishida J, Katsura I, Nishimura S. Gravity survey around Lake Biwa, Southwest Japan. *Journal of Physics of the Earth*. 1990;**38**:1-17. DOI: 10.4294/jpe1952.38.1
- [18] Nakagawa K, Ryoki K, Muto N, Nishimura S, Ito K. Gravity anomaly map and inferred basement structure in Osaka Plain, central Kinki, southwest Japan. *Journal of Geosciences*. Osaka City University. 1991;**34**: 103-117
- [19] Ohno I, Kono Y, Hujimoto H, Koizumi K. Gravity anomaly in and around the western Seto Inland Sea and subsurface structure of negative anomaly belt. *Zisin* (2). 1994;**47**:395-401
- [20] Kusumoto S, Fukuda Y, Takemoto S, Yusa Y. Three dimensional subsurface structure in the eastern part of the Beppu-Shimabara Graben, Kyushu, Japan, as revealed by gravimetric data. *Journal of the Geodetic Society of Japan*. 1996;**42**:167-181. DOI: 10.11366/sokuchi1954.42.167
- [21] Kudo T, Yamaoka K. Pull-down basin in the central part of Japan due to subduction-induced mantle flow. *Tectonophysics*. 2003;**367**:203-217. DOI: 10.1016/S0040-1951(03)00128-8
- [22] Itoh Y, Kusumoto S, Takemura K. Characteristic basin formation at terminations of a large transcurrent fault—Basin configuration based on gravity and geomagnetic data. In: Itoh Y, editor. *Mechanism of Sedimentary Basin Formation—Multidisciplinary Approach on Active Plate Margins*. Rijeka: InTech; 2013. pp. 255-272. DOI: 10.5772/56702
- [23] Takemura K, Kusumoto S. Plate tectonics in the Lake Biwa region. In: Kawanabe H et al., editors. *Lake Biwa: Interactions between Nature and People*. 2nd ed. Switzerland: Springer; 2020. pp. 3-8
- [24] Beiki M, Pedersen LB. Eigenvector analysis of gravity gradient tensor to locate geologic bodies. *Geophysics*. 2010;**75**:I37-I49. DOI: 10.1190/1.3484098
- [25] Martinetz C, Li Y, Krahenbuhl R, Braga MA. 3D inversion of airborne gravity gradiometry data in mineral exploration: A case study in the Quadrilatero Ferrifero, Brazil. *Geophysics*. 2013;**78**:B1-B11. DOI: 10.1190/geo2012-0106.1
- [26] Kusumoto S. Eigenvector of gravity gradient tensor for estimating fault dips considering fault type. *Progress in Earth and Planetary Science*. 2017;**4**:15. DOI: 10.1186/s40645-017-0130-0
- [27] Torge W. *Gravimetry*. Berlin: Walter de Gruyter; 1989. 465 p
- [28] Hofmann-wellenhof B, Moritz H. *Physical Geodesy*. Berlin: Springer; 2005. 403 p
- [29] Mickus KL, Hinojosa JH. The complete gravity gradient tensor derived from the vertical component of gravity: A Fourier transform technique. *Journal of Applied Geophysics*. 2001;**46**:159-174. DOI: 10.1016/S0926-9851(01)00031-3

- [30] Cevallos C. Automatic generation of 3D geophysical models using curvatures derived from airborne gravity gradient data. *Geophysics*. 2014;**79**:G49-G58
- [31] Li X. Curvature of a geometric surface and curvature of gravity and magnetic anomalies. *Geophysics*. 2015;**80**:G15-G26. DOI: 10.1190/GEM2015-048
- [32] Yusa Y, Takemura K, Kitaoka K, Kamiyama K, Horie S, Nakagawa I, et al. Subsurface structure of Beppu Bay (Kyushu, Japan) by seismic reflection and gravity survey. *Zisin (2)*. 1992;**45**:199-212
- [33] Kono Y, Nishiyama Y, Inokuchi H. Gravity measurements on islands in eastern part of the Seto Inland Sea, western Japan—A high gravity anomaly belt—. *Journal of the Geodetic Society of Japan*. 2001;**47**:649-658. DOI: 10.11366/sokuchi1954.47.649
- [34] Talwani M, Lamar WJ, Landisman M. Rapid gravity computations for two-dimensional bodies with application to the Mendocino Submarine Fracture Zone. *Journal of Geophysical Research*. 1959;**64**:49-59. DOI: 10.1029/JZ064i001p00049
- [35] Wangen M. *Physical Principles of Sedimentary Basin Analysis*. London: Cambridge University Press; 2010
- [36] Itoh Y, Takemura K. Three-Dimensional Architecture and Paleoenvironments of Osaka Bay—An Integrated Seismic Study on the Evolutionary Processes of a Tectonic Basin. Singapore: Springer Nature Singapore Pte Ltd.; 2019
- [37] Davis PJ, Rabinowitz P. *Methods of Numerical Integration*. 2nd ed. New York: Dover; 2007
- [38] Kusumoto S, Takemura K, Fukuda Y, Takemoto S. Restoration of the depression structure at the eastern part of Central Kyushu, Japan, by means of the dislocation modeling. *Tectonophysics*. 1999;**302**:287-296. DOI: 10.1016/S0040-1951(98)00287-X
- [39] Kusumoto S, Fukuda Y, Takemura K, Takemoto S. Forming mechanism of the sedimentary basin at the termination of the right-lateral left-stepping faults and tectonics around Osaka Bay. *Journal of Geography*. 2001;**110**:32-43. DOI: 10.5026/jgeography.110.32
- [40] Okada Y. Surface deformation due to shear and tensile faults in a half-space. *Bulletin of the Seismological Society of America*. 1985;**75**:1135-1154. DOI: 10.1785/BSSA0750041135
- [41] Itoh Y, Takemura K, Kamata H. History of basin formation and tectonic evolution at the termination of a large transcurrent fault system: Deformation mode of Central Kyushu, Japan. *Tectonophysics*. 1998;**284**:135-150. DOI: 10.1016/S0040-1951(97)00167-4
- [42] Kusumoto S, Fukuda Y, Takemura K. A distinction technique between volcanic and tectonic depression structures based on the restoration modeling of gravity anomaly: A case study of the Hohi volcanic zone, Central Kyushu, Japan. *Journal of Volcanology and Geothermal Research*. 1999;**90**:183-189. DOI: 10.1016/S0377-0273(99)00029-3
- [43] Kamata H. Shishimuta caldera, the buried source of the Yabakei pyroclastic flow in the Hohi volcanic zone, Japan. *Bulletin of Volcanology*. 1989;**51**:41-50
- [44] Miura S, Ueki S, Sato T, Tachibana K, Hamaguchi H. Crustal deformation associated with the 1998 seismo-volcanic crisis of Iwate volcano, northeastern Japan, as observed by a dense GPS network. *Earth, Planets and Space*. 2000;**52**:1003-1008

- [45] Lasserre C, Peltzer G, Crampe F, Klinger Y, Van der Woerd J, Tapponnier P. Coseismic deformation of the 2001 Mw=7.8 Kokoxili earthquake in Tibet, measured by synthetic aperture radar interferometry. *Journal of Geophysical Research*. 2005;**110**:B12408. DOI: 10.1029/2004JB003500
- [46] Hotta K, Kusumoto S, Takahashi H, Hayakawa YS. Deformation source revealed from leveling survey in Jigokudani valley, Tateyama volcano, Japan. *Earth, Planets and Space*. 2022;**74**:32. DOI: 10.1186/s40623-022-01593-7
- [47] Itoh Y, Kusumoto S, Furubayashi T. Quantitative evaluation of the Quaternary crustal deformation around the Takayama basin, central Japan: A paleomagnetic and numerical modeling approach. *Earth and Planetary Science Letters*. 2008;**267**:517-532. DOI: 10.1016/j.epsl.2007.11.062
- [48] Tamaki M, Kusumoto S, Itoh Y. Formation and deformation process of the late Paleogene sedimentary basins in the southern Central Hokkaido, Japan: Paleomagnetic and numerical modeling approach. *Island Arc*. 2010;**19**:243-258. DOI: 10.1111/j.1440-1738.2009.00698.x
- [49] Itoh Y, Kusumoto S, Inoue T. Magnetic properties of siliceous marine sediments in Northern Hokkaido, Japan: A quantitative tectono-sedimentological study of basins along an active margin. *Basin Research*. 2013;**25**:1-13. DOI: 10.1111/bre.12020
- [50] Kusumoto S, Itoh Y, Takano O, Tamaki M. Numerical modeling of sedimentary basin formation at the termination of lateral faults in a tectonic region where fault propagation has occurred. In: Itoh Y, editor. *Mechanism of Sedimentary Basin Formation—Multidisciplinary Approach on Active Plate Margins*. Rijeka: InTech; 2013. pp. 273-304. DOI: 10.5772/56558
- [51] Itoh Y, Kusumoto S, Maeda J. Cenozoic basin-forming processes along the northeastern margin of Eurasia: Constraints determined from geophysical studies offshore of Hokkaido, Japan. *Journal of Asian Earth Sciences*. 2009;**35**:27-33. DOI: 10.1016/j.jseas.2008.11.011
- [52] Research Group for Active Faults. *The Active Faults in Japan: Sheet Maps and Inventories*. Revised ed. Tokyo: University of Tokyo Press; 1991. 437 p
- [53] Geospatial Information Authority of Japan. Digital Map 250 m Grid (elevation). Tsukuba: Geospatial Information Authority of Japan; 1997
- [54] Yoshikawa S, Iwasaki Y, Ikawa T, Yokota H. Geological structure of the MTL in west Wakayama by reflection seismic study. *Memoirs of Geological Society of Japan*. 1992;**40**:177-186
- [55] Ito T, Ikawa T, Adachi I, Isezaki N, Hirata N, Asanuma T, et al. Geophysical exploration of the subsurface structure of the Median Tectonic Line, East Shikoku, Japan. *Journal of the Geological Society of Japan*. 1996;**102**:346-360
- [56] Kusumoto S, Takemura K, Itoh Y. Inconsistent structure and motion of the eastern Median Tectonic Line, southwest Japan, during the Quaternary. In: Itoh Y, editor. *Evolutionary Models of Convergent Margins—Origin of Their Diversity*. Rijeka: InTech; 2017. pp. 77-99. DOI: 10.5772/67964
- [57] Katzman R, ten Brink US, Lin J. Three-dimensional modeling of pull-apart basins: Implications for the tectonics of the Dead Sea Basin. *Journal of Geophysical Research*. 1995;**100**:6295-6312. DOI: 10.1029/94JB03101

[58] Yamazaki T, Okamura Y. Subducting seamounts and deformation of overriding forearc wedges around Japan. *Tectonophysics*. 1989;**160**:207-229. DOI: 10.1016/0040-1951(89)90392-2

[59] Kodaira S, Takahashi N, Nakanishi A, Miura S, Kaneda Y. Subducted seamount imaged in the rupture zone of the 1946 Nankaido earthquake. *Science*. 2000;**289**(5476):104-106. DOI: 10.1126/science.289.5476.104

[60] Park J-O, Tsuru T, Kaneda Y, Kono Y, Kodaira S, Takahashi N, et al. A subducting seamount beneath the Nankai accretionary prism off Shikoku, southwestern Japan. *Geophysical Research Letters*. 1999;**26**:931-934. DOI: 10.1029/1999GL900134

Neuro-Symbolic Methods for Times Series Analysis for Edge Computing

*Alberto Pacheco, Mariano Rivera, Ever Flores,
Raymundo Torres, Raul Chacon and Jose Duran*

Abstract

This chapter explores methodologies for time series analysis based on the Neuro-symbolic Artificial Intelligence paradigm, combining pattern recognition, knowledge representation, and logical reasoning abilities to improve the effectiveness and verification of AI applications. Despite its promise, Neuro-symbolic AI is still in its early stages and lacks widespread adoption. Taxonomies play a critical role in organizing this field, providing conceptual frameworks that help researchers understand the underlying architectures, interaction patterns, and functional roles of each component. These classifications not only facilitate comparison between approaches but also highlight emerging design patterns and gaps in current methodologies. In the first section, we provide some foundational notions and review prominent taxonomies that have shaped the discourse around the Neuro-symbolic integration. Building on this foundation, we propose a comprehensive taxonomy that reorganizes the existing classifications into a more coherent and functionally grounded structure. The last section provides some illustrative examples of each taxonomic category. The first approach translates signal spectrograms into symbolized signal token streams, allowing efficient pattern recognition and text processing using a finite automata and formal regular language principles. The second approach integrates neural and symbolic components into a one-liner coding style for real-time processing signal proper for low-cost, low-power, and small edge computing devices. Both methods are comprehensively discussed, highlighting their theoretical foundations, including nuances in real-time edge computing implementation, performance issues, and examples. Finally, some open challenges are discussed, including computational complexity, human-AI trust, and ethical implications, while proposing future directions into the neuro-symbolic intelligence applied for time series analysis.

Keywords: neuro-symbolic taxonomy, spectral analysis, signal symbolization, neural networks, edge computing

1. Introduction

Time-series analysis plays a crucial role in interpreting temporal patterns across domains such as healthcare, finance, retail, transportation, weather forecasting,

cybersecurity, energy, manufacturing, IoT, and monitoring to support forecasting, anomaly detection, and optimization tasks. In recent years, the integration of symbolic reasoning and neural network, known as Neuro-symbolic Artificial Intelligence, has emerged to enhance both the clarity and effectiveness of models dealing with time-dependent data.

