Bachelor Thesis

A PLATFORM FOR MULTI-ENTITY TIME SERIES ANALYSIS WITH AI-ASSISTED INTERPRETATION

Panayiota Bekri

University of Cyprus
Department of Computer Science

Department of Computer Science

May 2026

UNIVERSITY OF CYPRUS

Faculty of Pure and Applied Sciences

Department of Computer Science

A PLATFORM FOR MULTI-ENTITY TIME SERIES ANALYSIS WITH AI-ASSISTED INTERPRETATION

Panayiota Bekri

Advisor:

Dr. Panayiotis Kolios

The Thesis was submitted in partial fulfillment of the requirements for obtaining the Computer Science degree of the Department of Computer Science of the University of Cyprus

May 2026

ABSTRACT

This thesis presents the design and implementation of a web-based platform for exploratory time series analysis across multiple entities, with potential applications in domains such as bioinformatics, healthcare, and environmental data analysis. The system supports the ingestion, preprocessing, and analysis of heterogeneous datasets provided in CSV format.

The platform enables users to upload multiple datasets, align observations by entity and year, normalize values, assess stationarity, perform decomposition, and analyze relationships through correlation and lag-based cross-correlation techniques. In addition, the system provides exportable summaries and supports AI-assisted interpretation of analytical results.

The system follows a modular architecture consisting of a Django-based web application and a separate analytical core. The main contribution of this work is the development of an integrated environment that combines statistical time series processing with a user-friendly interface for comparative analysis.

The platform is intended as an exploratory analytical tool rather than a causal inference system. It supports hypothesis generation and pattern discovery while highlighting potential limitations and risks such as spurious correlations.

Keywords: time series analysis, multi-entity data, web-based platform, cross-correlation, AI-assisted interpretation

TABLE OF CONTENTS

LIST OF FIGURES

List of Abbreviations

ADF	Augmented Dickey-Fuller Test
FFT	Fast Fourier Transform
STL	Seasonal-Trend Decomposition using Loess
CSV	Comma-Separated Values
API	Application Programming Interface
AI	Artificial Intelligence
JSON	JavaScript Object Notation
ML	Machine Learning
CCF	Cross-Correlation Function
OLS	Ordinary Least Squares
UI	User Interface
API	Application Programming Interface
DF	DataFrame (Pandas)
OWID	Our World in Data
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript

1 Introduction

1.1 Background and Motivation

The increasing availability of large-scale data across domains such as healthcare, bioinformatics, and environmental science has created significant opportunities for data-driven analysis. Time series data, in particular, play a crucial role in capturing temporal dynamics and understanding complex real-world phenomena.

However, despite the availability of advanced statistical and computational techniques, the practical application of time series analysis remains challenging. Analysts are often required to combine multiple tools and manually perform preprocessing, alignment, and interpretation tasks, which introduces inefficiencies and increases the risk of errors.

In recent years, advances in artificial intelligence have opened new possibilities for assisting users in interpreting complex analytical results. Nevertheless, the integration of statistical analysis with AI-assisted interpretation in a unified platform remains limited.

Motivated by these challenges, there is a growing need for integrated platforms that simplify the process of multi-entity time series analysis while supporting modern AI-assisted interpretation. Such platforms are particularly relevant in biomedical and bioinformatics applications, where datasets are often heterogeneous and complex.

The developed system has been deployed online and is publicly accessible for demonstration purposes at the following URL:

`https://time-series-analyzer-wufp.onrender.com`

1.2 Problem Statement

Despite the existence of numerous statistical tools and analytical frameworks, current workflows for time series analysis remain fragmented and difficult to use, particularly in multi-entity settings.

Users are typically required to perform data preprocessing, alignment, and analysis using separate tools, resulting in inefficient workflows and increased complexity. Furthermore, existing systems often focus primarily on computation, offering limited support for interpreting analytical results, which reduces their effectiveness in exploratory research contexts.

This lack of integration and interpretability highlights the need for a unified platform that combines data processing, statistical analysis, and AI-assisted interpretation.

1.3 Aim and Objectives

The aim of this thesis is to design and implement an integrated web-based platform for multi-entity time series analysis that combines statistical processing with AI-assisted interpretation.

The system is intended to support exploratory data analysis by simplifying complex analytical workflows and enhancing result interpretability.

The objectives of this work are:

- To support the ingestion and validation of heterogeneous time series datasets
- To preprocess and align data across entities and time periods
- To apply statistical techniques including normalization, stationarity testing, decomposition, and correlation analysis
- To detect relationships using lag-based cross-correlation methods
- To provide AI-assisted interpretation of analytical results

1.4 Research Questions

This thesis addresses the following research questions:

- How can heterogeneous time series datasets be efficiently integrated and analyzed within a unified platform?
- Which statistical and computational intelligence techniques are most suitable for exploratory multi-entity time series analysis?
- How can AI-assisted interpretation improve the usability and understanding of analytical results?
- How can such a system be applied to real-world domains such as bioinformatics and healthcare?

1.5 Contribution

The main contribution of this thesis is the development of a unified platform that integrates data ingestion, preprocessing, statistical analysis, and AI-assisted interpretation within a single workflow.

The system enables comparative analysis across multiple entities, supports lag-based relationship detection, and provides human-readable explanations of analytical results. By combining statistical methods with AI-based interpretation, the proposed platform improves both usability and accessibility for non-expert users.

Additionally, the system is evaluated using real-world datasets through case studies in healthcare and socioeconomic domains, demonstrating its effectiveness as an exploratory analytical tool.

1.6 Structure of the Thesis

This chapter introduced the motivation, problem statement, objectives, and contributions of the thesis, highlighting its relevance to computational intelligence applications in data analysis and bioinformatics. The remainder of this thesis is organized as follows. Chapter 2 presents the theoretical background and related work. Chapter 3 describes the research methodology. Chapter 4 presents the system architecture and design. Chapter 5 details the implementation of the platform. Chapter 6 presents the experimental evaluation and results. Finally, Chapter 7 concludes the thesis and outlines directions for future work.

2 Background and Related Work

2.1 Introduction

Time series analysis is a fundamental tool for understanding temporal patterns in data across multiple domains, including economics, healthcare, bioinformatics, and environmental science. It enables the identification of trends, seasonal effects, and relationships between variables over time.

In multi-entity settings, time series analysis becomes more complex due to data heterogeneity, missing values, and differing time spans, which require careful preprocessing and alignment.

This chapter presents the theoretical background required for this thesis, including time series concepts, statistical analysis methods, and computational techniques. It also reviews related work in multi-entity time series analysis and data-driven platforms.

2.2 Time Series Analysis

A time series is a sequence of observations recorded over time and can be represented as a sequence x_t, where t denotes time. Time series analysis aims to extract meaningful patterns such as trends, seasonality, and relationships between variables.

In multi-entity settings, time series are analyzed across different entities, such as countries, regions, or biological samples. These datasets are often heterogeneous, containing missing values and varying time spans, which makes preprocessing and alignment essential steps in the analysis process.

2.3 Data Preprocessing and Alignment

Before applying statistical analysis, time series data must be preprocessed to ensure consistency and comparability. This includes handling missing values, converting data types, aggregating duplicate entries, and aligning datasets across common time indices.

In multi-entity analysis, datasets are typically aligned using join operations (e.g., outer joins) based on common keys such as entity and time. This allows pairwise comparisons while preserving available data.

2.4 Normalization

Normalization is used to transform data into a common scale. One widely used method is z-score normalization:

$$z = \frac{x - \mu}{\sigma}$$

where μ is the mean and σ is the standard deviation.

This method assumes approximately normally distributed data and is widely used in statistical analysis to ensure comparability across variables.

This transformation ensures that different time series can be compared regardless of their original scale.

2.5 Stationarity and Augmented Dickey-Fuller Test

Stationarity is a key concept in time series analysis. A stationary time series has constant statistical properties over time, such as mean and variance. Many statistical methods assume stationarity; therefore, it is important to test whether a series satisfies this property.

The Augmented Dickey-Fuller (ADF) test is commonly used for detecting stationarity [1, 2]. If a time series is found to be non-stationary, differencing can be applied to transform it into a stationary form.

2.6 Time Series Decomposition

Time series decomposition separates a series into three components: trend, seasonal, and residual. This allows better understanding of the underlying structure of the data.

A widely used method is Seasonal-Trend Decomposition using Loess (STL) [3]. Decomposition improves interpretability by isolating long-term patterns from short-term fluctuations.

2.7 Frequency Analysis

Frequency analysis is used to identify periodic patterns in time series data. The Fast Fourier Transform (FFT) is commonly used to transform a time series from the time domain to the frequency domain [4, 5].

