Knowledge-based Analysis of Open Data

Marius Marian Dinu and Anca Daniela Ionita
Automation and Industrial Informatics Department, Faculty of Automatic Control and Computer Science, National University of Science and Technology POLITEHNICA Bucharest, Romania

marius_marian.dinu@upb.ro
anca.ionita@upb.ro

Abstract: There is a shortage between the emerging availability of time stamped open data and the capacity of extents in the application sphere to turn it into a reliable source of knowledge without the help of data scientists. Time series now concern every aspect in our daily life, from environmental monitoring to music, or the telemetry of our actions on social media. This struggle of understanding the underlaying data behaviour deepens when considering the heterogeneity problem of the scattered series, that must be handled consistently to produce a reliable source of knowledge. The current approach proposes to target this bottleneck by generating a versatile and domain-neutral representation of data, containing statistics, designed to be consumed both by people (via a brief interaction) and by systems (via a standardized enriched structure). This method has the aim of converting tacit data into explicit artefacts, prioritizing time series knowledge representations. The paper will present a knowledge-based approach to time series interpretation, which takes a dataset from an open data source, without domain assumptions, transforms it into a common format, with semantic enrichment of metadata and includes analytical results. The open data are mapped onto a sub-part of the SemTS (Semantic Time Series) ontology, so that series characteristics can be stored and reused as knowledge artefacts. The final output is a knowledge specialized JSON, partitioned into insightful sections (e.g. dimensions, feature families), plus explanations in natural language, with the aim of reducing comprehension barriers for non-specialists and facilitating subsequent integration into knowledge management flows, like cataloguing, searching or comparing. The proposed format encompasses insights regarding the series, drawing a profile for the series that includes quality indications (e.g. missingness, gaps, outliers), trend patterns, seasonality (cyclicality). The contribution is a practical knowledge aspect for time series analysis, a tool to convert raw time-stamped open data into reusable knowledge artefacts, that could guide downstream decisions in the data discovery procedure. The intent is to assist with accessible diagnostics, rather than to replace analytical expert judgment.

Keywords: Knowledge-based data structure, Open data interoperability, Time series knowledge, Semantic enrichment, Insight extraction, Analytical results

1. Introduction

As a result of digital transformations, open data has changed its purpose, from a transparency initiative to a pillar of knowledge that drives decisions. However, this is not enough because, while the volume of available data continues to pile up, the insight gap between the raw data provided and the usable knowledge widens continuously. Open data sets capture the information into a rigidly syntactic form, without semantical context, sinking the underlaying knowledge (Damagatla and Bossi, 2025). The transition from the simple availability of data to the generation of meaningful value in society and economy remains an uneven process, marked by a divergence between the published information and the actual absorption capacity of the market, mainly caused by heterogeneity and lack of standardization (e.g. different formats, inconsistent metadata, unclear meaning) of data sources (Ortiz-de-Urbina-Criado, Abella and Garcia-Luna, 2023). Without processing, data remains in a stationary state, with no added value, insufficient to sustain complex decision-making. A major challenge in this fast-paced landscape is the need for uniformity of data infrastructure that should ensure not only accessibility, but also interoperability for integrated analytics (Davies et al, 2019).

In this state of constant data volume expansion, the knowledge management (KM) tools must go beyond the role of storing resources and step in, introducing active components that ensure the capacity to absorb data properly (Kamalruzaman, Mohd and Maaz, 2025). The limitations in the current landscape of temporal data management and the lack of widely used standardized models limit the reuse of data and reduces the fusion of results originated from different analytical methods (Moreira, 2024).

This paper seeks to spotlight and remedy aspects of these problems by offering an insight into how data can be transformed from an inert resource into a tool for knowledge sharing. It positions itself within the KM framework by addressing the representation of time series originating from public repositories, converting raw data into structural knowledge, emerging when the data is contextualized, interpreted and semantically organized. This paper is organized as follows: firstly, it positions the study within knowledge management and semantic analysis of open data; we draw the background section acknowledging the open data reuse and ontology-based approach used. Following, the third section frames the method, describing the current state of the research based, proposing a pipeline that turns open-source time series data into semantically structured knowledge. In section 4, the solution is instantiated on a sustainability-oriented open dataset to assess its practical applicability. Finally, we conclude with a discussion of the empirical results that point the contribution of knowledge capturing, reuse and interoperable capabilities.