We focus on two complementary methodologies: one grounded in symbolic representation and the other based on deep neural architectures. The symbolic approach transforms audio signals into discrete sequences using frequency-based alphabet encoding, enabling the application of regular expressions and formal language techniques. In contrast, the neural approach leverages learned representations derived from the Mel spectrogram, where each time slice serves as a token processed by recurrent or convolutional neural networks. Both methodologies aim to detect specific auditory patterns—such as the two-tones siren effect—using real-time, low-cost, and interpretable mechanisms.

By using symbolic methods to segment and label relevant audio events, and neural networks to capture the underlying temporal dynamics, this neuro-symbolic strategy benefits from both worlds: the precision and interpretability of symbolic models and the flexibility and pattern recognition capabilities of neural networks. Through this hybrid framework, the system can classify and interpret sirens effectively, even when implemented on constrained edge computing platforms.

In this chapter, we detail the theoretical foundations of symbolic signal processing, neural audio modeling, and their integration under a neuro-symbolic taxonomy. We propose a new classification schema that accounts for hybrid systems with loosely or tightly coupled symbolic and neural components. Finally, we present a practical case study in siren recognition, demonstrating how both symbolic pattern detection and deep learning can collaboratively contribute to robust time series analysis.

2. Foundational concepts

In this section, we will introduce some key concepts that might be useful for the rest of the chapter discussion.

AI for time series analysis: numerous empirical investigations have illustrated that machine learning algorithms can significantly improve signal classification and the detection of anomalies within the realm of time series analysis. For example, long-short term memory (LSTM) networks have been applied with skill to identify anomalies in ECG signals, achieving a recall rate of 99 without the need for extensive preprocessing [1]. Random forest regressors have exhibited enhanced accuracy when juxtaposed with conventional models such as ARIMA, and advanced machine learning techniques have been utilized to discern outliers in time series datasets [2]. Moreover, innovative methodologies that combine deep reinforcement learning with variable auto-encoders have been proposed to dynamically discern novel categories of anomalies across diverse domains, thus illustrating the flexibility of machine learning within this field [3]. Recent works have proposed novel methods such as neuro-symbolic time series classification (NSTSC), a neuro-symbolic model that leverages signal temporal logic (STL) and neural network (NN) to accomplish TSC tasks using multiview data representation and expresses the model as a human-readable, interpretable formula [4]. In summary, the amalgamation of machine learning methodologies with time series analytical techniques establishes a comprehensive framework to improve anomaly detection proficiency across a wide array of applications.

Feature engineering: feature extraction is a fundamental process in data analysis, machine learning, and computer vision. It consists of identifying, selecting, and extracting the most relevant information from the original data to facilitate tasks such as classification, pattern recognition, or prediction [5]. To be able to summarize the most important information from a complex data set, you will need to transform the raw data into a set of measurable characteristics or features that the system can understand and work with it more efficiently, making it easier for algorithms to process and learn the data [6]. Selecting and obtaining the relevant features to analyze the data inside the audio signal is a critical step that could have repercussions on the complex, effectiveness, and efficiency of the audio classifier [7]. This task of feature extraction is performed by the classifier designer (feature engineering); however, there are a few methods, deep learning, for example, that are known to bypass manual feature extraction; this step of feature extraction is made inside the model and, therefore, is learned from raw data [8]. The set of relevant features of the audio signal is called 'signal fingerprint' [9], which each relevant feature in that set would include a type of descriptive information that will be a focus of analysis. The most important task of good feature extraction is to magnify the information contained in the signal that is important for the given problem and to reduce as much as possible the information that is not relevant [5].

Signal spectrograms: a signal processing approach involves the extraction of features as well as information from measured digital signals [2–4] using techniques such as fast Fourier transform (FFT), wavelet transforms, and adaptive filtering that enable precise time-frequency analysis. However, traditional DSP methods often struggle with real-time processing, noise resilience, and automated decision-making in complex environments. A digital signal is defined as any discrete physical encoded magnitude that varies with time, space, or as a function of one or more independent variables [10]. After converting some input analog signal by sampling it to obtain a discrete-time quantified set of signal values, various transformations and techniques can be applied, such as domain transformations, signal sampling, binary quantization [11], filtering, compression, and modulation, among other techniques. In the case of audio signal analysis, there are several techniques for feature extraction [12], as shown in **Figure 1**: (a) the *cepstrum* or inverse Fourier transform of the logarithm of the spectrum resulting from the Fourier transform: [insert formula]; (b) the *chromatogram* allows the analysis of the tones (pitch) present in each time unit, obtaining the intensity of each of the 12 musical semitones (chroma) per time period; (c) the *Mel frequency cepstrum* is a spectrogram of the spectrum, based on Mel frequency cepstral coefficients that can be calculated from a logarithmic Mel spectrogram by applying the discrete cosine transform, where the Mel scale is more in accordance with the human auditory perception mechanism; (d) the *logarithmic Mel spectrogram* that applies windowing and logarithmic banks of Mel filters, in which calculation includes window size, stride, number of filters, and number of samples parameters, where the STFT is estimated and multiplied by the Mel filter bank, generating a time series with size $M \times N$, where M is the number of filters used and N is the number of windows. In order to convert to decibels (Mel logarithmic spectrogram) [13], we use

$$\text{MelSpectre}(\tau, \omega) = 10 \log_{10} |\text{FourierSpectrum}\{x(t)\}|. \quad (1)$$

Edge computing: provides an alternative to the cloud computing model, considering the possibility that even hardware restricted devices can act as tiny, low-cost, distributed, mobile, thin clients, or agents that are able to perform real-time local signal

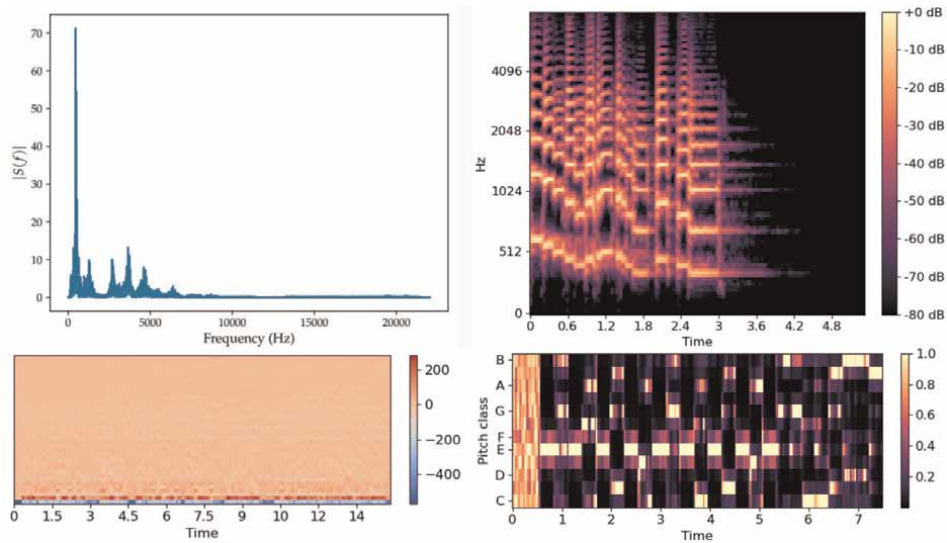


Figure 1.
Spectra examples: cepstrum, log Mel, chromatogram, and Mel cepstrum.

processing tasks closer to where sequential data are generated, preserving information security and privacy, improving system scalability, responsiveness, intelligence and resilience, reducing network bandwidth, providing faster data transfer rates, lowering cloud processing and storage requirements, risks, latencies, and costs. This model is well suited for the most recent, advanced, and cost-effective IoT and AI ecosystems.

3. The neuro-symbolic artificial intelligence paradigm

Neuro-symbolic AI represents an innovative framework that combines the strengths of symbolic logical reasoning AI and neural networks to address the limitations inherent in each approach. However, it encounters challenges in learning from unprocessed data and in adjusting to novel information, capabilities that are inherent to neural networks. With neural network (NN) models learn from data to identify patterns by means of being trained on large datasets, where neural networks can handle pattern identification and learning. With symbolic reasoning, organized decision-making and explainability is ensured. With the integration of both approaches, some limitations can be overcome. Furthermore, more capable systems can be created with the integration of neural network pattern recognition with symbolic AI's logical reasoning [14]. Neuro-symbolic AI combines the pattern recognition capabilities of deep learning (e.g., subsymbolic deep learning, CNNs, Transformers) with the interpretability and rule-based reasoning of symbolic AI (e.g., expert systems, knowledge graphs, rule-based reasoning, or logic programming). Neuro-symbolic AI takes advantage of the uncertainty of data from the real world and the application of clear reasoning rules, thus combining their own characteristics and creating a more powerful and versatile AI system than just separate approaches.

Within IA deep learning (DL) also offers the capability to identify and extract significant patterns and characteristics from high-dimensional data. There are several DL techniques, and one, the recurrent neural networks (RNN s), is designed to handle

sequential data, such as speech, text, or time series data. However, DL has some limitations, such as testability, verifiability, and interpretability, even though DL offers solutions. Under these challenges, the neuro-symbolic paradigm addresses those limited scenarios by combining the robustness of symbolic AI with the use and flexibility of DL [15]. Finally, some open challenges can be discussed, including computational complexity, human-AI trust, and ethical implications of autonomous systems, while proposing future directions for self-healing grids powered by DSP and neuro-symbolic intelligence.

In contrast, neural networks demonstrate remarkable proficiency in pattern recognition and extracting insights from large datasets. However, they often function as “black boxes,” making their decision-making processes difficult to interpret and validate [16, 17]. The integration of these two paradigms within Neuro-symbolic AI aims to leverage the synergistic benefits of both: the powerful learning capabilities of neural networks and the explanatory and inferential strengths of symbolic AI [18, 19].

This combination not only improves the effectiveness of artificial intelligence systems but also improves the testing, evaluation, verification, and validation processes, which are essential for the development of secure and reliable AI applications [16]. Moreover, neuro-symbolic artificial intelligence is considered a critical step toward achieving Artificial General Intelligence (AGI) by enabling systems to assimilate knowledge from data while also engaging in reasoning that encompasses both prior and newly acquired information [20].

Despite its significant potential, neuro-symbolic artificial intelligence is still in an early stage of development and has not yet achieved widespread implementation, partly due to the challenges associated with reconciling the various data representation methods used by symbolic and neural systems [18, 21]. However, the field continues to progress, and current research is focused on developing frameworks and architectural designs that effectively integrate these two paradigms, thus increasing the adaptability and reliability of AI systems [22, 23].

The ongoing advances in neuro-symbolic AI highlight its ability to bridge the gap between interpretability and performance, making it a highly promising area for future research and application across various domains. This potential is particularly noteworthy in fields such as healthcare, education, and autonomous systems, where both reasoning and learning are essential for effective decision making. As academic research progresses, the fusion of these methodologies is expected to lead to the creation of more robust, adaptable, and transparent AI systems capable of addressing complex real-world challenges more effectively.

Prospective developments in neuro-symbolic AI offer the possibility of creating systems that not only learn efficiently but also provide justifications for their decisions, thereby fostering greater trust and accountability in AI applications. The continued exploration of neuro-symbolic AI is likely to produce innovative solutions that enhance both technical capabilities and societal acceptance of AI technologies. As the field evolves, it will be essential to address the challenges related to validation and verification processes to ensure the reliability and safety of neuro-symbolic AI systems [16], ultimately facilitating the establishment of trustworthy AI frameworks that can be effectively deployed in various sectors.

3.1 Taxonomies: Different kinds of neuro-symbolic AI systems

As interest in neuro-symbolic AI continues to grow, efforts to systematically classify the various ways in which neural and symbolic methods can be integrated have

become increasingly important. Taxonomies play a critical role in organizing the field, providing conceptual frameworks that help researchers understand the underlying architectures, interaction patterns, and functional roles of each component. These classifications not only facilitate comparison between approaches but also highlight emerging design patterns and gaps in current methodologies. In this section, we review three prominent taxonomies that have shaped the discourse around neuro-symbolic integration. Each offers a different perspective on how symbolic reasoning and neural networks can be combined, either sequentially, in parallel, or through multiple other interactions. Building on this foundation, we propose a refined taxonomy that introduces a novel category and reorganizes the existing classifications into a more coherent and functionally grounded structure.

3.1.1 Kautz's taxonomy

The Kautz taxonomy [24] has been widely embraced in recent literature [16, 25–30]; it is classified into six categories described below.