This transformation allows the detection of dominant cycles and recurring patterns in the data.

2.8 Correlation and Cross-Correlation

Correlation measures the strength and direction of the relationship between two variables. The Pearson correlation coefficient is commonly used to quantify linear relationships [6].

In time series analysis, cross-correlation extends this concept by considering lagged relationships [7]. This allows the identification of delayed effects between variables, which is particularly important in domains such as epidemiology, economics, and bioinformatics.

2.9 Computational Intelligence and AI-Assisted Analysis

Computational intelligence techniques, including machine learning and AI-based systems, have significantly enhanced data analysis workflows. These approaches allow the extraction of patterns from complex datasets and support decision-making processes.

AI-assisted interpretation can help users understand analytical results by generating explanations, summarizing findings, and highlighting important relationships. This is especially useful in exploratory analysis, where users may not have extensive statistical expertise.

2.10 Related Work

Time series analysis has been extensively studied across multiple domains, including economics, healthcare, and environmental science. Traditional approaches focus on statistical techniques such as decomposition, stationarity analysis, and correlation methods for identifying relationships between variables.

In recent years, several platforms and tools have been developed to support time series analysis and data exploration. Examples of widely used tools include Tableau, Power BI, and Python-based analytical environments such as Pandas and Jupyter notebooks. These tools provide strong visualization and analysis capabilities but typically require manual integration of preprocessing, analysis, and interpretation steps.

However, they often require significant user expertise and do not provide a unified workflow for multi-entity time series analysis.

Furthermore, recent research in computational intelligence has explored the use of artificial intelligence techniques to assist in data analysis. AI-based systems can support interpretation by generating natural language explanations and highlighting patterns within data.

Despite these advancements, there remains a gap in the development of unified platforms that combine statistical time series analysis with AI-assisted interpretation. The system proposed in this thesis aims to address this gap by integrating these components into a single workflow.

These limitations highlight the need for integrated systems that combine statistical analysis with AI-assisted interpretation, particularly in multi-entity time series contexts, motivating the development of the platform proposed in this thesis.

2.11 Summary

This chapter presented the theoretical background necessary for understanding the methods used in this thesis, including time series analysis, preprocessing, normalization, stationarity, decomposition, and correlation.

These concepts provide the foundation for the system design and analytical methods developed in the following chapters.

3 Research Methodology

3.1 Introduction

This chapter describes the methodology followed in this thesis for the design, development, and evaluation of the proposed system. The study adopts a design-oriented approach, focusing on the implementation of an integrated platform for multi-entity time series analysis and its evaluation using real-world datasets. The analytical methods used in this system are based on established time series analysis techniques presented in academic literature and university-level course material [2].

3.2 Research Approach

The research follows a system design and experimental evaluation methodology. The main objective is to develop a functional platform that integrates data preprocessing, statistical analysis, and AI-assisted interpretation within a unified workflow. Modern data-driven systems increasingly integrate data processing, modeling, and deployment pipelines [8].

The approach consists of the following stages:

- System design and architecture definition
- Implementation of analytical components
- Integration of AI-assisted interpretation
- Experimental evaluation using real-world datasets

This approach is suitable for exploratory data analysis, where the objective is to identify patterns and relationships rather than to establish causal inference. This distinction is particularly important in time series analysis, where observed relationships may be influenced by underlying trends, external factors, or data limitations.

3.3 Data Sources

The experimental evaluation is based on publicly available datasets from sources such as Our World in Data [9]. These datasets include indicators related to healthcare, socioeconomic factors, and population statistics.

The data used in this study are characterized by:

- Multiple entities (e.g., countries)
- Temporal observations across multiple years
- Heterogeneous formats and varying data quality

These characteristics make them suitable for evaluating the capabilities of the proposed platform.

3.4 Data Processing Workflow

The system follows a structured workflow for processing and analyzing time series data:

1. Data ingestion and validation
2. Data cleaning and preprocessing
3. Alignment of datasets across entities and time
4. Normalization of values
5. Statistical analysis (stationarity, decomposition, correlation)
6. Lag-based cross-correlation analysis
7. Generation of results and AI-assisted interpretation

Each step is implemented as a modular component, allowing flexibility and extensibility.

3.5 Analytical Methods

The system employs a set of statistical techniques for time series analysis:

- Z-score normalization for data scaling
- Augmented Dickey-Fuller test for stationarity analysis
- Time series decomposition (STL) for trend and seasonal analysis
- Fast Fourier Transform (FFT) for frequency analysis
- Pearson correlation for measuring linear relationships
- Cross-correlation for detecting lag-based relationships

These methods are selected due to their effectiveness in exploratory analysis and their applicability to heterogeneous datasets.

3.6 AI-Assisted Interpretation

The system integrates an AI-assisted interpretation module that generates natural language summaries of analytical results.

This module processes structured outputs from the analytical core and produces explanations that highlight key findings, such as strong correlations, lag relationships, and potential limitations.

The purpose of this component is to improve interpretability and support users who may not have extensive statistical expertise.

3.7 Evaluation Methodology

The system is evaluated through case studies using real-world datasets. The evaluation focuses on the system's ability to identify meaningful relationships and patterns across entities.

Two main case studies are conducted:

- Analysis of vaccination, health expenditure, and child mortality
- Analysis of working hours and life expectancy

The results are compared with findings from existing literature to assess their validity and consistency.

This evaluation is qualitative and exploratory in nature, focusing on the interpretability and consistency of the results rather than quantitative performance metrics. This approach is appropriate for exploratory analytical systems, where the primary goal is to support insight generation rather than predictive accuracy.

This evaluation strategy ensures that the system is assessed based on its ability to provide meaningful and interpretable insights rather than purely numerical performance.

3.8 Summary

This chapter described the methodology followed in this study, including the research approach, data sources, analytical methods, and evaluation process. These elements provide the foundation for the system implementation and experimental analysis presented in the following chapters.

4 System Architecture and Design

4.1 Introduction

This chapter presents the architecture and design of the proposed system. The platform is developed as a web-based application that integrates data ingestion, preprocessing, statistical analysis, and AI-assisted interpretation into a unified workflow.

The system is designed with a modular structure to ensure flexibility, scalability, and ease of maintenance.

4.2 System Overview

The platform consists of two main components:

- A web application layer responsible for user interaction and data handling
- An analytical core responsible for data processing and statistical analysis

Users interact with the system through a web interface, where they can upload datasets, trigger analysis, and view results. The backend processes the data and returns structured outputs.

4.3 Architecture Design

The system follows a layered architecture:

- Presentation Layer (User Interface)
- Application Layer (Django backend)
- Analysis Layer (time series processing modules)

This separation ensures that each component operates independently, improving maintainability and extensibility.

As shown in Figure 4.1, the proposed system follows a modular architecture that separates the web interface, application logic, analytical core, result generation, and AI-assisted interpretation components.