2. Background

Hahnel et al. (2026) report reflects a growing transition in the awareness of practicing the FAIR (findable, accessible, interoperable and reusable) guiding principles for scientific data management, that serve as a rulebook to data producers and publishers in their activity. There has been an increase rate of active users that use artificial intelligence (AI) for data processing, transitioning to advanced knowledge management practices.

A possible solution is the integration of an existing ontology into a tool for analysing open data, where the datasets constitute raw data that acquire informational value by structured contextualization and become knowledge when it is interpreted, enabling decision-making actions (Rowley, 2007). Although the open data processing is considered as a technical endeavour, there are premises to transform open data into organizational information and expose the knowledge through an ontology (Kawtar, Hind and Dalila, 2022).

The Semantic Time Series (SemTS) ontology was used in our approach to accommodate the need for semantic structuring of insight derived from multivariate time series studies, blending them with domain specific expertise (Grass et al, 2025). SemTS was conceived to identify and describe segments within time series, defined as specific data points or time intervals, each segment caring its own knowledge characteristics, associated to statistical features, anomalies, patters, etc. or priori information provided by experts in the field. In this ontology, the time series segment class extends classes from the DCAT (Data Catalog) vocabulary to allow segments to be part of other segments, providing a hierarchical structure that reflects the fractal nature of temporal data (Grass et al, 2024). DCAT conforms to RDF (Resource Description Framework) vocabulary that model data interchange on the web, representing the information as a graph.

By adopting the SemTS ontology, the tool proposed in the paper employs knowledge graphs (KG) to present the information in a more accessible manner for semantic searches or automatic reasoning in KM systems (Mohammad, Nachouki and Mohamed, 2025). KG represent more than a static database, but a way to model the expertise and structure of a specific subject, so that a machine can easily subtract the semantics behind the data, while maintain human readability (Grass et al, 2023). KG enable one to represent stored information as a dynamic interpretable knowledge ecosystem by AI systems, providing meaning understanding withing the ontological representation (Peng et al, 2023).

3. Knowledge-based Analysis

3.1 Method

The proposed method spirals around the idea that time series should not only be handled as tabular static structures, but also as resources that could be organized, described and reused, without losing their semantic structure, to achieve the knowledge level. Thus, we chose to explore the capabilities of SemTS ontology, which provides an applicable blueprint for representing time series, exploring their temporal segments, characterizing attributes and the information extracted from analysis. Based on this theory, the work aims to transform a standard time series representation into an explicit semantic form, compliant with SemTS.

To implement this approach for analysing open data, we developed a software tool that integrates the ontology guidelines and applies feature extraction methods into the data manipulation process. Afterwards, the tool was experimentally evaluated on a public dataset, exemplifying a possible solution to demonstrate the practical applicability of the tool and underline the leverage offered by the ontology in structuring knowledge artefacts.

Overall, this method is not limited to a simple conversion of a file from a format to another, but it defines a systematic process that transforms temporal data into semantic information objects enriched with analytical features. Looking from this perspective, the research contribution consists in the semantic formalization of a time series according to the SemTS ontology, whereas integrating a characterization component, with the purpose of representing data as a conglomerate of knowledge artifacts, organized into a knowledge graph. In this light, this approach responds to the need of data analysis, while also achieving the need for knowledge management, aiming for a reusable and interoperable representation.

The method follows the logic of the DIKW (data, information, knowledge, wisdom) hierarchy (Rowley, 2007) and goes through a gradual mechanism, emphasising different levels of processing. Data corresponds to the raw tabular contents without an explicit role, information reveals after establishing the semantics of the columns, contextualizing the observations and knowledge emerges from the structure build on top of these observations, allowing reasoning.

3.2 Proposed Tool for Open Data Analysis

The data conversion and analysis tool is implemented as a Python CLI (Command-Line Interface) application that loads a time series from a tabular form and a set of running settings, and returns its representation accompanied by the derived analytical features, according to SemTS formalities. From a technical point of view, the tool is projected to process the data flow in sequential phases, from ingesting and formalizing the input presented as a CSV (Comma Separated Values) to generating a serialized graph response, in form of a JSON-LD (JavaScript Object Notation for Linked Data) and representing it as a knowledge graph. This structural approach translated the series data into a knowledge artefact, which can be interrogated or visualized into a hierarchical manner. In addition, the tool incapsulates an inspection mechanism that allows the visualization of the graph structure, so the user can easily understand the relationships between elements and their properties, facilitating a knowledge-oriented data exploration initiative.