- Type 1: *SymbolicNeuroSymbolic*. The first approach represents the standard procedure of deep learning systems for natural language processing, where the input is symbolic in nature, as it consists of words. Then, it is processed by a neural network, which ultimately produces an output that is subsequently mapped back to a symbolic category.
- Type 2: *Symbolic[Neuro]*. Type 2 approach implements a neural network subroutine within a symbolic problem solver. Robotics and autonomous vehicles represent frequent applications of this architecture.
- Type 3: *Neuro|Symbolic*. Type 3 and type 2 are similar, the main difference being that in type 3 the neuro part is a co-routine instead of a subroutine. As an example, a neural network could process complex pattern recognition from noisy raw data and the symbolic system could handle reasoning and decision making by incorporating domain knowledge and rules.
- Type 4: *Neuro : Symbolic → Neuro*. This approach is based on the standard procedure for deep learning systems (similar to Type 1), but incorporates a training strategy grounded in symbolic knowledge. This symbolic information is embedded within the weight structure of the neural network. For example, consider a math solver trained in examples of mathematical operations and their outcomes. Although it can produce correct results for new problems, it does so through pattern recognition rather than by reasoning through the necessary steps.
- Type 5: *NeuroSymbolic*. In this approach, symbolic rules serve as structural templates within the neural network, introducing abstraction and partial hierarchies. Essentially, each symbolic rule is encoded as an embedding that functions as a regularizer within the loss function of the network.
- Type 6: *Neuro[Symbolic]*. According to Kautz, Type 6 represents the most promising approach. It is characterized by a continuously active neural engine that governs the system's operation, with an embedded symbolic reasoning engine that is invoked by the neural component when necessary.

3.1.2 Chaudhuri's taxonomy

Chaudhuri's taxonomy [31] is considerably referenced in the recent literature [28, 32, 33]; it shares Kautz's perspective on several architectures but adds a couple of approaches that are quite relevant.

- Type 1: Symbolic-after-neural pipeline. Chaudhuri's Type 1 can be viewed as equivalent to Kautz's Type 3, in which the output of one system interacts with the other. In this case, a neural component processes the input, and its output is utilized by a symbolic component. A representative example of this interaction is an object detection system based on neural networks, followed by a symbolic reasoning stage dedicated to answering queries.
- Type 2: Neural-after-symbolic pipeline. This approach represents the inverse of Type 1: inputs are initially processed by the symbolic component before being passed to the neural network. It reflects a methodology commonly employed in earlier machine learning paradigms, where manual feature engineering played a more central role, as some symbolic techniques proved particularly effective for pre-processing tasks.
- Type 3: Neural module networks. In the context of visual question answering, a synthesized symbolic representation of the question is used to construct the most suitable configuration of neural modules to solve the task. Although symbolic reasoning is not performed, the structured nature of symbolic representations proves valuable in guiding the network layout predictor.
- Type 4: Algebraic compositions. In this approach, each component operates independently and their output is subsequently aggregated. For example, the result produced by the neural component may be used to refine or adjust the output of the symbolic component.
- Type 5: Neurally accelerated symbolic programs. Chaudhuri characterizes this approach as one in which the neural component is used only to guide the execution of the symbolic system, improving efficiency without affecting correctness. In this configuration, it corresponds to Kautz's Type 2. However, Chaudhuri also added a configuration in which the neural component performs most of the work while the symbolic component functions primarily as a monitor.

3.1.3 Yu's taxonomy

Yu's taxonomy [34] presents a shorter representation, aligns with the views of several other works [16, 27], and shows relevance due to its simple characterization.

- Type 1: Learning for reasoning. Yu's first category corresponds to Kautz's Type 3, although Yu distinguishes between two distinct aspects within it. The first involves using a neural component to constrain the search space of the symbolic component. The second entails the neural engine extracting relevant features from the input data, which the symbolic component then leverages to perform reasoning tasks.

- Type 2: Reasoning for learning. This category aligns with Kautz’s Type 5, in which symbolic systems are used to support the learning process of the neural system, typically encoding symbolic knowledge that can be integrated as a regularization term within the loss function.
- Type 3: Learning-reasoning. This final category is similar to Kautz’s Type 6, with a key distinction. While Kautz frames the neural component as the primary driver that activates the symbolic system as needed, Yu’s Type 6 envisions both paradigms as equal partners engaged in bidirectional interaction. Nevertheless, the core principle is shared in both cases: a continuous, reciprocal interaction between the two systems.

3.2 A multidimensional neuro-symbolic taxonomy

Current neuro-symbolic taxonomies, while insightful, tend to emphasize specific paradigms or use cases, limiting their ability to generalize across the full range of hybrid AI architectures. This narrow focus can obscure alternative forms of integration that fall outside of the dominant patterns. Moreover, most frameworks lack clearly defined classification dimensions, such as interaction type, execution order, or degree of coupling, making it difficult to systematically organize or compare different approaches. In response to these gaps, a new multidimensional structured taxonomy is introduced. By organizing systems along explicit dimensions and accounting for both conventional and less-integrated configurations, the proposed framework aims to provide a clearer, more comprehensive view of the neuro-symbolic design space. The proposed taxonomy is shown in **Figure 2**. It is structured hierarchically across three levels, each informed by principles from software architecture and multi-agent systems. These criteria aim to capture both the structural dependencies and the interaction dynamics between neural and symbolic components. At the first level, systems are categorized as coupled or independent, drawing from architectural classifications based on inter-dependencies.

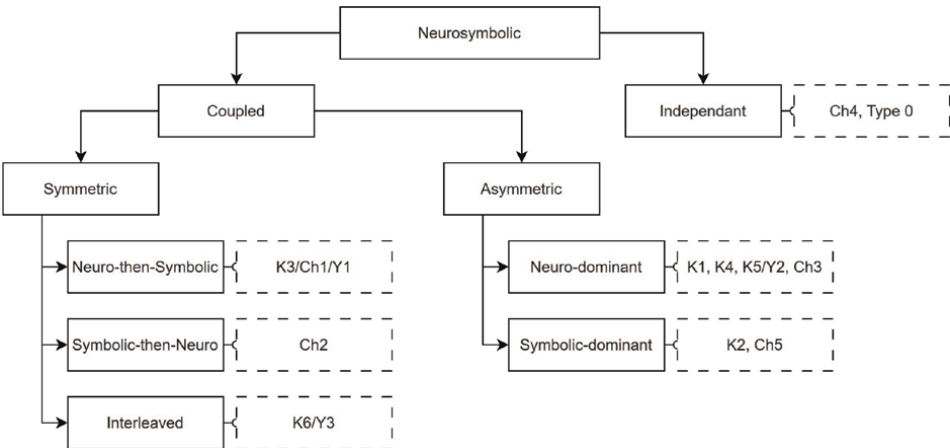


Figure 2. Neuro-symbolic taxonomy. Types from the taxonomies of Kautz, Chaudhuri, and Yu are abbreviated using the author’s initial followed by their type number. Approaches separated by slashes ([image]) denote equivalence; those separated by commas represent distinct types grouped within the same category.

- Coupled: the systems interact directly during the execution process.
- Independent: systems operate in isolation from one another, without direct interaction, though their outputs may still contribute to a shared goal or complementary insights.

The second level further subdivides coupled systems into symmetric and asymmetric configurations, inspired by role-based dynamics in multi-agent systems.

- Symmetric: involve a balanced collaboration, with both components contributing comparably to the task.
- Asymmetric: a primary role is assigned to one component, while the other serves a supporting or auxiliary function.

At the third level, the taxonomy differs in style between Symmetric and Asymmetric systems. For asymmetric systems, the classification remains role-based:

- Neuro-dominant: neural processing constitutes the core of the system's functionality.
- Symbolic-dominant: symbolic processing constitutes the core of the functionality of the system.

For symmetric systems, the focus shifts to execution order, and flow of interaction.

- Neuro-then-symbolic: the neural component precedes the symbolic component in the processing.
- Symbolic-then-neuro: the symbolic component precedes the neural component in the processing.
- Interleaved: both components engage in a bidirectional or iterative manner, alternating interactions throughout the process.

This proposed taxonomy builds on the taxonomies reviewed in Section 3.1, incorporating the additional category introduced in this chapter as Type 0. As a result, the various types of neuro-symbolic system and their corresponding definitions can be systematically aligned with the structure of the new taxonomy, as outlined below.

Symmetrically coupled neuro-symbolic approaches:

- Neuro-then-Symbolic: Kautz Type 3, Chaudhuri Type 1, Yu Type 1. See [35–37].
- Symbolic-then-Neuro: Chaudhuri Type 2. See [38].
- Interleaved: Kautz Type 6, Yu Type 3. See [39].

Asymmetrically coupled neuro-symbolic approaches:

- Neuro-dominant: Kautz Types 1, 4 and 5; Chaudhuri Type 3, Yu Type 2. See [40–44].
- Symbolic-dominant: Kautz Type 2, Chaudhuri Type 5. See [45, 46].

Independent neuro-symbolic approaches: Chaudhuri Type 4, Type 0. See [47].

The proposed taxonomy proved instrumental in the analysis and guide for the development of the case study presented in this chapter. By clearly distinguishing between coupled and independent configurations, as well as the varying roles and interaction styles of neural and symbolic components, the taxonomy highlighted a viable path forward: integrating a symbolic layer to enrich the semantic capabilities of an initially neural-based approach. Beyond informing this specific design choice, the taxonomy also revealed alternative configurations and potential future extensions, enabling a broader understanding of possible neuro-symbolic strategies. In a broader sense, the taxonomy offers a structured lens through which hybrid systems can be conceptualized, enabling clearer reasoning about system architecture, component roles, and execution dynamics. This clarity not only supports more informed design decisions but also facilitates the identification of unexplored configurations and a more consistent evaluation across neuro-symbolic implementations.

4. A study case: Emergency vehicle detection

The case study examined in this chapter consists of two approaches. The first (and general) approach exemplifies a neuro-symbolic approach that diverges from those previously outlined. In this configuration, the neural and symbolic components operate independently, with no integration of their outputs. The systems do not exchange data during execution; however, each contributed meaningful insights to the overall research process, as intermediate outputs from one system informed the development and refinement of the other. A detailed account of this approach is provided in the following sections.

A review of existing taxonomies reveals that the majority of neuro-symbolic approaches involve direct interaction between the neural and symbolic components. Among the approaches analyzed, only Chaudhuri's Type 4 departs from this pattern, each component operating independently and their outputs subsequently aggregated. The general approach presented in this chapter similarly involves independent execution of both components; however, rather than aggregating their output, it leverages each system to generate insights that assist researchers in understanding and addressing the problem, potentially through either component. As this configuration is not represented in existing taxonomies, it is proposed here as a novel category of neuro-symbolic integration, designated as Type 0. The second approach, specific to the neural processing described in subsequent sections, aligns with Type 4 of Kautz's taxonomy, as it employs symbolic inputs to train a neural network.

4.1 Symbolic times series analysis literature review

A hierarchy of various Times Series Analysis representations in the literature is shown in **Figure 1**. For the study case illustrated below, one approach corresponds to the symbolic string representation and the other uses the Discrete Fourier Transform spectra. Symbolic analysis of time series to identify features of interest was applied in [48] to provide an explainable model using probabilistic finite state automata and through the Symbolic Aggregate Approximation (SAX) technique [49, 50], which is a symbolic representation for times series data, where different signal segments are mapped as symbols based on their normal distributions, enabling dimensionality reduction and providing distance measures that lower-bound the original time series

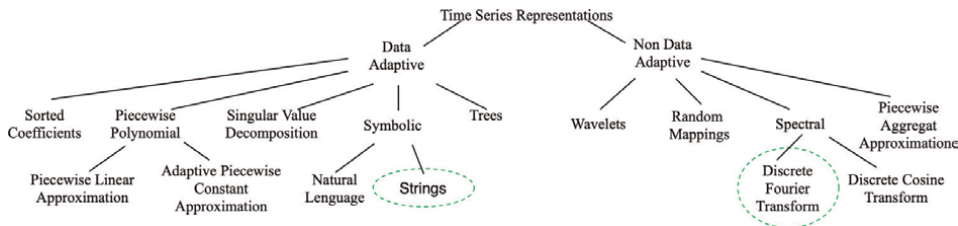


Figure 3.
A taxonomy for different time series representations alternatives taken from [49].

distances, to guarantee no false dismissals in data mining. They used a unique discretization procedure using an intermediate representation between the raw time series and the symbolic strings. First, we transform raw data into a symbolic representation of strings called the Piece Aggregate Approximation (PAA), where the alphabet of these strings is an arbitrary integer, using predetermined breakpoints to map the PAA coefficients into SAX symbols, allowing reducing dimensionality, storage, and computational costs while preserving essential features. SAX introduces a distance measure, which guarantees that the distance between symbolic representation is always less than or equal to the true distance between the original time series. This property is crucial for efficient and accurate data mining. The effectiveness of SAX applies in various data mining tasks, including clustering, classification, anomaly detection, motif discovery, and visualization. Their empirical validation shows that SAX often outperforms or matches traditional methods such as Euclidean distance- and wavelate-based approaches (Figure 3).

4.2 Signal symbolization for emergency vehicle identification

As an illustration example for the neuro-symbolic taxonomy, study case how to identify two-tones or hi-lo siren effect from input acoustic signals. To recognize in real-time audio signals, symbolizing and selecting manually different features from its corresponding log Mel spectrogram, we will consider the following symbol alphabet Σ :

$$\Sigma = \{a, b, n\}. \quad (2)$$

The terminal symbol ‘a’ represents a high tone, the symbol ‘b’ represents a low tone, and the symbol ‘n’ encompasses any value outside each tonal range of interest (noise), as shown in Figure 4.

For a two-tones siren full-cycle feature detection, we can consider the following simplified regular expression one-line definition:

$$hi - lo \text{ siren} = n^+(aabb)^+n^+. \quad (3)$$

For example, given the input sequence ‘nanaabbaabbanbana’, then the previous regular expression will match the substring indicated as follows in boldface: ‘**nanaabbaabban**bana’.