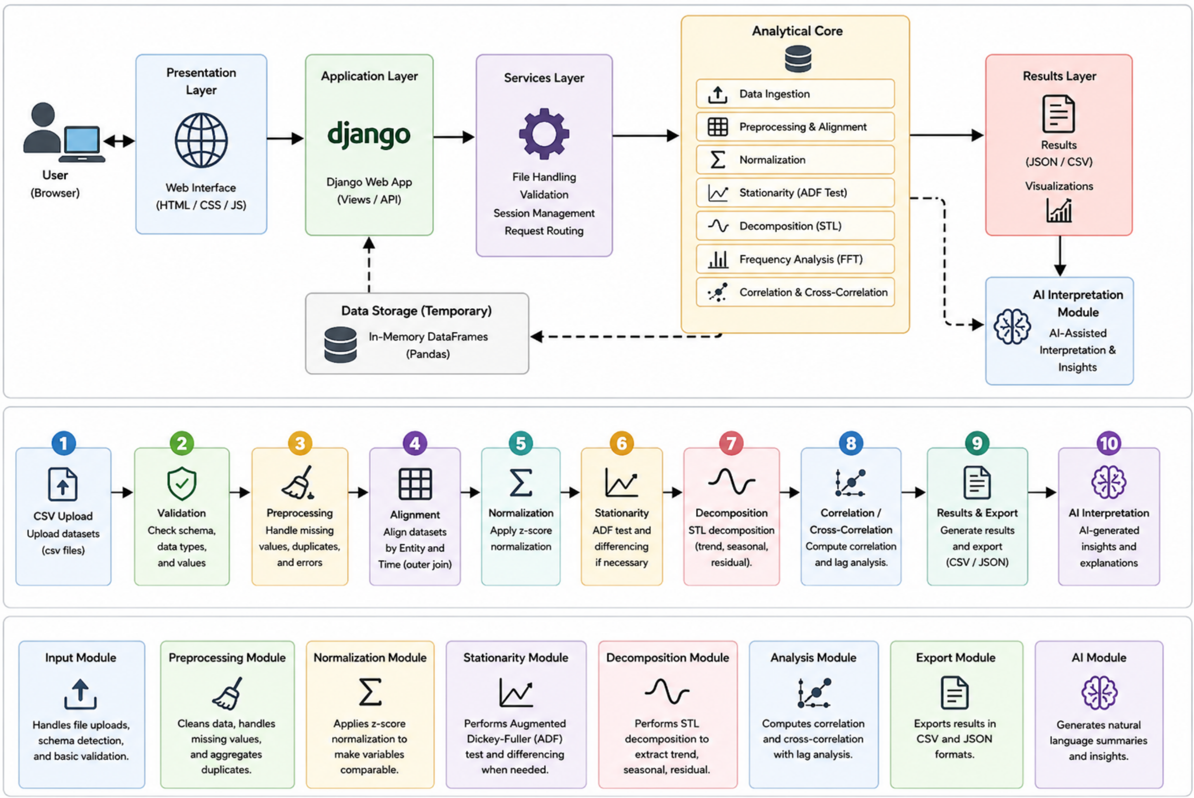


Figure 4.1: System architecture, data processing pipeline, and module breakdown of the proposed platform.

4.4 Web Application Layer

The web application is implemented using the Django framework. It is responsible for handling user requests, managing file uploads, and returning analysis results.

Key functionalities include:

- Uploading CSV datasets
- Validating input data
- Triggering the analysis pipeline
- Displaying results and summaries
- Exporting results in structured formats (CSV and JSON)

The web layer communicates with the analytical core through service functions that process the data and return results.

4.5 Analytical Core

The analytical core is implemented as a separate module that performs all data processing and statistical analysis tasks.

The main responsibilities of the analytical core include:

- Data ingestion and schema detection
- Data preprocessing and alignment
- Normalization of time series
- Stationarity analysis
- Time series decomposition
- Frequency analysis
- Correlation and cross-correlation analysis

The core is designed to be reusable and independent of the web framework. This design enables extensibility and allows the analytical components to be reused independently of the web interface.

4.6 Data Processing Pipeline

The system follows a structured data processing pipeline. The workflow is illustrated in Figure 4.1, which shows the sequence of steps from data ingestion to AI-assisted interpretation.

1. Data upload and validation
2. Data cleaning and preprocessing
3. Alignment of datasets across entities and time
4. Normalization of values
5. Statistical analysis (stationarity, decomposition, correlation)
6. Generation of results and summaries

Each step of the pipeline is implemented as a separate module, ensuring modularity and clarity.

4.7 Module Breakdown

As illustrated in Figure 4.1, the system is organized into distinct modules, each responsible for a specific part of the workflow. The input module handles file uploads, schema detection, and basic validation. The preprocessing module cleans the uploaded datasets, handles missing or invalid values, and aggregates duplicate entries.

The normalization module applies z-score normalization to make variables comparable across different scales. The stationarity module performs Augmented Dickey-Fuller testing and applies differencing when necessary. The decomposition module extracts trend, seasonal, and residual components where sufficient data are available.

The analysis module computes pairwise correlations and lag-based cross-correlations. The export module generates structured output files in CSV and JSON formats. Finally, the AI module produces natural language summaries and explanations based on the analytical results.

4.8 Data Storage and Formats

The system operates primarily on in-memory data structures using data frames. Input datasets are provided in CSV format, while outputs can be exported as CSV or JSON files.

This design allows flexibility in integrating the system with other tools and workflows.

4.9 AI-Assisted Interpretation Module

The system includes an AI-assisted interpretation component that generates explanations of analytical results.

This module receives structured outputs from the analytical core and produces human-readable summaries. The purpose is to help users understand correlations, lag relationships, and potential limitations of the analysis.

The AI module does not replace statistical methods but complements them by improving interpretability.

4.10 System Design Considerations

The system was designed with the following considerations:

- Modularity: separation between web interface and analytical logic
- Scalability: ability to handle multiple datasets and entities
- Usability: simple interface for non-expert users
- Flexibility: support for different types of time series data

4.11 Summary

This chapter described the architecture and design of the proposed system. The platform integrates a web-based interface with a modular analytical core and an AI-assisted interpretation component.

The next chapter presents the implementation details and methodological aspects of the system.

5 Implementation

5.1 Introduction

This chapter focuses on the practical realization of the proposed system, translating the architectural design into a fully functional implementation.

The system is implemented as a web-based application using the Django framework, combined with a modular analytical core for time series processing.

5.2 Technologies Used

The system is developed using a combination of modern technologies:

- **Python**: Used as the main programming language for both backend and analytical processing
- **Django**: Used for the web application framework and request handling
- **Pandas**: Used for data manipulation and preprocessing
- **NumPy**: Used for numerical computations
- **Statsmodels**: Used for statistical analysis, including the Augmented Dickey-Fuller test
- **SciPy**: Used for signal processing and numerical methods

These technologies provide a robust environment for handling time series data and implementing statistical methods.

5.3 System Structure

The system is organized into two main parts:

- The Django-based web application
- The analytical core module

The web application handles user interaction, while the analytical core performs data processing and analysis. Communication between the two components is handled through service functions.

5.4 User Interface

The system provides a simple web-based interface that allows users to upload datasets and initiate the analysis process.

The interface is designed to be simple and user-friendly, enabling users to perform complex analyses with minimal technical knowledge.

The user interface of the system is illustrated in Figure 5.1.

Figure 5.1: Web interface for uploading datasets and initiating analysis.

5.5 Data Ingestion

The system supports the upload of datasets in CSV format. Upon upload, the system automatically detects the structure of the dataset and identifies the relevant columns.

The ingestion process includes:

- Detection of the entity column
- Detection of the time column (Year)
- Identification of the numerical value column

This design allows flexibility in handling datasets with different structures.

5.6 Data Preprocessing Implementation

Data preprocessing is implemented as a sequence of transformations applied to the input datasets.

The preprocessing module performs:

- Conversion of data to appropriate types
- Removal of invalid or missing values
- Aggregation of duplicate records using mean values

- Alignment of datasets using common entity and time fields

An example of the data loading and preprocessing logic is shown below. The system detects the numerical value column, converts the year and value fields to numeric types, removes invalid rows, sorts observations by entity and year, and aggregates duplicate records using mean values.

```
def load_csv_series(file_path, series_name=None):
    df = pd.read_csv(file_path)

    for col in ["Entity", "Year"]:
        if col not in df.columns:
            raise ValueError("Invalid CSV format.")

    value_col = _detect_value_column(df)
    name = series_name or value_col

    keep = [c for c in ["Entity", "Code", "Year", value_col]
            if c in df.columns]
    df = df[keep].copy()

    df["Year"] = pd.to_numeric(df["Year"], errors="coerce").astype("Int64")
    df[value_col] = pd.to_numeric(df[value_col], errors="coerce")

    df = df.dropna(subset=["Entity", "Year", value_col])
    df["Year"] = df["Year"].astype(int)

    df = df.sort_values(["Entity", "Year"]).reset_index(drop=True)

    if df.duplicated(["Entity", "Year"]).any():
        agg = {value_col: "mean"}
        if "Code" in df.columns:
            agg["Code"] = "first"

        df = (
            df.groupby(["Entity", "Year"], as_index=False)
              .agg(agg)
              .sort_values(["Entity", "Year"])
              .reset_index(drop=True)
        )
```

```
    return df
```

The alignment step ensures that all datasets can be compared consistently across entities and time periods.

5.7 Normalization Implementation

The system applies z-score normalization to each time series. This is implemented using standard statistical operations on each dataset.

Normalization is applied independently to each series to ensure comparability across different variables and units.

5.8 Stationarity Analysis Implementation

Stationarity is evaluated using the Augmented Dickey-Fuller test provided by the Statsmodels library.

If a time series is not stationary, the system applies differencing iteratively, up to a predefined limit, to achieve stationarity. The number of differencing steps is recorded and included in the results.

5.9 Time Series Decomposition Implementation

Time series decomposition is implemented using the STL method when sufficient data are available.

For shorter time series, where STL is not applicable, the system applies a simpler moving average approach to approximate the trend component.

This fallback mechanism ensures that decomposition is available for a wide range of datasets.