To ensure the generality of the solution, while considering the particularities of the input datasets, the tool call is parametrized with a specific configuration to adapts the process to the data needs. The configuration sets key structures on the input data, as the entity identification field, the label field of the entity, the time field, the time interpretation option (the way to represent the time in a uniform manner), the encoding granularity of the records and the presence of the descriptive features resulted from the analysis.

To emphasize the analysis fact, besides the representation of the input data, the result integrates semantic knowledge extracted from numerical transformations over the data. The method integrated an analysis step based on features, inspired from the framework presented in Hyndman et al (2025), converting the series data points into knowledge measures that capture its behaviour. The features extracted using the *tsfeatures* Python package are organized into thematic families, targeting some aspect of the temporal dynamics of the data (such as statistical description – mean, median, variance; autocorrelation; etc.).

3.3 Data to Knowledge Flow

The tool follows a data-to-knowledge flow, in which raw data are semantically formalized and decorated with descriptive features of the time series, generating semantic knowledge objects. This approach supports the transfer of knowledge, because it contains the context information of the studied entity.

In our context, knowledge is defined as semantically structured information, that can be contextualized. The input table values are treated as data, becoming information when associated with the ontology element; knowledge surface when overviewing the whole representation. The time series knowledge is organized into a semantic infrastructure that allows further explorations (queries, comparisons, integrations with other knowledge flows).

Internally, the tool can be described as a sequence of specialized stages (Figure 1), organized as a semantic processing pipeline. Its architecture segregates data loading, structural role configuration, ontological representation according to SemTS, observation encoding, derived knowledge integration and result serialization, followed by visual representation as a knowledge graph.

Data ingestion. The time series is loaded along with the build configuration, and it is normalized so it can be processed uniformly, regardless of the source particularities. In this stage, the tool manages the missing values, based on the policy chosen by the user, either by omitting them or represent them explicitly.

Structural binding. The build configuration is applied to mark the columns that play the role of entity, label and time; the remaining columns are interpreted as series dimensions, supporting the construction of generic schema for semantic structuring.

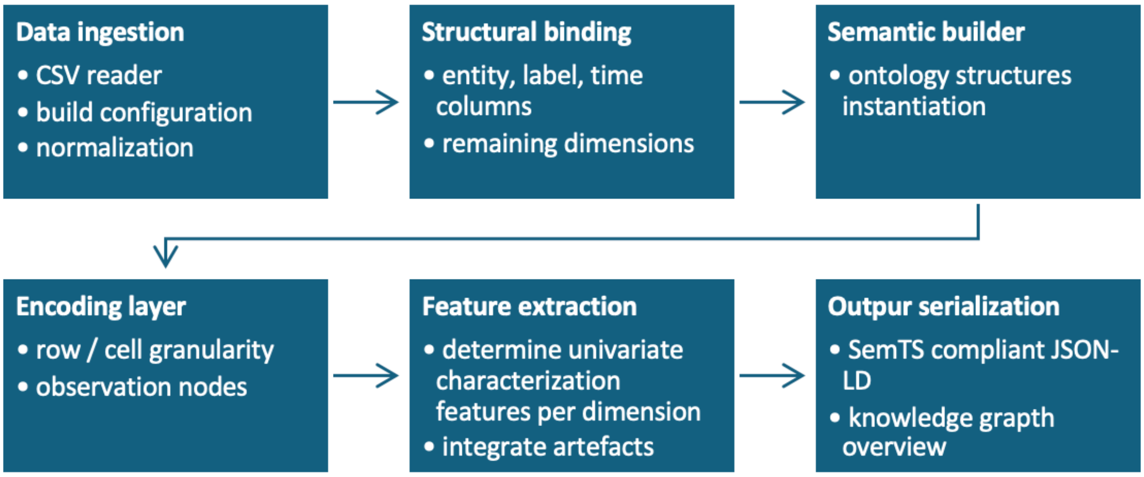

Figure 1: Data flow for knowledge-based analysis

Semantic builder. Based on the previous interpretation, the tool builds ontological structures for the main segment of the series, the temporal segments and the semantic dimensions, using SemTS classes and properties.

Encoding layer. The observations nodes are generated accordingly to the chosen granularity from the raw data and are bonded with the segments.

Feature extraction. If the analytics are enabled, the tool computes univariate features per dimension and integrates them as derived knowledge, along with their provenance.

Output serialization. The resulting structure is serialized as a JSON-LD, which could be auxiliary exported with a graphical representation.