In summary, after transforming (e.g., into energy spectrum, FFT or Mel spectrogram) and symbolizing the input signal, any programming language supporting regular expressions may benefit from this one-liner coding style for real-time signal processing (e.g. classifier or anomaly detection tasks) using low cost, low power, small ARM architecture edge computing devices.

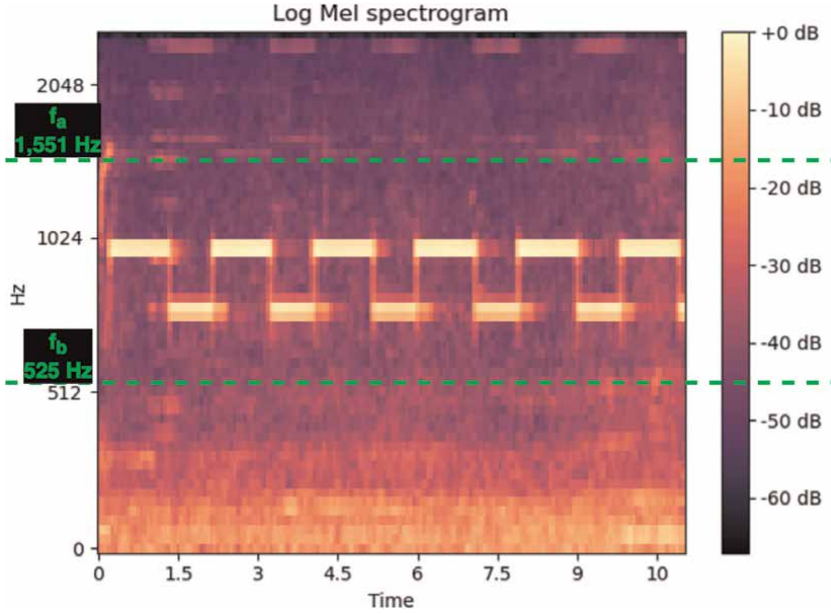


Figure 4.
Mel spectrogram snapshot of an ambulance two-tones siren periodic signal.

5. Recurrent neural network: Based modeling of Mel - Spectrogram sequences

In this approach, we treat the columns of a Mel spectrogram as basic tokens of a time series. Given an input audio signal, its Mel spectrogram can be represented by

$$X = [x_1, x_2, \dots, x_T] \in \mathbb{R}^{M \times T}, \quad (4)$$

where M is the number of Mel frequency bins and T the number of time frames. We denote each column $x_t \in \mathbb{R}^M$ as a ‘token’ representing the spectral content in the time frame t .

Thus, a recurrent neural network (RNN) [51], such as a long short-time memory (LSTM) or gated recurrent unit (GRU), processes this sequence of tokens one frame at a time, maintaining a hidden state that captures temporal dependencies. Formally, at each time step t , the recurrent cell updates its hidden state h_t as

$$h_t = \text{RNN}(x_t, h_{t-1}), \quad (5)$$

where RNN is a recurrent neural network with input, forget and output gates. The initial hidden state h_0 is typically initialized to zero. After processing all T tokens, we obtain a final hidden state h_T which summarizes the entire sequence.

Depending on the downstream task (e.g., classification, regression, or sequence labeling), we transform h_T (or the sequence $\{h_t\}$) through one or more fully connected layers. For a classification task with C classes, we compute

$$\hat{y} = \text{softmax}(W_o h_T + b_o), \quad (6)$$

where $W_o \in \mathbb{R}^{C \times H}$, $b_o \in \mathbb{R}^C$, and H is the dimensionality of the hidden state of the RNN.

The network is trained end-to-end by minimizing a suitable loss function. For classification, we use the categorical cross-entropy loss

$$\mathcal{L} = - \sum_{c=1}^C y_c \log \hat{y}_c, \quad (7)$$

where y is the one-hot ground-truth label. We optimize using backpropagation through time (BPTT) with an adaptive optimizer such as Adam [52], applying gradient clipping to mitigate exploding gradients.

This approach presents the advantages of capture data's Temporal Context, that is, RNNs explicitly capture long-range temporal dependencies across Mel spectrogram frames, which is crucial for tasks like speech recognition and audio event detection. By varying the hidden size H and the number of layers, the capacity of the model and the computational cost can be traded. However, we note that RNNs can be slower to train than convolutional architectures and may suffer from vanishing gradients over very long sequences. Techniques such as residual connections and attention mechanisms can help alleviate these problems [53].

6. Two-dimensional convolutional neural network for Mel spectrogram classification

In this section, we discuss an alternative based on neural networks to recognize with short-time series audio signals encoded in Mel spectrograms. This approach is based on the two-dimensional convolutional neural network (2D-CNN). By treating the Mel spectrogram as a time-frequency 'image', 2D-CNN leverages spatially localized feature extraction to distinguish audio signatures from background noise and other sounds.

Each audio clip is first converted into a Mel spectrogram $S(f, t)$ of dimensions $F \times T$, where F denotes the number of Mel bands (typically 64–128) and T is the number of time frames.

The core of the network comprises a stack of convolutional blocks, each extracting increasingly abstract patterns. Each block consists of:

- *Convolution*: 2D kernels of size 3×3 (or optionally 5×5) slide over the spectrogram, with stride 1 and padding to preserve spatial dimensions. The initial layers may use 32 filters, doubling in subsequent blocks to 64 and 128.
- *Activation*: Rectified linear units (ReLU) introduce non-linearity, enabling the network to model complex spectral shapes.
- *Batch normalization*: Applied after each convolution to reduce the internal covariate shift and accelerate convergence.
- *Max-pooling*: A 2×2 pooling operation reduces the time–frequency resolution by half along each axis, imparting translation invariance to slight shifts in siren onset time or frequency modulation.

Stacking multiple such blocks increases the effective receptive field, enabling the network to capture long-range temporal sweeps characteristic of audio chunks.

Empirically, three to five blocks strike a balance between representation power and computational cost. After the final pooling stage, the tensor of shape $\frac{F}{2^B} \times \frac{T}{2^B} \times C$ (where B is the number of pooling blocks and C the number of channels) is flattened and passed through one or two fully connected layers: *Dense layer(s)*: A 128-unit dense layer with ReLU activation refines the learned features and an *output layer*: A softmax (or sigmoid, for binary classification) outputs the probability that the input contains the audio signal of interest.

The network is trained end-to-end using cross-entropy loss. Key considerations include the following. Use of the Adam algorithm as an optimizer with an initial learning rate of 10^{-3} . A batch size of 32 to 64 ‘images’ spectrograms, balancing gradient stability and GPU memory constraints. Data Augmentation strategy consisting of random brightness and contrast changes of 10% the original values; and a random circular change in the interval $[-10\%, 10\%]$.

The 2D-CNN excels at local feature learning, capturing short-term spectral signatures (e.g., harmonic sweeps), and hierarchical abstraction, learning mid- and long-term patterns through deeper layers. Its weight-sharing and pooling make it computationally efficient for real-time applications. However, careful tuning of depth, kernel sizes, and regularization is essential to avoid overfitting on limited siren datasets and to generalize across different siren types and recording conditions.

7. Discussion

This chapter explores the integration of symbolic representations with neural network processing. Here, we underscore the potential of neuro-symbolic models in addressing complex perceptual tasks such as emergency siren detection. By embedding symbolic inputs, whether derived from structured frequency representations or time-frequency encodings such as Mel spectrograms, into learned or predefined vector spaces, we bridge the gap between symbolic reasoning and subsymbolic learning. This allows neural networks to operate over sequences of tokens that carry semantically relevant information.

Our case study, focused on the recognition of emergency automotive sirens, demonstrates how different embedding strategies can serve as symbolic intermediaries between raw audio signals and neural classifiers. Specifically, the frequency range tokens provide a compact symbolic abstraction of spectral content, while the Mel spectrogram-based tokens offer a temporally resolved view of the signal. Both modalities represent auditory input in symbolic form, enabling the use of sequence-processing architectures.

Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) emerge as particularly suitable for this task, because of their ability to model dependencies over time and space. RNNs capture the temporal evolution of audio events, which is critical to recognizing the characteristic temporal patterns of sirens. CNNs, when applied to spectrogram representations, effectively exploit local time-frequency structures through convolutional filters. Each architecture leverages the structure of the symbolic input differently: RNNs emphasize sequential order, while CNNs focus on local correlations.

An important insight from our exploration is the role of the embedding strategy in shaping the performance and interpretability of the neuro-symbolic system. The learned embeddings allow the model to discover task-specific abstractions, potentially

improving classification performance. In contrast, hand-crafted or constrained embeddings may enhance transparency and offer inductive biases that align with domain knowledge (e.g., known frequency bands used in emergency sirens).

In addition, the neuro-symbolic approach provides a flexible framework that can be extended beyond audio classification. The core methodology, embedding symbolic inputs in vector spaces and processing them with neural architectures, can generalize to other modalities and tasks involving structured data, such as natural language, visual scenes, or structured sensor readings. In doing so, it supports hybrid reasoning paradigms in which neural inference is grounded in symbolic abstraction.

8. Conclusions

Neuro-symbolic AI is a hybrid approach that combines the pattern recognition abilities of neural networks with the logical reasoning of symbolic AI systems [1]. Neuro-symbolic takes advantage of the uncertainty of data from the real world and the application of clear reasoning rules and symbolic, high-level, semantic, and contextual information, thus combining their own characteristics and creating a more powerful and versatile AI system than just separate approaches.

Within IA, deep learning (DL) also offers the ability to identify and extract significant patterns and characteristics from high-dimensional data. There are several DL techniques, and one, the recurrent neural networks (RNNs) is designed to handle sequential data, such as speech, text, or time series data. However, DL has some limitations, such as testability, verifiability, and interpretability even though DL offers solutions. In these challenges, the neuro-symbolic paradigm addresses those limited scenarios by combining the robustness of symbolic AI and the flexibility of DL [6].

We noted that neuro-symbolic models can be used in domains where raw sensory data can be transformed into interpretable symbolic forms. Using the strengths of both symbolic abstraction and neural representation learning, these models present a promising avenue for building intelligent systems that are both data-driven and semantically grounded.

Finally, some open challenges are discussed, including computational complexity, human-AI trust, and ethical implications of autonomous systems, while proposing future directions real-time DSP and neuro-symbolic intelligence.

Acknowledgements

The authors were supported in part by SECIHTI, Mexico (RT, MEng. scholarship grant; MR, Research grant CB-A1-43858), and TecNM, Mexico, Grant 17458.23-P (AP).

Author details

Alberto Pacheco^{1*†}, Mariano Rivera^{2†}, Ever Flores^{1†}, Raymundo Torres^{1†},
Raul Chacon^{1†} and Jose Duran^{1†}

1 Tecnológico Nacional de México, Instituto Tecnológico de Chihuahua, Chihuahua, Mexico

2 Centro de Investigación en Matemáticas AC, Guanajuato, Mexico

*Address all correspondence to: alberto.pg@chihuahua.tecnm.mx

† These authors contributed equally.

IntechOpen

© 2025 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Katikala US, Ranjith R. Detecting anomaly in ECG signal using time series forecast approach. In: *Proceedings of the International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. Piscataway, New Jersey: IEEE Xplore; 2023. pp. 323-328
- [2] Sinha H. Analysis of anomaly and novelty detection in time series data using machine learning techniques. *Multidisciplinary Science Journal*. 2025;7(6):1-14
- [3] Golchin B, Rekabdar B. Anomaly detection in time series data using reinforcement learning, variational autoencoder, and active learning. In: *Proceedings of the 2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*. Piscataway, New Jersey: IEEE Xplore; 2024. pp. 1-8
- [4] Yan R, Ma T, Fokoue A, Chang M, Julius A. Neuro-symbolic models for interpretable time series classification using temporal logic description. In: *Proceedings of the 2022 IEEE International Conference on Data Mining (ICDM)*. Piscataway, New Jersey: IEEE Xplore; 2022. pp. 618-627
- [5] Huzaifah M. Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. *arXiv preprint*
- [6] Boddapati V, Petef A, Rasmusson J, Lundberg L. Classifying environmental sounds using image recognition networks. *Procedia Computer Science*. 2017;112:2048-2056
- [7] Nanni L, Maguolo G, Brahnam S, Paci M. An ensemble of convolutional neural networks for audio classification. *Applied Sciences*. 2021;11(13):5796
- [8] Zinemanas P, Rocamora M, Miron M, Font F, Serra X. An interpretable deep learning model for automatic sound classification. *Electronics*. 2021;10(7): 850
- [9] Zurek E, Gamarra MR, Escorcía JR, Gutierrez C, Bayona H, Pérez RP, et al. Acoustic fingerprint recognition using artificial neural networks. *Polibits*. 2014; 50:63-73
- [10] Soria Olivas E, Martínez Sober M, Francés Villora JV, Camps-Valls G. Tratamiento digital de señales. In: *Problemas y ejercicios resueltos*. Madrid: Prentice-Hall; 2003
- [11] Trirat P, Shin Y, Kang J, Nam Y, Na J, Bae M, et al. Universal time-series representation learning: A survey. *arXiv preprint arXiv:240103717*. 2024. Available from: <https://arxiv.org/abs/2401.03717>
- [12] Sharma G, Umapathy K, Krishnan S. Trends in audio signal feature extraction methods. *Appl Acoust*. 2020;158:107020
- [13] Padhy S, Tiwari J, Rathore S, Kumar N. Emergency signal classification for the hearing impaired using multi-channel convolutional neural network architecture. In: *2019 IEEE Conference on Information and Communication Technology*. Piscataway, New Jersey: IEEE Xplore; 2019. pp. 1-6
- [14] Nawaz U, Anees-ur Rahaman M, Saeed Z. A review of neuro-symbolic AI integrating reasoning and learning for advanced cognitive systems. *Intelligent Systems with Applications*. 2025;26(200541):1-14
- [15] Lu Z, Afridi I, Kang HJ, Ruchkin I, Zheng X. Surveying neuro-symbolic approaches for reliable artificial

intelligence of things. The Journal on Reliable Intelligent Environments. 2024; **10**(3):257-279