5.10 Frequency Analysis Implementation

Frequency analysis is implemented using the Fast Fourier Transform (FFT). The system converts time series data into the frequency domain and identifies dominant frequencies.

This allows the detection of periodic patterns within the data.

5.11 Correlation and Cross-Correlation Implementation

The system computes pairwise Pearson correlation coefficients between variables for each entity.

In addition, cross-correlation is implemented to evaluate lagged relationships between variables. The system computes correlations across a range of time lags and identifies the lag with the strongest association.

The implementation evaluates correlations across a range of positive and negative lags and selects the strongest valid association while considering the available overlap between the two series.

```
def cross_correlation(x, y, max_lag=10, min_overlap=3):
    x = np.asarray(x, dtype=float).reshape(-1)
    y = np.asarray(y, dtype=float).reshape(-1)

    n = min(len(x), len(y))
    lags = list(range(-max_lag, max_lag + 1))

    x = x[-n:]
    y = y[-n:]

    corrs = []
    overlaps = []

    for k in lags:
        if k < 0:
            xa = x[:k]
            ya = y[-k:]
        elif k > 0:
            xa = x[k:]
            ya = y[:-k]
        else:
            xa = x
            ya = y

        mask = np.isfinite(xa) & np.isfinite(ya)
        xa2 = xa[mask]
        ya2 = ya[mask]
```

```
        overlaps.append(len(xa2))

        if len(xa2) < min_overlap:
            corrs.append(None)
            continue

        if np.nanstd(xa2) == 0 or np.nanstd(ya2) == 0:
            corrs.append(None)
            continue

        corrs.append(float(np.corrcoef(xa2, ya2)[0, 1]))

    return {
        "lags": lags,
        "correlations": corrs,
        "overlap_n": overlaps,
    }
```

This functionality enables the detection of delayed relationships between time series.

5.12 Result Generation and Export

The system generates structured outputs containing:

- Summary statistics
- Correlation results
- Cross-correlation analysis
- Stationarity information

Results can be exported in CSV and JSON formats, allowing users to further analyze the data or integrate it into other systems. The system presents results in a structured and visual format to facilitate interpretation.

An overview of the generated results is presented in Figure 5.2.

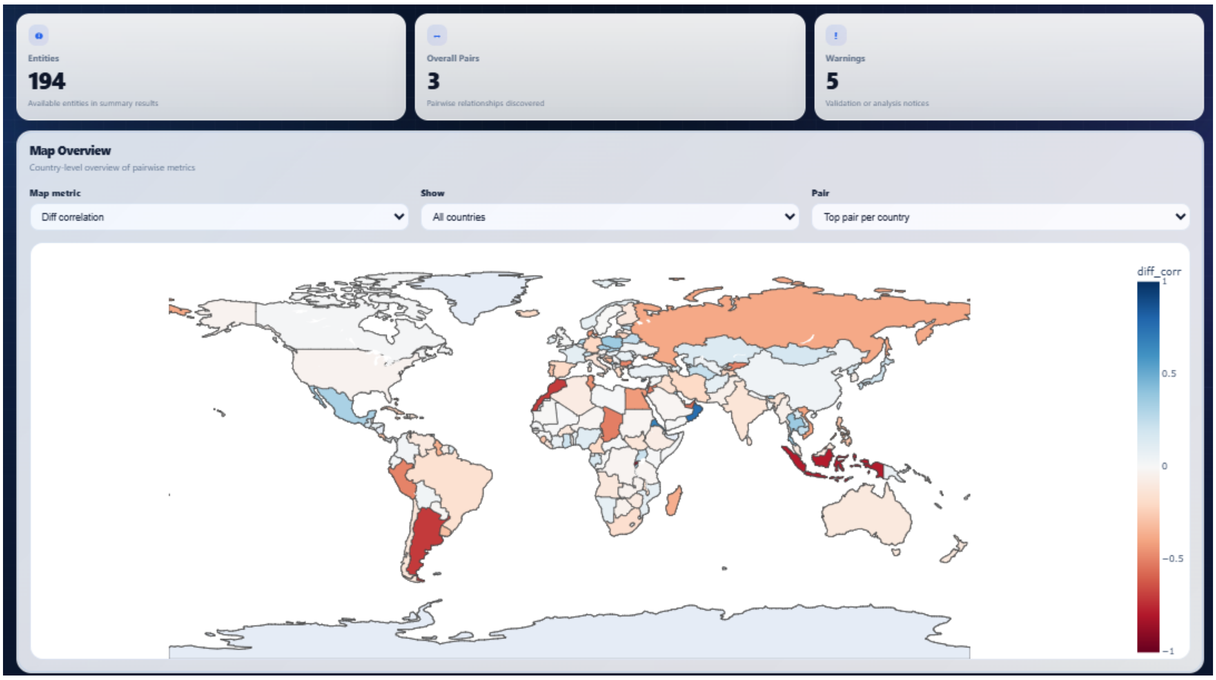


Figure 5.2: Overview of analysis results including entity statistics and geographic visualization.

This structured presentation enhances interpretability and facilitates further analysis by users.

5.13 AI-Assisted Interpretation Implementation

The AI-assisted interpretation module is implemented as a separate component that processes the outputs of the analytical core.

It generates human-readable summaries that describe key findings, such as strong correlations, lag relationships, and potential limitations of the analysis.

This component enhances the usability of the system, particularly for users without a strong statistical background.

5.14 System Deployment

To make the application accessible online, the system was deployed using the Render cloud platform, which supports the deployment of Django-based web applications through integration with GitHub repositories.

The application source code was first uploaded to a GitHub repository, enabling version control and automated deployment. The deployment process required the inclusion of production dependencies in the `requirements.txt` file, including:

- `gunicorn` for serving the application in a production environment
- `whitenoise` for handling static files
- `dj-database-url` for database configuration
- `plotly` for visualization support

A build script was created to automate the deployment process. This script installs dependencies, applies database migrations, and collects static files required by the application.

```
#!/usr/bin/env bash
set -o errexit

pip install -r tsproj/requirements.txt
python manage.py collectstatic --no-input
python manage.py migrate
```

The application was configured on Render as a Web Service connected to the GitHub repository. The main configuration parameters included:

- Build command: `./build.sh`
- Start command: `gunicorn tsproj.wsgi:application --timeout 600`

The extended timeout setting ensures that computationally intensive analysis tasks can be completed without interruption.

During deployment, missing dependencies were identified and resolved based on the platform logs. After successful deployment, the application became publicly accessible at:

`https://time-series-analyzer-wufp.onrender.com`

It should be noted that the free hosting plan introduces limitations, such as automatic inactivity sleep, which may cause initial loading delays. However, once activated, the application performs as expected.

This deployment further demonstrates the system's readiness for real-world usage and its ability to operate as a scalable web-based analytical platform.

The implementation is based on standard statistical techniques commonly used in time series analysis, as presented in the literature, and adapted for integration within the proposed system. Additionally, AI-based tools were utilized during development for code assistance and documentation support [10].

5.15 Summary

This chapter described the implementation of the proposed system, including the technologies used, system structure, and the implementation of each analytical component.

The next chapter presents the experimental evaluation of the system using real-world datasets.

6 Experimental Evaluation

6.1 Introduction

This chapter presents the experimental evaluation of the proposed system. The goal is to assess the effectiveness of the platform in identifying meaningful relationships in real-world time series data.

Two case studies are conducted using country-level datasets. The results are analyzed and compared with existing literature to evaluate the validity of the system.

6.2 Experimental Setup

The experiments are conducted using time series datasets obtained from publicly available sources such as Our World in Data. The datasets include multiple indicators across countries and time periods.

The system processes the datasets using the full analytical pipeline, including preprocessing, normalization, stationarity analysis, decomposition, and correlation analysis.

The system produces pairwise statistical relationships across entities, as shown in Figure 6.1.