[16] Renkhoff J, Feng K, Meier-Doernberg M, Velasquez A, Song HH. A survey on verification and validation, testing and evaluations of neurosymbolic artificial intelligence. The IEEE Transactions on Artificial Intelligence. 2024;**5**(8):3765-3779. Available from: <https://ieeexplore.ieee.org/document/10385139/>

[17] Odense S, d'Avila Garcez AS. A semantic framework for neurosymbolic computation. Artificial Intelligence. 2014;**340**(104273):1-31

[18] Liang B, Wang Y, Tong C. AI reasoning in deep learning era: From symbolic AI to neural-symbolic AI. Mathematics. 2025;**13**(1707):1-42

[19] Velik R, Bruckner D. Neuro-symbolic networks: Introduction to a new information processing principle. In: 2008 6th IEEE International Conference on Industrial Informatics. Piscataway, New Jersey: IEEE Xplore; 2008. pp. 1042-1047

[20] Gibaut W, Pereira L, Grassiotto F, Osorio A, Gadioli E, Munoz A, et al. Neurosymbolic AI and its Taxonomy: A survey. 2023. preprint arXiv:2305.08876. Available from: <https://arxiv.org/abs/2305.08876>

[21] Zhang X, Sheng VS. Bridging the gap: Representation spaces in neurosymbolic AI. 2024. preprint arXiv:2411.04393. Available from <https://arxiv.org/abs/2411.04393>

[22] Monroe D. Neurosymbolic AI. Communications of the ACM. 2022;**65**(10):11-13

[23] Sarker MK, Zhou L, Eberhart A, Hitzler P. Neuro-symbolic artificial

intelligence: Current trends. Ai Communications. 2022;**34**(3):197-209

[24] Kautz HA. The third AI summer: AAAI Robert S. Englemore memorial lecture. AI Magazine. 2022;**43**(1):105-125. Available from: <https://onlinelibrary.wiley.com/doi/10.1002/aaai.12036>

[25] Saha SS, Sandha SS, Aggarwal M, Wang B, Han L, Briseno JDG, et al. TinyNS: Platform-aware Neurosymbolic auto tiny machine learning. ACM Transactions on Computer Systems. 2024;**23**(3):1-48. Available from: <https://dl.acm.org/doi/10.1145/3603171>

[26] Wan Z, Liu CK, Yang H, Li C, You H, Fu Y, et al. Towards cognitive AI systems: A survey and prospective on neuro-symbolic AI. In: Workshop on Systems for Next-Gen AI Paradigms, 6th Conference on Machine Learning and Systems (MLSys). 2023. pp. 1-6. Available from: <http://arxiv.org/abs/522401.01040>

[27] Hamilton K, Nayak A, Božić B, Longo L. Is neuro-symbolic AI meeting its promises in natural language processing? A structured review. Semant Web. 2022;**15**(4):1265-1306. Available from: <https://journals.sagepub.com/doi/full/10.3233/SW-223228>

[28] Hassan M, Guan H, Melliou A, Wang Y, Sun Q, Zeng S, et al. Neuro-symbolic learning: Principles and applications in ophthalmology. arXiv preprint arXiv:2208.00374. 2022. Available from: <http://arxiv.org/abs/2208.00374>

[29] Garcez ADA, Lamb LC. Neurosymbolic AI: The 3rd wave. Artificial Intelligence Review. 2023;**56**(11):12387-12406

[30] Schneider H. A brain-inspired cognitive architecture (BICA) approach

to the neurosymbolic gap. In: Samsonovich AV, Liu T, editors. *Biologically Inspired Cognitive Architectures 2023*. Cham: Springer Nature Switzerland; 2024. pp. 775-786. Available from: <https://link.springer.com/10.1007/978-3-031-50381-884>

[31] Chaudhuri S, Ellis K, Polozov O, Singh R, Solar-Lezama A, Yue Y. Neurosymbolic programming. *Foundations and Trends® in Programming Languages*. 2021;7(3):158-243. Available from: <http://www.nowpublishers.com/article/Details/PGL-049>

[32] Li Z, Huang J, Naik M. Scallop: A language for neurosymbolic programming. *Proceedings of the ACM on Programming Languages*. 2023;7 (PLDI):1463-1487. Available from: <https://dl.acm.org/doi/10.1145/3591280>

[33] Ritchie D, Guerrero P, Jones RK, Mitra NJ, Schulz A, Willis KD, et al. Neurosymbolic models for computer graphics. In: *Computer Graphics Forum*. Vol. 42. No. 2. Weinheim, Germany: Wiley; 2023. pp. 545-568

[34] Yu D, Yang B, Liu D, Wang H, Pan S. A survey on neural-symbolic learning systems. *Neural Networks*. 2023;166: 105-126

[35] Mao J, Gan C, Kohli P, Tenenbaum JB, Wu J. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In: *International Conference on Learning Representations*. 2019. Available from: <https://openreview.net/forum?id=rJgMlhRctm>

[36] Valkov L, Chaudhari D, Srivastava A, Sutton C, Chaudhuri S. HOUDINI: Lifelong learning as program synthesis. *Advances in Neural Information Processing Systems*. 2018;31:1-12

[37] Qu M, Tang J. Probabilistic logic neural networks for reasoning. *Advances in Neural Information Processing Systems*. 2019;32:1-11

[38] Cheung A, Solar-Lezama A, Madden S. Using program synthesis for social recommendations. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. New York, NY: ACM; 2012. pp. 1732-1736

[39] Manhaeve R, Dumancic S, Kimmig A, Demeester T, De Raedt L. Deepproblog: Neural probabilistic logic programming. *Advances in Neural Information Processing Systems*. 2018; 31:1-11

[40] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. 2013. Available from: <http://arxiv.org/abs/1301.3781>

[41] Lample G, Charton F. Deep learning for symbolic mathematics. In: *International Conference on Learning Representations*. 2020. pp. 1-24. Available from: <https://openreview.net/forum?id=S1eZYeHFDS>

[42] Serafini L, Donadello I, d'Avila GA. Learning and reasoning in logic tensor networks: Theory and application to semantic image interpretation. In: *Proceedings of the Symposium on Applied Computing, Marrakech Morocco*: ACM; 2017. pp. 125-130. Available from: <https://dl.acm.org/doi/10.1145/3019612.3019642>

[43] Andreas J, Rohrbach M, Darrell T, Klein D. Neural module networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Piscataway, New Jersey: IEEE Xplore; 2015. pp. 39-48

- [44] Hu Z, Ma X, Liu Z, Hovy E, Xing E. Harnessing deep neural networks with logic rules. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics; 2016. Available from: <https://arxiv.org/abs/1603.06318>
- [45] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, et al. Mastering the game of go with deep neural networks and tree search. *Nature*. 2016;**529**(7587):484-489. Available from: <https://www.nature.com/articles/nature16961>
- [46] Hutter F, Babic D, Hoos HH, Hu AJ. Boosting verification by automatic tuning of decision procedures. In: Formal Methods in Computer Aided Design (FMCAD'07). Piscataway, New Jersey: IEEE Xplore; 2007. pp. 27-34
- [47] Cheng R, Verma A, Orosz G, Chaudhuri S, Yue Y, Burdick JW. Control regularization for reduced variance reinforcement learning. In: International Conference on Machine Learning (PLMR). 2019. pp. 1141-1150
- [48] Li Y, Ray A. Unsupervised symbolization of signal time series for extraction of the embedded information. *Entropy*. 2017;**19**(4):148
- [49] Lin J et al. A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. New York, NY: ACM; 2003. pp. 2-11
- [50] Lin J, Keogh E, Wei L, Lonardi S. Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery*. 2007; **15**(2):107-144
- [51] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*. 1997;**9**(8):1735-1780
- [52] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations. 2015. Available from: <https://arxiv.org/abs/1412.6980>
- [53] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. *Advances in Neural Information Processing Systems*. 2017;**30**

Time Series Prediction Based on Data Compression Methods

Alexander Lysyak

Abstract

This chapter suggests and experimentally investigates new methods of forecasting based on data compression algorithms. The main task is the application of these algorithms to the prediction of processes that generate values from a continuous interval. The chapter presents some theoretical descriptions of the application of data compression algorithms to this case and the practical results of prediction using these algorithms. A new modification, applicable to any forecasting algorithm, is also introduced. It is called the method of alphabet grouping. It allows the use of large alphabets without increasing complexity for forecasting algorithms. Using the examples of predicting currency exchange rates, the consumer price index, and the producer price index, the results show that the precision of these methods is relatively high.

Keywords: R-measure, prediction, CPI, PPI, data compression, financial forecasting, alphabet's grouping, continuous interval

1. Introduction

Methods of forecasting represent great practical interest and allow us to solve a wide range of tasks in science, technology, and economy. The analysis of social, economic, and geophysical processes, and the prediction of natural phenomena and economic events are related to them.

The most widespread statement of a forecasting problem is the prediction of time series, that is, predicting the function defined on an axis of time. This type of forecasting is very significant, because the class of tasks corresponding to it is widely connected with many problems of the economy, geophysics, and other areas of our lives. These methods play a key role in improving efficiency, reliability, and quality of technical, economic, biological, medical, and social systems.

For the last two decades there was a set of forecasting methods, which has shown rather high efficiency. In particular, these methods contain models of machine training [1, 2], the prediction based on a bilinear model [3], the autoregression analysis of various types [4–7], forecasting based on methods of Monte-Carlo [8], methods based on expert estimates (the so-called recursive strategy, see Refs. [9, 10]), forecasting with artificial neural networks [11] and many other things. Despite the existence of such a wide class of forecasting methods, there is a problem of increasing the precision of the forecasts [12].

The main result of this chapter consists of a theoretical description and some practical experimental forecasting results for processes that generate values from a

continuous interval (i.e., the alphabet is infinite and not countable). The offered methods are based on data compression algorithms. In particular, I develop a universal measure and universal coding for applying to a case of such processes' forecasting.

Forecasting methods based on the so-called methods of universal coding, or "data compression" were described in Refs. [13–16]. Forecasting methods based on the so-called methods of universal coding, or "data compression," were described in Refs. [13–16]. This class of methods applies specific techniques of information coding. These methods reduce the total bit size of the historical data used for forecasting. One of the main advantages is their ability to detect hidden patterns, which makes them applicable to a wide range of problems. Informally, the smaller the bit size of the output sequence, the more regularities are present in the historical data (input time series). These methods make it possible to estimate the probability of all possible next symbols (values) by adding new symbols to the end of the input data, and thus to estimate the probabilities of future values of the time series.

In earlier research on forecasting, attention was focused on time series with a finite alphabet. For the first time, the relationship between data compression and forecasting was shown [14]. The given dependence shows how the predicted values depend on the value of the sequence's compression for the case of a finite alphabet. Some theoretical studies on the use of data compression methods for predicting time series that generate values from a continuous interval were presented in Ref. [13]. However, no experimental results have been reported so far.

This chapter proposes an approach for predicting processes that generate values from a continuous interval. This is the main problem addressed here. Two main tasks are considered. First, the so-called "splicing" of different (refined) partitionings is proposed. Second, a solution is suggested for the problem of large alphabets, which arises in the case of a continuous interval.

Until now, no experimental results have been available for this type of time series forecasting. There are only some theoretical findings [13]. This chapter therefore presents both theoretical descriptions and practical experimental results for time series that generate values from a continuous interval.

This chapter describes a new approach to forecasting one-dimensional time series based on certain models of information theory (data compression). In particular, methods based on universal coding are used for predicting the values of stationary and ergodic processes. New approaches are also proposed for combining various forecasting methods, as well as for improving the computational complexity of forecasting without the loss of precision. The results of experimental research and optimization of the proposed method are presented.

The experiments showed interesting findings and a high level of forecasting accuracy. In particular, the best results were obtained for forecasts of inflation indexes: the consumer price index and the producer price index. Their description and results are given in Section 9 titled "Experimental results of forecasting," with subsection 9.2 specifically covering forecasting of inflation indexes.

2. Statement of a forecasting problem

In a general view, the forecasting problem of time series can be formulated as follows. Let there is some source generating sequence of elements $x_1x_2 \dots$ from some set A , called by the alphabet. The alphabet can be both finite and infinite (i.e., to represent some limited continuous interval). Let thus at the moment of time t I have

the finite sequence generated by a source $x_1x_2 \dots, x_t$. The task of forecasting is reduced to a prediction of the element following at the moment of time $(t + 1)$, that is, an element x_{t+1} . In a case when the alphabet A is discrete and finite, any algorithm of forecasting can be applied to this case naturally. “Naturally” means the method as follows. Let us give a finite alphabet A and finite time series $x_1x_2 \dots, x_t$. I want to predict the next value x_{t+1} . For this purpose I give all the letters $a \in A$ and estimate the probabilities $p(a) = F(\bar{x}, a)$, where $\bar{x} = x_1x_2 \dots, x_t$ and F is a prediction function. Then as the predicted value I get that one, which give the maximum probability $p(a)$. More detailed description of this type of methods can be found in Ref. [14, 15].

Let us denote the number of letters in the alphabet as N . It is supposed that process, or information source, is stationary and ergodic, that is, is informal; distribution of probabilities of symbols of this source does not change over time and does not depend on concrete realization of process. Let the source generates the message $x_1 \dots x_{t-1}x_t, x_i \in A, i = 1, 2, \dots, t$, and it is required to predict n of the following elements (in the elementary case – 1 element). A posteriori size of a deviation of a forecast from the valid state of object is called as an error of a forecast. Hereinafter as an forecasting error of n of elements I will understand an average error of a forecast of each of n of elements separately. It is clear that the error of a forecast characterizes the quality of forecasting.

It is obvious that if probability distribution of process outcomes is known in advance, the forecasting problem of the following values is solved rather simply. However in the majority of practical tasks, the described aprioristic data are absent and not always this distribution exists. In this work, I will consider such case. In this situation for the solution of a forecasting problem, it is possible to use the precise estimates of the specified sizes received by means of statistical methods, constructed based on of the interrelation analysis of consecutive outcomes of process and detection regularities.

3. Laplas predictor and its properties

Laplas was one of the first researchers, who has suggested to solve a problem forecasting with use of mathematical apparatus (in particular, probability theory). He has offered the predictor, whose input consists of all the time series and one expected value from the time series alphabet else. There are estimations of an expected symbol's probability's on the output of Laplas predictor.

Laplas predictor is as follows:

$$L_0(a|x_1 \dots x_t) = (\nu_{x_1 \dots x_t}(a) + 1)/(t + |A|), \quad (1)$$

where a is a predictable element (letter); $\nu_{x_1 \dots x_t}(a)$ is the number of occurrences of the letter a in the word $x_1 \dots x_t$, and $x_1 \dots x_t$ is a time series (called also as a word). Proceeding from a type of the predictor, clearly that the probability of each letter a cannot be equal to zero.

Laplas has considered application of this predictor for a solution of the problem of a probability estimation of that the Sun will arise the next day (i.e. tomorrow). Thus, I know that all previous days, how many I have taken, it arises. In this case, the alphabet A consists of two elements: 0 (“The sun will arise”) and 1 (“The sun will not arise”), t is the number of days since Creation.

As a result, I will receive a sequence $x_1 \dots x_{t-1} x_t = 0 \dots 00$. Let us calculate the values of Laplas predictor for the next letters 0 and 1 (for a length of the time series is equal to t):

$$L_0(0|0 \dots 00) = (t+1)/(t+2) = 1 - \frac{1}{t+2};$$

$$L_0(1|0 \dots 00) = \frac{1}{t+2}.$$

As it was described above, the value of Laplas predictor L_0 is connected with a probability of the next letters of the process as follows: The Laplas predictor is some estimation according to probabilities of each next letter and they can be looked as their probabilities and can be used for prediction. Therefore in our example, it is obvious that a probability of “the sun will arise” (which accords to a letter 0) goes to 1. And a probability of “the sun will not arise” goes to 0.

It is important to note that Laplas predictor states an estimation of unknown conditional probabilities of considered stochastic process that is quite natural. In Ref. [17], it is shown that the average error of Laplas predictor at probabilities' estimation of letters on the output of a i.i.d. source of the independent and equally distributed letters asymptotically go to 0 (i.e. $t \rightarrow \infty$).

4. Universal code, universal measure, and its properties

Let us consider the problem of prediction some stationary and ergodic source. And let us have a some arbitrary predictor γ . In particular, as γ I can use Laplas predictor. In Ref. [14], B. Ryabko showed that for any γ there exists a some stationary and ergodic source P such that with probability 1 the average error does not go to a zero infinitely often (see also Ref. [18]). He also suggests to use the Chesaro average:

$$t^{-1} \sum_{i=1}^t \rho_{K,P}(x_1, \dots, x_i),$$

where $\rho_{K,P}(x_1, \dots, x_i)$ is an error of prediction the source P by some measure K , which is defined as follows:

$$\rho_{K,P}(x_1, \dots, x_t) = \sum_{a \in A} P(a|x_1, \dots, x_t) \log \frac{P(a|x_1, \dots, x_t)}{K(a|x_1, \dots, x_t)}$$

It was shown in Ref. [14, 18], that this (Chesaro) average goes to zero for any stationary ergodic source. That's why I will use the Chesaro average in what follows. Also, let us define the average error at time t by

$$\rho^t(P||K) = E(\rho_{K,P}(x_1, \dots, x_t)) = \sum_{x_1, \dots, x_t \in A^t} P(x_1, \dots, x_t) \rho_{K,P}(x_1, \dots, x_t)$$

Analogously, I can define the errors per letter of estimator K as follows:

$$\bar{\rho}^t(P||K) = t^{-1} \sum_{x_1, \dots, x_t \in A^t} P(x_1, \dots, x_t) \log \frac{P(x_1, \dots, x_t)}{K(x_1, \dots, x_t)}$$

In Ref. [18], B.Ya. Ryabko showed that for any i.i.d. source P generating letters from an alphabet A and an integer t the average errors of the Laplace predictor and the average error of the Laplace estimator are upper bounded as follows:

$$\rho^t(P||L_0) \leq \frac{(|A| - 1)\log e}{t + 1}$$

$$\bar{\rho}^t(P||L_0) \leq \frac{(|A| - 1)\log t}{t} + O(1/t)$$

Now, let us consider the methods based on a universal measure, which is connected with universal data compression (it also was suggested in Refs. [9, 13]).

Let us give definition of a universal measure, and also I will explain communication between this approach and the approaches described in the previous part. The measure μ is called as universal if for any stationary and ergodic source P the following equalities are true:

$$\lim_{t \rightarrow \infty} \frac{1}{t} (-\log_2 P(x_1 \dots x_t) - \log_2 \mu(x_1 \dots x_t)) = 0$$

with probability 1, and

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{u \in A^t} P(u) \log_2 (P(u)/\mu(u)) = 0$$

These equalities show that, in a sense, the measure μ is a nonparametric estimation for (unknown) measure P . For this reason, universal measures can be used for an estimation of statistical characteristics of process, and also for an estimation of probabilities of the sequences generated by any stationary and ergodic sources on the finite alphabet.

It is important to note that the Laplace predictor (which is described in a part 2) satisfied the two conditions above and it is a universal measure for Bernoulli sources.

The universal measure is closely connected to concept of a universal code. Let us describe interrelation of these concepts. The code U is called as universal if for any stationary and ergodic source of P the following equalities are true:

$$\lim_{t \rightarrow \infty} |U(x_1 \dots x_t)|/t = H(P),$$

with probability 1, and

$$\lim_{t \rightarrow \infty} E_P(|U(x_1 \dots x_t)|)/t = H(P),$$

where $E_P(f)$ is an average value f in relation to P , and $H(P)$ is an entropy of P according to Shannon, that is,

$$H(P) = \lim_{t \rightarrow \infty} -t^{-1} \sum_{u \in A^t} P(u) \log P(u)$$

Universal measures have deep interrelation with universal codes, and if there is a universal code, it is possible to construct a universal measure on its basis.

The following theorem shows that I can construct a universal measure based on arbitrary universal code.

Theorem 1. Let U is a universal code and

$$\mu_U(\omega) = 2^{-|U(\omega)|} / \sum_{u \in A^{|\omega|}} 2^{-|U(u)|},$$

then μ is a universal measure.

The universal code is called optimum if it codes the sequence of symbols generated by a discrete source in such a manner that the length of the code on the output is asymptotically minimum. Universal codes for stationary and ergodic discrete sources have been described in the 1960th (see reference [19]).

5. The universal measure R and Krichevsky predictor

Let us describe an asymptotically optimal universal measure R from reference [14] which has been used as a basis for a forecasting method in this paper.

After representation of Laplas predictor, there exist the problem of finding such predictor, whose average error of probabilities' estimation for a source of the independent and equally distributed symbols would be minimal. This problem was solved by Krichevsky, which suggested the following predictor:

$$K_0(a|x_1 \dots x_t) = (\nu_{x_1 \dots x_t}(a) + 1/2)/(t + |A|/2), \quad (2)$$

where $\nu_{x_1 \dots x_t}(a)$ is the number of occurrences of the letter a in the word $x_1 \dots x_t$. It is easy to see the similarity of Krichevsky and Laplas predictors.

Further, Krichevsky has generalized the predictor on a measure which is universal for a set of Markov sources with memory, or connectivity $m, m \geq 0$. In a certain way, this measure is optimum for this set that is shown in Ref. [20].

The generalized measure looks as follows:

$$K_m(x_1 \dots x_t) = \begin{cases} \frac{1}{|A|^t}, t \leq m, \\ \frac{1}{|A|^m} \prod_{\vartheta \in A^m} \frac{\prod_{a \in A} (\Gamma(\nu_x(\vartheta a) + 1/2)/\Gamma(1/2))}{(\Gamma(\bar{\nu}_x(\vartheta) + |A|/2)/\Gamma(|A|/2))}, t > m; \end{cases} \quad (3)$$

where $\nu_x(\vartheta)$ is the number of the sequences ϑ occurs in the word x ; $\bar{\nu}_x(\vartheta) = \sum_{a \in A} \nu_x(\vartheta a)$; $x = x_1 \dots x_t$, and $\Gamma()$ is a gamma function. The parameter m is called depth of analysis and in fact, it determines the depth (quantity of elements) in analysis of the considered process.

Let us define also distribution of probabilities $\{\omega = \omega_1, \omega_2, \dots\}$ for integer $\{1, 2, \dots\}$ as follows:

$$\omega_i = 1/\log(i+1) - 1/\log(i+2) \quad (4)$$

Let us use further this distribution.

The measure R is defined as follows:

$$R(x_1 \dots x_t) = \sum_{i=0}^{\infty} \omega_{i+1} K_i(x_1 \dots x_t) \quad (5)$$

The item ω_i plays in this case a role of weight factors which with growth of an index i (i.e. an depth of a Krichevsky measure) become ever less. It is clear that too big orders in Krichevsky's measure should have smaller weight and less influence to a forecast. As a result distribution (4) has been chosen as weight factors. In general, weight factors represent the varied parameter of a method and can change, depending on a concrete time series and a method.

Thus, I have constructed a measure R which is an estimation of probabilities for a class of all stationary and ergodic sources on the finite alphabet and can directly be applied to forecasting.

So, the values of a measure R based on a formula (5) can serve as an estimations of probabilities for a stationary and ergodic process and can be used for the solution of a forecasting problem.

6. Prediction scheme for the sources that generate values from a finite alphabet based on a universal measure

Let us consider the forecasting scheme based on of a universal measure for sources on a finite discrete alphabet.

For implementation of a forecast, I will use R measure (5). It is clear, what to count the infinite sum (5) in real algorithms it will not be possible, therefore all calculations I will do according to that each following item in the sum (5) makes the lesser contribution to total value R . It is possible to prove this fact without effort how theoretically, and to check an empirical way. Owing to this property for calculation the R -measure I will use the first m elements of the sum and number m I will call the depth of the analysis of a method.

Firstly, I will consider a source generating values from the finite alphabet. In this case, the scheme of calculation of a measure R is simple. Let $x_1 \dots x_t$ is available temporary time series. For everyone $a \in A$ I will construct sequence $x_1 \dots x_t a$ and I will calculate the conditional probability based on R measure:

$$R(a|x_1 \dots x_t) = R(x_1 \dots x_t a) / R(x_1 \dots x_t) \quad (6)$$

Received by this way for everyone $a \in A$ values, it is possible to use as estimates of the corresponding unknown probabilities $P(x_1 \dots x_t a)$. Element a , which have the maximum estimation of probability, is a predicted value.

7. Prediction method for the sources that generate values from a continuous interval based on a universal measure

Here I consider the problem of forecasting for the source that generates values from a continuous interval (when alphabet is infinite). The application of this forecasting type is quite known and much more useful, than case of finite alphabet. It was previously shown only the theoretical results of prediction of such time series. In particular, these results are described in Ref. [13]. There are description of the implementation of the R -method to this kind of forecasting and that there are some interesting practical and experimental results.

Let us consider now the forecasting scheme for a source generating values from a continuous interval. Let there be the stochastic process generating sequence X_t and

each element is from the standard Borel space Ω , which represents a continuous interval $[A, B]$. And let us also $\{\Pi_n\}, n \geq 1$ as an increasing sequence of finite partitions of an interval $[A, B]$ with n parts (we will call this process by a quantization). In our case, partition of intervals was made evenly on equal subintervals; that is, the size of each subinterval is defined as follows: $h = \frac{B-A}{n}$. Justification of a this fact will be given further. Let us define also $x^{[k]}$, as an element of Π_k containing a point x .