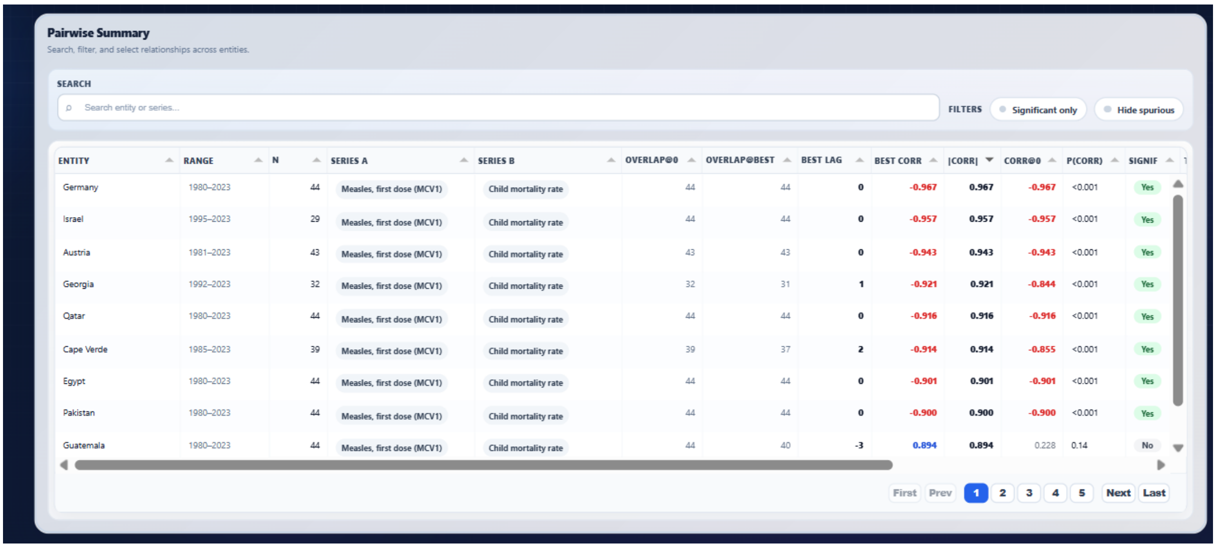


Figure 6.1: Pairwise correlation results across entities.

6.3 Case Study 1: Vaccination, Health Expenditure, and Child Mortality

6.3.1 Vaccination and Child Mortality

The analysis revealed a strong and consistent inverse relationship between measles vaccination coverage (MCV1) and under-5 mortality across countries.

Across 193 entities, the mean correlation was approximately -0.598, with a median of -0.762, and nearly 90% of countries exhibiting negative correlations. Several countries demonstrated particularly strong associations, including Germany (-0.967), Israel (-0.957), and Egypt (-0.901), indicating a near-linear inverse relationship over time.

These results are consistent with global health evidence. According to the World Health Organization, immunization programs have been among the most effective public health interventions, significantly reducing child mortality worldwide [11]. Similarly, reports from UNICEF highlight that increases in vaccination coverage are strongly associated with declines in preventable child deaths [12].

These results are further supported by the scientific literature. Previous studies have demonstrated that expanded vaccination coverage leads to substantial reductions in child mortality, particularly for preventable infectious diseases [13, 14]. This reinforces the validity of the observed inverse relationship and highlights the importance of immunization as a key driver of improved child health outcomes.

Lagged correlation analysis further supports this relationship, as in several countries improvements in vaccination coverage precede reductions in mortality, indicating temporal consistency.

6.3.2 Public Health Expenditure and Child Mortality

To account for differences in healthcare system capacity, public health expenditure was introduced as an additional variable.

Across 52 countries, the mean correlation between health expenditure and child mortality was approximately -0.737, with a median of -0.835, and over 98% of countries exhibiting negative correlations.

This suggests that increased investment in public health systems is strongly associated with reductions in child mortality.

These evidence align with established literature. The World Bank reports that increased health expenditure improves population health outcomes [15]. Similarly, studies published in *The*

Lancet emphasize the importance of healthcare system strengthening [16].

Additional empirical studies support this relationship, showing that higher public health expenditure is associated with improved health indicators, including reduced mortality rates and increased life expectancy [17]. These findings further validate the strong inverse relationship identified in the analysis.

6.3.3 Relationship Between Health Expenditure and Vaccination

The relationship between public health expenditure and vaccination coverage was found to be weaker, with a mean correlation of approximately $+0.151$.

While this indicates a general positive association, the variability across countries suggests that vaccination programs depend on additional factors such as policy decisions, infrastructure, and international support.

6.3.4 Integrated Interpretation

Combining all three variables provides a more comprehensive understanding of the observed dynamics.

The results suggest that:

- Public health expenditure has a strong inverse relationship with child mortality
- Vaccination coverage also shows a strong inverse relationship with mortality
- The relationship between expenditure and vaccination is moderate and variable

These findings indicate that both system-level investment and targeted interventions contribute to improved health outcomes.

6.3.5 Country-Level Validation: Brazil

To further validate the observed global patterns, a country-level case study was conducted for Brazil.

The analysis reveals a strong negative relationship between public health expenditure (as a share of GDP) and child mortality over the period 2000–2022, with correlations ranging from approximately -0.60 to -0.73, and statistical significance ($p \approx 0.0026$). The relationship is primarily trend-driven, with minimal residual correlation, indicating that it is dominated by long-term structural changes rather than short-term fluctuations.

The relationship between the variables can also be examined visually, as shown in Figure 6.2.

Figure 6.2: Normalized time series for Brazil (2000–2022) illustrating trends in public health expenditure, measles vaccination coverage (MCV1), and child mortality.

As shown in Figure 6.2, public health expenditure exhibits an increasing trend over time, while child mortality shows a steady decline. Vaccination coverage remains relatively high for most of the period, with a noticeable decrease in recent years.

This visual evidence further supports the statistical findings, reinforcing the observed inverse relationship between health expenditure and child mortality.

Additionally, a lag of approximately three years was observed, suggesting that increases in public health investment may be followed by improvements in child mortality outcomes after a delay.

The analysis further indicates that this long-term pattern is temporally aligned with major public health developments in Brazil, including:

- The expansion and consolidation of the Sistema Único de Saúde (SUS)
- The scaling of the Family Health Strategy (FHS) during the 2000s
- The introduction of Bolsa Família (2003)
- Progress during the Millennium Development Goals (2000–2015)

These findings are consistent with the scientific literature. A study by Aquino et al. (2009) demonstrates that the expansion of the Family Health Program (FHP) in Brazil was associated with a significant reduction in child mortality, particularly through improved access to primary healthcare services [18].

While the cited study focuses on healthcare coverage rather than expenditure directly, it provides strong evidence that strengthening the health system leads to improved child health outcomes.

Therefore, the results of the present analysis align with existing research: the system identifies a long-term statistical pattern linking health system investment to reduced mortality, while the literature confirms that real-world policy interventions in Brazil had precisely this effect.

Importantly, this comparison suggests that the observed statistical relationship is not merely abstract, but reflects a pattern that is temporally consistent with documented public health reforms and empirically validated outcomes.

6.4 Case Study 2: Working Hours and Life Expectancy

6.4.1 Working Hours and Life Expectancy

The analysis revealed a consistent inverse relationship between working hours and life expectancy across countries.

Across 130 entities, the mean correlation was approximately -0.291, with a median of -0.515, and over 65% of countries exhibiting negative correlations.

Several countries showed strong relationships, including the United Kingdom (-0.966), Spain (-0.923), and France (-0.936).

These findings suggest that higher working hours are generally associated with lower life expectancy.

This is consistent with reports from the World Health Organization, which link long working hours to increased mortality risk [19]. Similarly, the International Labour Organization highlights the negative health effects of excessive working time [20].

Recent large-scale epidemiological studies further confirm that long working hours are associated with increased risk of cardiovascular disease and premature mortality [21, 22]. These findings provide strong empirical support for the observed inverse relationship between working hours and life expectancy.

6.4.2 Interpretation of the Relationship

Further analysis indicates that the relationship is not purely trend-driven.

While a strong long-term inverse trend exists, significant residual correlation suggests the presence of meaningful short-term co-movement.

Lag analysis shows that the strongest relationship occurs at lag 0, indicating that the variables evolve simultaneously rather than sequentially.

This suggests that the relationship reflects both long-term socioeconomic transformation and short-term historical effects.

Importantly, this pattern is consistent with broader research on the determinants of mortality. The study by Cutler et al. demonstrates that increases in life expectancy over the 20th century are strongly associated with improvements in living conditions, healthcare access, and working conditions, including reductions in labor intensity and working time [23].

More generally, time series analysis literature emphasizes that such relationships often reflect complex structural dynamics rather than simple causal mechanisms [24,25]. This highlights the importance of careful interpretation when analyzing long-term temporal trends.

6.4.3 Country-Level Validation: United Kingdom

To further validate the observed global pattern, a country-level case study was conducted for the United Kingdom.

The analysis reveals an extremely strong inverse relationship between working hours and life expectancy over the period 1870–2023, with a correlation of approximately -0.97 and high statistical significance.

This relationship can also be examined visually through time series analysis, as shown in Figure 6.3.

Figure 6.3: Normalized time series for the United Kingdom (1870–2023) showing the inverse relationship between working hours and life expectancy.