Let us define joint distribution P_n for $(X_1, X_2, \dots, X_n)$, as a function of density of probability $p(x_1 x_2 \dots x_n)$ in relation to a sigma-finite measure L . As a measure L I can use the measure of Lebesgue or any other countable measure.

For the integer s and n I will define an estimation of probability's density $p(x_1 x_2 \dots x_n)$:

$$p^s(x_1, \dots, x_n) = P(x_1^{[s]}, \dots, x_n^{[s]}) / L(x_1^{[s]} \dots x_n^{[s]}) \quad (7)$$

Let us define now an estimation of probability's density r as follows:

$$r(x_1 \dots x_t) = \sum_{s=1}^{\infty} \omega_s R(x_1^{[s]} \dots x_t^{[s]}) / L(x_1^{[s]} \dots x_t^{[s]}) \quad (8)$$

Factors ω_s are defined by the scheme (4) and bear a role of weight factors for a case of each Partition from Π_k . Thus, there is a normalization of everyone item (representing some estimation of probability of sequence at the set partition) on a sigma-finite measure L which represents some estimation of probabilities of subintervals' sizes. Thus, I connect among themselves estimates of probability's density for a case of various increasing partitions. It relieves us of dependence of forecast's results on a concrete partition. This process of connection of estimates of probabilities' density with a normalization on L and with weight factors ω_s is called as a "splice." As a result, I can use for prediction any sequences of finite partitions (also any).

As shown in Refs. [9, 14], the value $r(x_1 \dots x_t)$ is an estimation of unknown probability density $p(x_1 \dots x_t)$, and the corresponding conditional probability

$$r(a|x_1 \dots x_t) = r(x_1 \dots x_t a) / r(x_1 \dots x_t) \quad (9)$$

is a suitable estimation of probability $p(a|x_1 \dots x_t)$. It is obvious that the future precision of forecast strongly depends on the process of quantization and the value n (quantity of letters that consist of alphabet). In fact, that size of an alphabet has an influence on the error of prediction. The items' quantity in the sum (8) at realization of the described algorithm, as in a discrete case, is equal to a depth of the analysis m .

8. Method of the alphabet's grouping

Specifics of work of forecasting algorithms' based on R-measure is that for getting enough precise results of a forecast and the accounting of big lengths of sequences and alphabets I are often needed in great lengths' of time series' and the size of their alphabets that does prediction work very complex. Complexity of methods exponential depends on a depth of the analysis and at least as a square from the size of an alphabet. As a result, the universal method, allowing to reduce operating time generally any algorithm of forecasting, has been developed. This method has been offered

for the first time by B. Ya. Ryabko in Ref. [21]. Here, the method is generalized for the forecasting algorithm. Let us describe this algorithm under the name “a method of alphabet’s grouping”.

As it was shown above, the alphabet’s size has an influence on the error of prediction (larger alphabet gets the better precision) and therefore I try to use the larger alphabets, but limited by a hardware. For resolving this problem, I use the partition of the large alphabet into the some small parts (classes) of large alphabet. Each part includes some quantity letters of original alphabet. Then, I predict not the only one letter, but the batch of original letters and then apply our prediction method for the letters in this batch. It is important to note that if I use the small original alphabet, then at the second step we’ll get the small quantity of letters and it’s not too good for the prediction. That’s why I will use rather large alphabets for applying this method.

Let us have the alphabet A , which consists of elements of a temporary number of some big length N . Then I choose some recurrent partitions of the alphabet A according to the following algorithm:

- Let us choose a set of disjoint subsets of a set A : $B_1, B_2, \dots, B_{N_1}$ where $N_1 \ll N$ and everyone B_i contains one or several elements from A .
- The received elements of the alphabet A , contained in each B_i , split for N_2 parts of B_{i,N_2} , which also contain the corresponding elements of a set A .
- We continue this recurrent process while not receiving the set number of partitions.

Then, I continue process on the following algorithm:

- We write down an initial time series by the terms of the alphabet B (i.e., for all elements of an initial time series it will be transformed to the corresponding symbols from the alphabet B) and then I predict an element from alphabet B . Let us denote it, as B_i .
- We filter an initial time series on predicted value B_i : I leave in it only those elements from A which belong to the predicted B_i .
- Further, I write down a time series by the terms of alphabet B_i (i.e., the second level alphabet).
- We predict an element of the alphabet of the second level, and then, I continue by this way procedure to the last level of partition.

Let us give an example of algorithm’s work. Let the alphabet is given $A = \{i\}$, where $i = \overline{1, \dots, 12}$. And let a time series is given: $X(A) : 1, 3, 5, 5, 6, 7, 8, 1, 3, 5, 5, 6, 7, 8, 1, 3, 5$. It is required to predict the following element (which, obviously, it should be equal 5). Let us separate the initial alphabet on four equal parts, having determined by that the new alphabet $B: \{B_i\}, i = \overline{1, \dots, 4}$. Let us make partition uniform. As a result, I will receive the following values $B_i: B_1 = \{1, 2, 3\}; B_2 = \{4, 5, 6\}; B_3 = \{7, 8, 9\}; B_4 = \{10, 11, 12\}$.

Each of parts B_i contains three elements from a set A which would be included in the secondary alphabet B_{ij} . As a result, to everyone A_i there will unequivocally correspond to an element B_{ij} .

Now, I will rewrite an initial time series by the terms of B_i : $X(B)$: 1, 1, 2, 2, 2, 3, 3, 1, 1, 2, 2, 2, 3, 3, 1, 1, 2. Now, I will apply any method of forecasting to the received time series and I will find the predicted value in terms of the alphabet B . In this case, predicted value, obviously, will be 2.

Further I realize a filtration of initial sequence by a principle: If element $X_i(A)$ is in a set B_2 , I leave it among, differently I delete it. Let us receive the following sequence: 5, 5, 6, 5, 5, 6, 5. There are only 2 symbols of the alphabet from 12 (since the capacity of a set B_i is equal 3, and number 4 in initial sequence does not meet never). Therefore, it is possible to copy the set time series in terms of the new alphabet from three elements (four passes in 1, 5 – in 2, 6 – in 3): 2, 2, 3, 2, 2, 3, 2. Further, it is simply defined predicted value in the received time series and it lead to in terms of the initial alphabet. It is obvious that the predicted value is equal to 2 which corresponds to number 5 in the initial alphabet. The symbol of the initial alphabet 5 also will grow out of forecasting.

In general, the number of partitions of the alphabet is not limited; however for use of this algorithm in our concrete method of forecasting, I will take one partition of the initial alphabet. As the given experiments and results show further, it is quite enough for essential decrease the complexity of the algorithm without loss a precision of the methods.

9. Experimental results of forecasting

9.1 Forecasting of currency exchange rates

The described methods based on an R-measure have been realized on a supercomputer and tested on forecasts of a real data. Results of these forecasts are given in tables written below.

All research studies were carried out in two modes. The first mode—on-line—means forecasting of values of a temporary time series for 1 step forward. The second mode—on 10 steps forward—forecasting of values of a time series for 10 steps forward. Thus, forecasting for 10 steps forward was considered as follows: Thus, firstly we predict the next value of time series, then we add this predicted value to time series as last element, then we predict the next (already second) value by using supplemented time series, further forecasting was conducted on one step, but with use of already grown stout time series and so on I continue to the 10th predicted element then I consider an error. Forecasting was carried out on 10 different time series with the subsequent averaging of an error.

Firstly, I realized forecasting by a natural way, which means forecasting absolute values of the time series. Then I realize preliminary transformation of the time series as follows. I replaced each value of the sequence on the difference between neighboring values: $x'_t \rightarrow (x_{t+1} - x_t)$. Then I predict the x'_t value and find the x_{t+1} as follows: $x_{t+1} = x'_t + x_t$.

As the experimentally research shows, the second method (using transformed sequence) gives much better precision of a forecasting. Besides, such approach allows to reduce the necessary size of a continuous interval which consists of the predicted values; and also allows to reveal linear and quasilinear trends and the periods on them. It was impossible when forecasting the absolute sizes of a temporary sequence.

Length of time series L	Partition n	Depth of the analysis m	R-measure On-line	R-measure 10 steps
500	10	2	0.0084	0.0299
		5	0.0084	0.0299
	20	2	0.0083	0.0159
		5	0.0083	0.0159
	50	2	0.0083	0.0187

Table 1.
Exchange rate of euro/dollar, D1.

Delimitation of an interval was carried out naturally: the size of maximum and minimum (taking into account a sign) deviations of the current and previous elements was considered; the received values also undertook as left and the right border of an interval further broke into the quantity of parts equal to the capacity of the alphabet.

Parameters are specified in each of the following below tables: the amount of time series, quantity of parts of partition of a continuous interval, parameter m , and the error of the method.

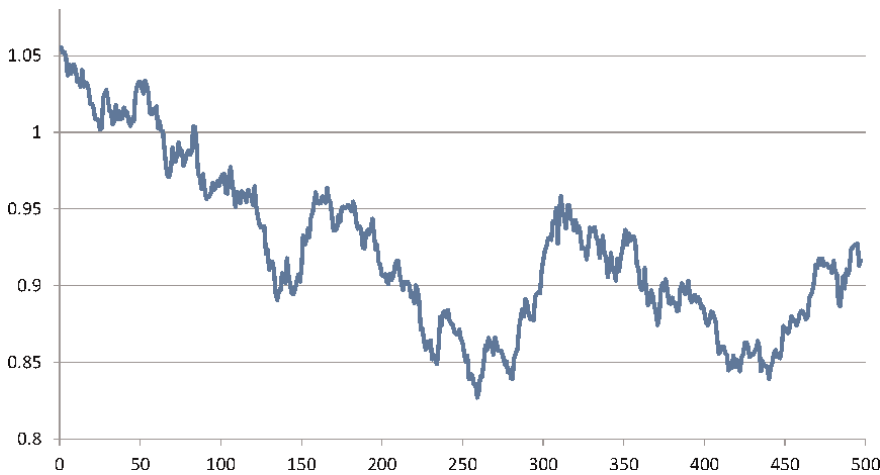


Figure 1.
Schedule of a rate of euro/dollar (period: 1 day).

Length of time series L	Partition n	Depth of the analysis m	R-measure On-line	R-measure 10 steps
500	10	2	0.00114	0.00131
		5	0.00114	0.00131
	20	2	0.00103	0.00131
		5		0.00131
	50	2	0.00104	0.00141

Table 2.
Exchange rate of euro/dollar, H1.

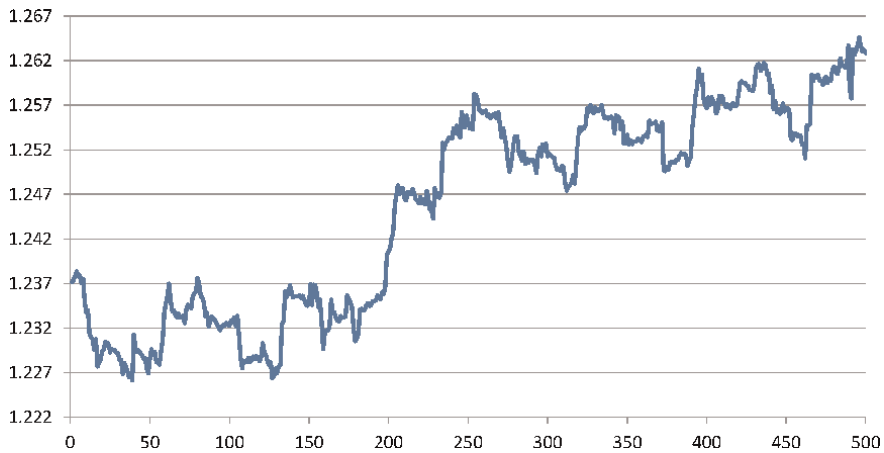


Figure 2.
Schedule of a rate of euro/dollar (period: 1 hour).

In **Table 1** data of a forecast of exchange rate of euro/dollar with a time interval (timeframe) 1 day (D1) are provided in the period with 6/20/2012. The schedule of a time series is provided in **Figure 1**.

In **Table 2**, results of forecasting of the same rate of euro/dollar, but already on a time series with the period (timeframe) of 1 hour, are given. The schedule of a time series is provided in **Figure 2**.

Apparently from above provided tables, there is a certain limit of the size of the alphabet (in our case there are partitions of a continuous interval) after which precision of methods does not grow. Also, from the received data it is visible that depth of the analysis after value $m = 2$ improves precision of a forecast rather insignificantly and after some value m the precision, as well as in a case with the size of the alphabet, does not change any more. Actually, it means that for getting optimum forecasts for acceptable time it is enough to pick up such minimum values of the alphabet's size and analysis' depth of both methods which will give optimum (approached to limit of precision) values of errors.

The described limits of precision of methods exist, because in case of forecasting of difficult time series in which there are no visible or rather simple regularities, both methods do not find them and simply average value of a trend (a difference between the next elements) and use it as a forecast. In the presence of any regularity among, algorithms will need a deep water of the analysis (when m is less, than the length of the period regularity, described algorithm simply will not reveal it).

9.2 Forecasting of inflation indexes

Let us consider the cases described in part 8.1 with application of a method of alphabet's group. The received results are given in **Tables 3** and **4**. Thus, Partition of the alphabet A was carried out on equipotent disjoint subsets. The sizes of subsets are specified in a column of tables with Partition size: there are two number in brackets, first of which is the number of Partition (quantity of subsets) of a set A , and the second, respectively, is the size of each of this subsets.

Apparently from given aforesaid data, the modification of a method of alphabet's grouping does not decrease precision, but speed of the algorithm's work essentially

Length of time series <i>L</i>	Partition <i>n</i>	Depth of the analysis <i>m</i>	R-measure On-line	R-measure 10 steps
500	10 (10*1)	2	0.0084	0.0299
		5	0.0084	0.0299
	20 (20*1)	2	0.0080	0.0159
		5	0.0080	0.0159
	20 (10*2)	2	0.0100	0.0345
	20 (5*4)	2	0.0080	0.0159
	50 (10*5)	2	0.0099	0.0173
		2	0.0080	0.0119
	50 (25*2)	2	0.0083	0.0107
	50 (50*1)	2	0.0083	0.0187

Table 3.
Exchange rate of euro/dollar, D1.

increases. The error of the method depends not only on the size of the initial alphabet, but also on the chosen depth of the analysis. Thus, forecasting for 10 steps forward gives an error, comparable as with a margin error of on-line method that speaks about good applicability of the R-method when forecasting for a set of steps forward.

Let us consider results of forecasting of various economic indicators, in particular, indexes of the consumer and producer prices. In the following two tables, data of a forecast of the general total consumer price index on the territory of the USA since 01.1990 up to 02.2013 with a time interval is 1 month. Time series values were predicted since 02.2012 up to 02.2013. Graphs of these time series are provided on **Figures 3 and 4**, respectively.

Length of time series <i>L</i>	Partition <i>n</i>	Depth of the analysis <i>m</i>	R-measure On-line	R-measure 10 steps
500	10 (10*1)	2	0.00114	0.00131
		5	0.00114	0.00131
	20 (20*1)	2	0.00103	0.00131
		5		0.00131
	20 (10*2)	2	0.00103	0.00185
	20 (5*4)	2	0.00103	0.00185
	50 (10*5)	2	0.00104	0.00141
		2		0.00141
	50 (25*2)	2	0.00104	0.00141
	50 (50*1)	2	0.00104	0.00141

Table 4.
Exchange rate of euro/dollar, H1.

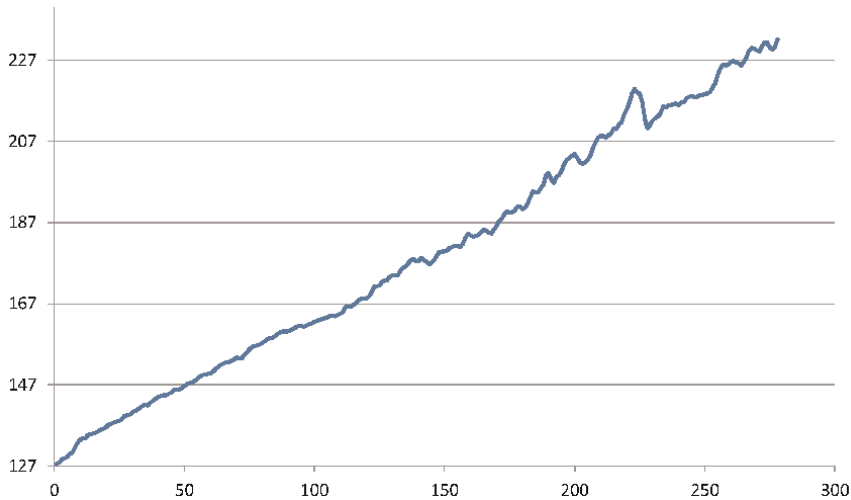


Figure 3.
Consumer price index.

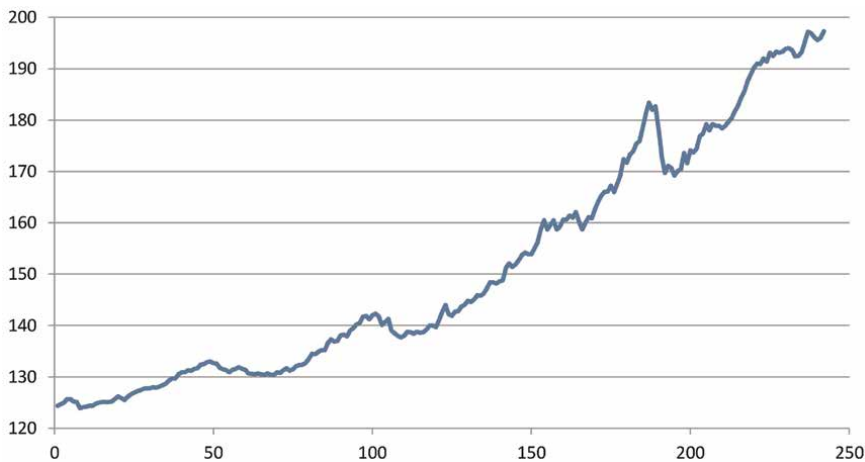


Figure 4.
Producer price index.

In **Table 5**, the results of a forecast of the last part of a sequence, in which dispersion of time series obviously was more than in the middle (i.e., well visible from **Figure 3**), are reflected. In **Table 6** the results of a forecast of the central part of the schedule which looks more smooth are shown. Apparently from tables, the results of a forecast of a time series with lower dispersion (**Table 6**) it is notable better (despite the smaller amount of time series) that is quite natural.

Apparently by the above-stated results, in the second table, reflecting values on smaller time series and with smaller dispersion, the forecast error is much less, than in the first table corresponding to a greater length of time series by a wide spacing of values.

Length of time series L	Partition <i>n</i>	Depth of the analysis <i>m</i>	R-measure On-line	R-measure 10 steps
277	5 (5*1)	2	0.112769	0.257077
	10 (10*1)	2	0.092615	0.088154
		5		
	20 (20*1)	2	0.107846	0.136
		5		
	20 (10*2)	2	0.097692	0.136
	20 (5*4)	2	0.107846	0.136
	100 (10*10)	2	0.100769	0.230308
		5		
	100 (20*5)	2	0.111846	0.230308
	100 (5*20)	5	0.111846	0.230308

Table 5.
Consumer price index. Period: 03.2012–2102.2013.

Length of time series L	Partition <i>n</i>	Depth of the analysis <i>m</i>	R-measure On-line	R-measure 10 steps
240	5 (5*1)	2	0.056	0.105231
	10 (10*1)	2	0.050462	0.193077
		5		
	20 (20*1)	2	0.047692	0.065231
		5		
	20 (10*2)	2	0.047692	0.065231
	20 (5*4)	2	0.047692	0.095385
	100 (10*10)	2	0.047692	0.11
		5		
	100 (20*5)	2	0.047692	0.11
	100 (5*20)	5	0.047692	0.11
	240 (24*10)	5	0.045692	0.131231

Table 6.
Consumer price index. Period: 03.2002–2002.2003.

Further, I will consider forecasting of a producer price index. Results are given in **Table 7**.

Given results confirm the conclusions drawn earlier. It is visible that the method of alphabet's grouping essentially increases the speed of algorithm's work for the fixed partition, without worsening the results of a forecast.

Length of time series L	Partition <i>n</i>	Depth of the analysis <i>m</i>	R-measure On-line	R-measure 10 steps
277	5 (5*1)	2	0.11093	0.217442
	10 (10*1)	2	0.105814	0.158488
		5		
	20 (20*1)	2	0.10593	0.125116
		5		
	20 (10*2)	2	0.10593	0.125116
	20 (5*4)	2	0.10593	0.125116
	100 (10*10)	2	0.105814	0.117791
		5		
	100 (20*5)	2	0.11186	0.118721
	100 (5*20)	5	0.105814	0.117791
	240 (24*10)	5	0.105814	0.11814

Table 7.
Producer price index.

10. Conclusion

Apparently from given aforesaid results, a method based on a universal measure shows rather a high level of precision of the forecasting on a real data. It is true for comparison to other methods of forecasting that is in detail shown in Ref. [9]. Besides, more effective realization of the method R, allowing to reduce complexity of search of various options, has been created. Complexity has decreased to constant concerning length of the alphabet (earlier it was linear). Also, I can remove restrictions on hardware resources and parameters by using a method of alphabet's grouping. It has allowed to reach the same complexity than earlier was reached only on a supercomputer.

Besides, it is important to note that good results of forecasting of time series that generate values from a continuous interval were obtained by developing the implementation of R-measure to the case of continuous interval time series. There were a unique experimental results.

Advantage of the offered method in complex with a method of alphabet's grouping is that these methods of forecasting can be generalized easily on a case of multidimensional time series where each element has many attributes (not only the one value) properties correlating with this time series.

Author details

Alexander Lysyak
Novosibirsk State University, Novosibirsk, Russia

*Address all correspondence to: accemt@gmail.com

IntechOpen

© 2025 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Ahmed N. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*. 2010;**29**(5–6):594–621
- [2] Bontempi G. Local Learning Techniques for Modeling, Prediction and Control. Ph.d. Belgium: IRIDIA-Universit de Libre de Bruxelles; 1999
- [3] Poskitt DS, Tremayne AR. The selection and use of linear and bilinear time series models. *International Journal of Forecasting*. 1986;**2**(1):101–114
- [4] Tong H. Non-linear Time Series: A Dynamical System Approach. Oxford University Press; 1990
- [5] Tong H. Threshold Models in Nonlinear Time Series Analysis. Berlin: Springer Verlag; 1983
- [6] Tong H, Lim KS. Threshold autoregression, limit cycles and cyclical data. *Journal of the Royal Statistical. Series B (Methodological)*. 1980;**42**(3): 245–292
- [7] Engle R. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom. *Econometrica*. 1982;**50**(4):987–1007
- [8] Clements MP. Forecasting economic and financial time-series with non-linear models. *International Journal of Forecasting*. 2004;**20**(2):169–183
- [9] Pristavka P, Ryabko B. Experimental investigation of the efficiency of universal measures based on universal coding. *SibSUTIS Vestnik*. 2010;**4**:26–35
- [10] Cheng H. Multistep-ahead time series prediction. *Lecture Notes in Computer Science*. 2006;**3918**: 765–774
- [11] Zhang G, Patuwo BE, Michael YH. Forecasting with artificial neural networks. The state of the art. *International Journal of Forecasting*. 1998;**14**(1):35–62
- [12] Palit AK, Popovic D. Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications (Advances in Industrial Control). New York: Secaucus, NJ, USA: Springer-Verlag; 2005
- [13] Ryabko B. Compression-based methods for nonparametric prediction and estimation of some characteristics of time series. *IEEE Transactions on Information Theory*. 2009;**55**(9):4309–4315
- [14] Ryabko BY. Prediction of random sequences and universal coding. *Problems of Information Transmission*. 1988;**24**(2):87–96
- [15] Ryabko B, Monarev V. Experimental investigation of forecasting methods based on data compression algorithms. *Problems of Information Transmission*. 2005;**41**(1):65–69
- [16] Nevill-Manning CG, Witten IH, Paynter GW. Lexically-generated subject hierarchies for browsing large collections. *International Journal of Digital Libraries*. 1999;**2**(3):111–123
- [17] Orlitsky A, Santhanam NP, Zhang J. Always good Turing: Asymptotically optimal probability estimation. *Science*. 2003;**302**(5644):427–431
- [18] Ryabko BY. Application of universal source coding to statistical analysis of time series, chapter 1, world. *Scientific Review*. 2008;**55**(1):5–7
- [19] Nevill-Manning CG, Witten IH. Identifying hierarchical structure in

sequences: A linear-time algorithm.
Journal of Artificial Intelligence
Research. 1997;7:67-82

[20] Krichevsky R. Universal
Compression and Retrival. Kluwer
Academic Publishers; 1993

[21] Ryabko B, Astola J, Gammernan A.
Adaptive coding and prediction of
sources with large and infinite alphabets.
IEEE Transactions of Information
Theory. 2008;54(8):3808-3813

*Edited by Jorge Rocha, Cláudia M. Viana,
Inês Boavida-Portugal and Eduardo Gomes*

Time Series Analysis - Frontiers in Research and Practice offers a concise overview of cutting-edge methods for understanding, forecasting, and applying models to temporal data. The volume combines Markov Chain Monte Carlo, Bayesian ARIMA, and Posterior Predictive Distribution to enable uncertainty-aware time series forecasting. It reinterprets temporal data through co-evolving time series, causal dependencies, shift/visibility graph representations, graph representation learning, multiplex networks, and temporal metagraphs, tools suited for dependency regime forecasting and critical fields such as cybersecurity. Bridging theory and application, it advances data-driven prediction using Takens' embedding theorem, dynamic models, neural network algorithms, and demonstrates deployment-ready, efficient pipelines via neuro-symbolic techniques for edge computing, integrating spectral signal processing, pattern recognition, signal symbolization, and convolutional neural networks. Additionally, it features compression-based prediction with universal coding and decision trees, providing transparent and robust baselines. The book's strengths include a unified cross-disciplinary view, focus on interpretability and uncertainty, guidance for real-time, resource-limited environments, and practical examples (such as numerical and dislocation modeling in geoscience) that turn techniques into actionable insights. Geared toward researchers, practitioners, and postgraduate students, this edited volume provides a modern, interoperable toolkit linking rigorous modeling with real-world applications and decision-making.

Published in London, UK

© 2025 IntechOpen
© Gitanna / iStock

IntechOpen