As shown in Figure 6.3, working hours exhibit a long-term decreasing trend, while life expectancy steadily increases over time. This strong inverse pattern visually confirms the statistical relationship identified in the analysis.

In addition, short-term fluctuations are also visible, reflecting historical events such as pandemics and socio-economic disruptions, which affect life expectancy in the short run.

Importantly, the system indicates that this relationship is not purely trend-driven, as substantial residual correlation is also present. This suggests that the association reflects both long-term structural changes and shorter-term co-movements.

The detected pattern is temporally consistent with major historical developments in the United Kingdom, including:

- Gradual reduction of working hours through labor regulation (late 19th–20th century)
- Introduction of the National Insurance Act (1911)
- Establishment of the National Health Service (1948)
- Expansion of the welfare state after World War II

These structural changes are supported by economic and public health research. According to Cutler et al., long-term improvements in health outcomes, including life expectancy, are closely linked to systemic changes such as improved healthcare access, better working conditions, and rising living standards [23].

Additionally, the residual component of the relationship aligns with major short-term events, such as:

- The 1918–1919 influenza pandemic
- World War I and World War II disruptions
- The Great Smog of 1952 and subsequent environmental regulation
- The COVID-19 pandemic (2020–2021), which temporarily reduced life expectancy

Empirical studies have confirmed that such shocks can produce short-term deviations in mortality trends. For example, Almond (2006) shows that the 1918 influenza pandemic had significant and measurable effects on population health [26].

Therefore, the results of the present analysis align with existing research: the system identifies a strong long-term statistical relationship between working conditions and health outcomes, while the literature confirms that historical policy interventions and health shocks contributed to these patterns.

Importantly, this comparison suggests that the observed statistical relationship is not merely abstract, but reflects real-world structural transformations and historically documented health improvements.

6.5 AI-Assisted Interpretation Results

The platform provides AI-generated explanations that help interpret statistical results.

Figure 6.4: AI-assisted interpretation of analytical results.

Figure 6.4 presents an example of the AI-generated interpretation interface.

The AI-assisted interpretation module was evaluated using the outputs generated from the analytical pipeline.

The system produces natural language summaries that describe key statistical relationships, including correlation strength, lag effects, and potential limitations. These explanations are designed to assist users in understanding complex analytical results without requiring advanced statistical expertise.

This approach aligns with recent research in interpretable machine learning, which emphasizes the importance of explainability and transparency in analytical systems [27, 28].

The generated interpretations highlight:

- Strong positive or negative correlations between variables
- Lag relationships indicating delayed effects
- Whether relationships are trend-driven or residual-based
- Potential risks such as spurious correlations

Overall, the AI module enhances usability by bridging the gap between statistical output and human interpretation, making the platform more accessible to non-expert users. Additionally, the AI module enables a more interactive analytical experience, allowing users to explore results and receive contextual explanations dynamically.

6.6 Discussion

The experimental results demonstrate that the proposed system is capable of identifying meaningful and consistent relationships in time series data, supporting its validity as a reliable exploratory tool for multi-entity analysis.

The findings are closely aligned with existing literature, reinforcing the appropriateness of the analytical methods employed. The system successfully captures both global patterns and country-specific dynamics, indicating its effectiveness across heterogeneous datasets.

Moreover, the platform provides insights into lagged relationships, enabling a deeper understanding of temporal dependencies between variables. This capability is particularly important in time series analysis, where delayed effects often play a critical role in interpreting real-world phenomena.

Overall, the results indicate that the system effectively captures both large-scale statistical patterns and localized temporal dynamics. This highlights its potential as a flexible and scalable exploratory tool, especially in domains characterized by data heterogeneity and complex temporal interactions.

Furthermore, the integration of statistical analysis with AI-assisted interpretation demonstrates the value of combining quantitative methods with explainable analytical systems. This approach enhances interpretability and supports more accessible, transparent, and informed decision-making for both technical and non-expert users.

6.7 Limitations

Despite the strong results, several limitations should be acknowledged:

- The analysis is based on correlation and does not establish causality. As highlighted in the econometric literature, correlation does not imply causation, and observed relationships may be driven by underlying confounding factors [29].
- Multiple factors influence the observed relationships
- Data availability and quality vary across countries

6.8 Summary

This chapter presented the experimental evaluation of the system using real-world datasets. The results demonstrate that the platform is capable of identifying meaningful patterns and relationships consistent with existing research.

The system can therefore be considered a useful exploratory tool for multi-entity time series analysis.

7 Conclusion and Future Work

7.1 Conclusion

This thesis presented the design and implementation of a web-based platform for multi-entity time series analysis with AI-assisted interpretation. The system integrates data ingestion, preprocessing, statistical analysis, result export, and interpretation into a unified workflow.

The platform was developed using a modular architecture consisting of a Django-based web application and a separate analytical core. This separation improves maintainability and allows the analytical components to be reused independently of the web interface.

The implemented system supports CSV data upload, automatic schema detection, data cleaning, alignment by entity and year, normalization, stationarity testing, time series decomposition, frequency analysis, correlation analysis, and lag-based cross-correlation analysis.

The experimental evaluation demonstrated that the platform can identify meaningful patterns in real-world datasets. The first case study examined vaccination coverage, public health expenditure, and under-5 mortality, while the second case study examined working hours and life expectancy. In both cases, the results were consistent with existing literature, supporting the usefulness of the system as an exploratory analytical tool.

Overall, the proposed system demonstrates how the integration of statistical analysis and AI-assisted interpretation can support modern data-driven research workflows, particularly in domains such as healthcare, bioinformatics, and socioeconomic analysis.

The system has been successfully deployed online and is publicly accessible for demonstration purposes at the following URL:

`https://time-series-analyzer-wufp.onrender.com`

7.2 Research Questions Revisited

The first research question examined how heterogeneous time series datasets can be integrated and analyzed within a unified platform. This was addressed through the design of a preprocessing and alignment pipeline that supports multiple CSV datasets and aligns them by entity and time.

The second research question focused on suitable statistical and computational methods for exploratory multi-entity time series analysis. The implemented system combines normalization, stationarity testing, decomposition, frequency analysis, correlation, and cross-correlation methods.

The third research question examined how AI-assisted interpretation can improve usability. The system addresses this by generating natural language summaries that help users understand statistical outputs and potential limitations.

The fourth research question considered the applicability of the platform to real-world domains such as healthcare and bioinformatics. The experimental case studies demonstrated that the system can be applied to health-related and socioeconomic datasets and can produce results consistent with previous research.

7.3 Limitations

Although the system provides useful exploratory analysis, several limitations remain. First, the platform is based primarily on correlation and cross-correlation methods and therefore does not establish causal relationships.

Second, the quality of the results depends on the completeness, consistency, and temporal coverage of the input datasets. Missing values, inconsistent reporting, and short time series may affect the reliability of the results.

Third, the AI-assisted interpretation component depends on the quality of the statistical outputs provided to it. It should therefore be used as an explanatory support tool rather than as an independent source of scientific conclusions.

7.4 Future Work

Future work could extend the platform in several directions. First, additional statistical and machine learning methods could be integrated, such as forecasting models, clustering techniques, and anomaly detection.

Second, the visualization capabilities of the platform could be improved by incorporating interactive plots, dashboards, and enhanced entity-level comparison views.

Third, the system could be extended to support additional data formats and more flexible time representations beyond yearly observations.

Finally, future versions could include stronger support for biomedical and bioinformatics datasets, making the system more directly applicable to computational intelligence applications in healthcare and life sciences.

7.5 Final Remarks

Overall, this thesis demonstrates that an integrated web-based platform can support exploratory multi-entity time series analysis by combining statistical processing, modular software design, and AI-assisted interpretation.

The proposed system contributes to the growing field of computational intelligence by providing an accessible and extensible tool for exploratory data analysis, with applications in domains such as bioinformatics, public health, and socioeconomic research.

BIBLIOGRAPHY

[1] D. A. Dickey and W. A. Fuller, “Distribution of the estimators for autoregressive time series with a unit root,” *Journal of the American Statistical Association*, 1979.

[2] P. Kolios, “Time series analysis lecture notes,” 2024, department of Computer Science, University of Cyprus, KIOS Research and Innovation Center of Excellence.

[3] R. B. Cleveland and W. S. Cleveland, “Stl: A seasonal-trend decomposition procedure based on loess,” *Journal of Official Statistics*, 1990.

[4] J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex fourier series,” *Mathematics of Computation*, 1965.

[5] P. Kolios, “Fft and spectral analysis lecture notes,” 2024, department of Computer Science, University of Cyprus.

[6] K. Pearson, “Note on regression and inheritance,” *Proceedings of the Royal Society of London*, 1895.

[7] P. Kolios, “Cross-correlation function (ccf) lecture notes,” 2024, department of Computer Science, University of Cyprus.

[8] V. J. Reddi, “Machine learning systems: Principles and practices of engineering intelligent systems,” 2024, open-access textbook. [Online]. Available: https://mlsysbook.ai/

[9] O. W. in Data, “Our world in data,” 2024. [Online]. Available: https://ourworldindata.org/

[10] OpenAI, “Chatgpt (gpt-5.3),” 2026, aI language model used for assistance in code development and documentation.

[11] W. H. Organization, “Immunization coverage,” 2023.

[12] UNICEF, “Child mortality trends,” 2023.

[13] C. e. a. Victora, “Reducing deaths from measles,” *Lancet*, 2003.

[14] G. Alliance, “Impact of vaccination on child mortality,” *Vaccine*, 2019.

[15] W. Bank, “Health expenditure data,” 2023.

[16] “Universal health coverage study,” *The Lancet*, 2020.

[17] M. e. a. Farag, “Health expenditures and health outcomes,” *Health Economics*, 2013.

[18] A. et al., “Impact of health programs on mortality,” *American Journal of Public Health*, 2009.

[19] W. H. Organization, “Working hours and mortality,” 2021.

[20] I. L. Organization, "Working time and health," 2022.

[21] F. e. a. Pega, "Global burden of disease from long working hours," *Environment International*, 2021.

[22] M. Kivimäki, "Long working hours and risk of coronary heart disease," *Lancet*, 2015.

[23] C. et al., "Determinants of mortality," *Journal of Economic Perspectives*, 2006.

[24] G. Box, G. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting and Control*. Wiley, 2015.

[25] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2018.

[26] D. Almond, "Is the 1918 influenza pandemic over? long-term effects of in utero exposure," *Journal of Political Economy*, 2006.

[27] F. Doshi-Velez, "Towards a rigorous science of interpretable ml," *arXiv*, 2017.

[28] Z. Lipton, "The mythos of model interpretability," *Queue*, 2018.

[29] C. Granger, "Investigating causal relations by econometric models," *Econometrica*, 1969.

APPENDICES

APPENDIX A

System Interface and Additional Outputs

This appendix presents additional interface components and system outputs of the proposed platform. The included figures illustrate the main functionalities of the system, including dataset upload, statistical analysis, interactive visualization, AI-assisted interpretation, and export capabilities.

A.1 Dataset Upload Interface

The platform allows users to upload one or multiple CSV files for analysis through the main control interface.

Figure A.1 illustrates the dataset upload workflow, where users select the input datasets before initiating the analytical pipeline.

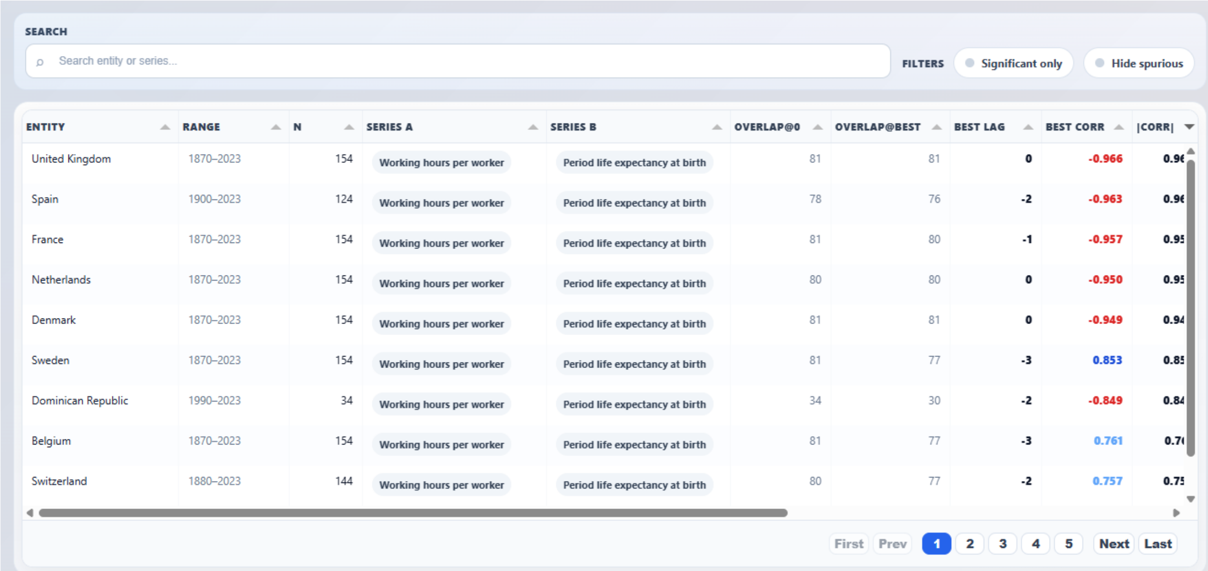
The interface supports automated processing of the uploaded datasets, including preprocessing, normalization, stationarity analysis, decomposition, and pairwise correlation analysis.

Figure A.1: Dataset upload interface and analysis controls.

A.2 Pairwise Correlation Results Interface

The platform provides an interactive interface for exploring pairwise statistical relationships across entities and variables.

Figure A.2 presents the main correlation analysis table generated by the system. The interface allows users to inspect relationships between pairs of time series across multiple entities.



ENTITY	RANGE	N	SERIES A	SERIES B	OVERLAP@0	OVERLAP@BEST	BEST LAG	BEST CORR	|CORR|
United Kingdom	1870–2023	154	Working hours per worker	Period life expectancy at birth	81	81	0	-0.966	0.96
Spain	1900–2023	124	Working hours per worker	Period life expectancy at birth	78	76	-2	-0.963	0.96
France	1870–2023	154	Working hours per worker	Period life expectancy at birth	81	80	-1	-0.957	0.95
Netherlands	1870–2023	154	Working hours per worker	Period life expectancy at birth	80	80	0	-0.950	0.95
Denmark	1870–2023	154	Working hours per worker	Period life expectancy at birth	81	81	0	-0.949	0.94
Sweden	1870–2023	154	Working hours per worker	Period life expectancy at birth	81	77	-3	0.853	0.85
Dominican Republic	1990–2023	34	Working hours per worker	Period life expectancy at birth	34	30	-2	-0.849	0.84
Belgium	1870–2023	154	Working hours per worker	Period life expectancy at birth	81	77	-3	0.761	0.7
Switzerland	1880–2023	144	Working hours per worker	Period life expectancy at birth	80	77	-2	0.757	0.75

Figure A.2: Interactive pairwise correlation analysis interface.

The table includes several statistical metrics and analytical indicators, including:

- The analyzed entity and temporal range
- The compared time series variables
- Correlation values at lag 0 and at the optimal lag
- Best lag identification through cross-correlation analysis
- Statistical significance indicators
- Trend, residual, and differenced correlations
- Detection of potentially spurious relationships

The interface also provides filtering and search capabilities, enabling users to focus on statistically significant relationships and exclude potentially misleading correlations.

Correlation values are color-coded to improve interpretability, allowing users to quickly identify strong positive and negative relationships across entities.

A.3 Entity Time Series Visualization

The platform provides interactive visualization tools for exploring temporal relationships between variables at the entity level.

Figure A.3 presents the entity visualization interface for the United Kingdom, showing the relationship between working hours and life expectancy over time.

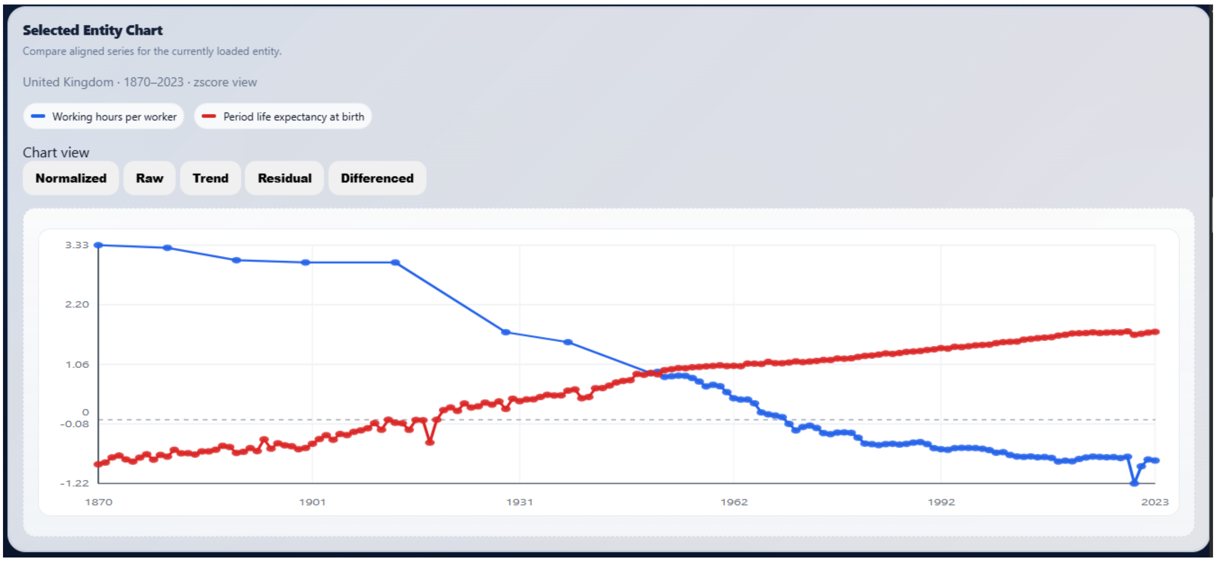


Figure A.3: Interactive entity-level time series visualization interface.

The interface supports multiple visualization modes, including:

- Normalized series visualization
- Raw time series inspection
- Trend component analysis
- Residual component analysis
- Differenced series visualization

These visualization modes enable users to examine both long-term structural patterns and short-term fluctuations in the data.

The chart interface also supports aligned comparison of multiple variables for a selected entity, allowing users to visually inspect correlation patterns, temporal synchronization, and potential lag relationships identified during the statistical analysis.

Such interactive visual exploration complements the quantitative analysis by improving interpretability and enabling deeper inspection of country-level temporal dynamics.

A.4 Geospatial Correlation Visualization

The platform also provides geospatial visualization capabilities for exploring country-level statistical relationships.

Figure A.4 presents the interactive world map interface used for visualizing pairwise analytical metrics across entities.

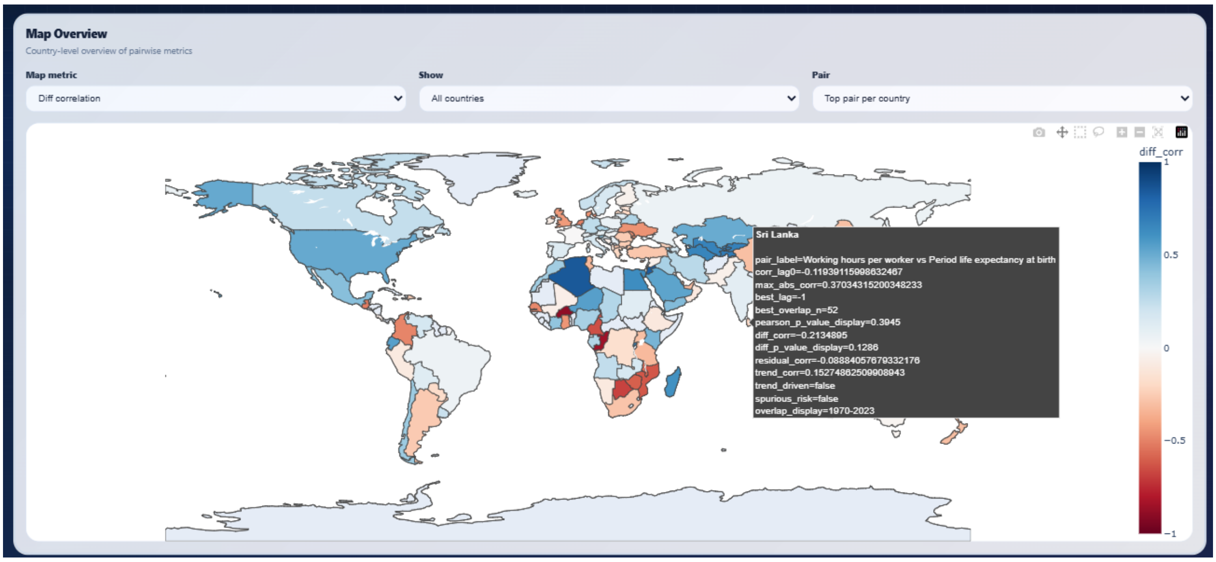


Figure A.4: Geospatial visualization of country-level analytical metrics.

The map interface enables users to visualize statistical indicators geographically, facilitating the identification of regional patterns and country-level variations.

Users can dynamically explore multiple analytical metrics, including:

- Correlation coefficients
- Trend and residual correlations
- Differenced correlations
- Statistical significance measures
- Spurious relationship indicators

The interface also supports interactive inspection through hover-based metadata visualization, allowing users to examine detailed statistical information for individual entities directly on the map.

This geospatial analytical layer complements the tabular and time series visualizations by providing an additional perspective for interpreting large-scale multi-entity temporal relationships.

A.5 AI-Assisted Interpretation Interface

In addition to statistical analysis, the platform provides AI-assisted interpretation capabilities that generate human-readable explanations of analytical results.

Figure A.5 presents the AI interpretation interface integrated into the analytical workflow.

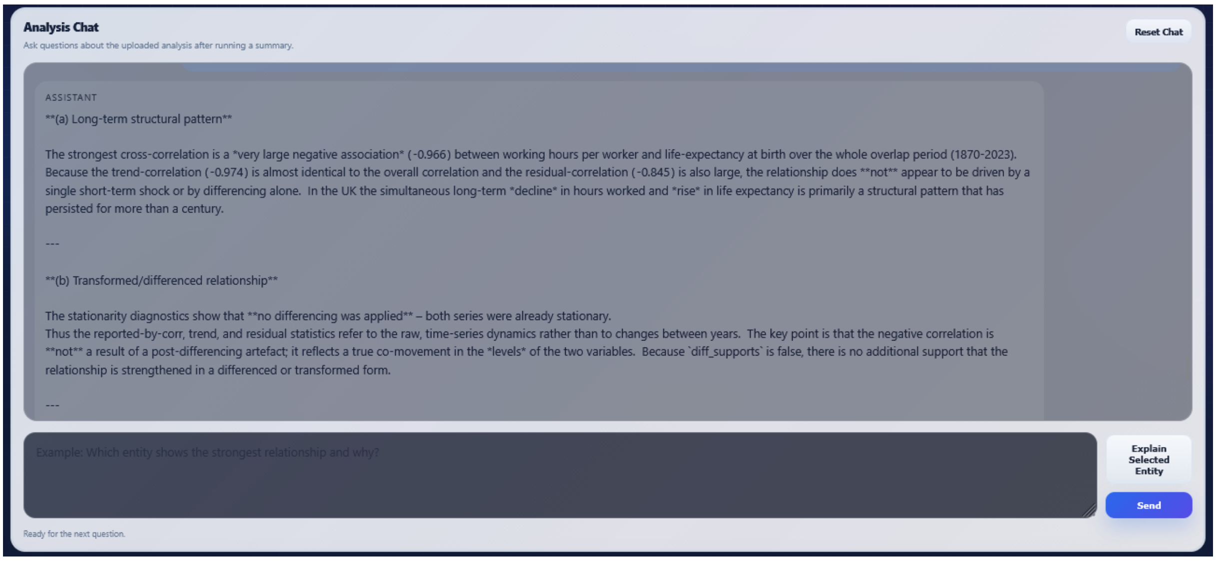


Figure A.5: AI-assisted interpretation interface for analytical results.

The AI module analyzes the statistical outputs generated by the system and produces contextual explanations describing:

- Correlation strength and direction
- Lag relationships between variables
- Trend-driven and residual-based relationships
- Potential limitations and risks of spurious correlations

This functionality improves interpretability by translating quantitative outputs into natural language explanations that can be more easily understood by non-expert users.

The integration of explainable AI techniques with statistical analysis enhances the usability of the platform and supports more transparent interpretation of complex temporal relationships.

A.6 Export and Reporting Features

The platform provides multiple export options that allow users to save and reuse analytical results outside the application environment.

Figure A.6 presents the export controls integrated into the main analysis interface.

Figure A.6: Export and reporting controls available in the platform.

The system supports exporting analytical outputs in multiple formats, including:

- CSV files containing processed and transformed datasets
- Structured JSON outputs for programmatic access
- ZIP archives containing complete analytical result packages
- PDF reports summarizing statistical findings and visualizations

These export capabilities improve reproducibility and facilitate integration with external analytical workflows and reporting processes.

In addition, the export functionality enables users to preserve intermediate analytical results, supporting further statistical analysis and documentation outside the platform environment.