3.4 Semantic Correspondences

The first step in the structuring process is assigning the analysis unit, the time series being decomposed into three structural components: entity, time, and dimensions, laying the support for all subsequent operations. The entity defines the central unit of observation, representing the element that is measured and holds the knowledge. Time constitutes an order reference of the observations and dimensions are the actual variables that characterize the entity at a time. Afterwards, these concepts are instantiated as class objects of SemTS ontology; the equivalence between the elements of the time series and the ontology representation is illustrated in Tabel 1. SemTS provides knowledge at the conceptual level, defining how time series data elements (segments, dimensions, values) can be semantically described.

Before any semantic formalization, the time is converted to a generic format according to the OWL-Time (Web Ontology Language). To ensure a deterministic behaviour of the output, under the same factors, there are defined stable IRIs (Internationalized Resource Identifiers) for the graph nodes.

The main challenge lies in organizing the process of the time series into a hierarchical configuration within the SemTS ontology. A segment (*semts: TimeSeriesSegment*) is defined as a basic unit that can represent either a complete series, a sub-period of it, or even a projection onto a single dimension, an aspect that spots the flexibility to use the class for different level of semantic granularity, different layer, but same meaning.

For each entity in the dataset, one creates a parent segment node, which serves as a container for the entire temporal history that is going to be connected to all dimensions (*semts: segmentDimension*) identified. This architectural approach allows multivariate structured series handling. Observations at a given moment are instantiated as sub-segments, connected to the main segment (via *semts: isPartOf*), and have attached their own temporal index (semts: segmentIndex), attached to a time node (*time: Instant*).

The encoding of the values originated from time series in the semantic graph holds a challenge related to the equilibrium between the expressivity of the output and the performance of the conversion. The tool proposes two strategies regarding the granularity of the result, each with its set of technical and analytical implications.

Table 1: Proposed time series data mapping to SemTS ontology

Input time series element	Proposed semantic role	SemTS correspondent semts: …	SemTS properties semts: …
Structural elements			
Complete series of an entity	Main segment of the series	*TimeSeriesSegment*	*title, segmentDimension, segmentIndex*
Time observation	Time segment of the series	*TimeSeriesSegment*	*isPartOf, segmentIndex, rowIndex, title*
Projection of the series on a single dimension	Segment on a dimension	*TimeSeriesSegment*	*isPartOf, segmentDimension, segmentKnowledge, title*
Analytical indicator / column	Series dimension	*DataDimension*	*title, dimensionPosition*
Relationship between full series and dimensions	Defining multivariate structure	Segment – dimension relationship	*segmentDimension*
Relationship between full series and time observations	Series hierarchy	Relationship between segments	*isPartOf*
Time associated with a segment	Semantic temporal index	Segment index	*segmentIndex*
Content elements			
The complete observation of a row (row granularity)	Knowledge attached to the temporal segment	*SegmentDataCharacteristic*	*segmentKnowledge, hasValue*
An individual value per dimension (cell granularity)	Knowledge point		
The compact value of the observation	Embedded value of the observation	*EmbeddedValue*	*valueString*
The associated individual value	Embedded value of the point		
Extracted features (mean, acf, entropy, trend, etc.)	Derived characteristics of the series	*SegmentDataCharacteristic*	*title, description, hasValue, knowledgeGenerationEntity*
Value of a feature	Embedded value of the feature	*EmbeddedValue*	*valueString*
Knowledge elements			
The complete set of features for a dimension	Main knowledge group	*KnowledgeGroup*	*segmentKnowledge, groupKnowledge, title*
Family of features (stats, acf, stl, etc.)	Thematic knowledge group	*KnowledgeGroup*	*groupKnowledge, title*
Feature calculation method	Methodological provenance	*KnowledgeGenerationMethod*	*title, description*
Relationship between feature and calculation method	Provenance relationship	Knowledge – method relationship	*knowledgeGenerationEntity*

The row granularity design attaches each temporal segment to a single knowledge node (*semts: SegmentDataCharacteristic),* which contains all dimensions in a JSON object serialized as a string (*valueString*); this is recommended for large datasets, since it maintains the cohesion of the observation for a time point, reducing the number of nodes in the graph. The cell granularity expands a segment, a knowledge node and a value node for each individual cell. Even though it offers semantic navigation, this is suitable for small datasets, since it generates extensive elements in the graph.

If the features option is enabled by configuration, the tool determines descriptive features for each dimension of the time series and adds them to the structure. The tool extracts the values over time for each dimension and forms a univariate series, on which one applies feature calculations. To integrate these results into the representation, it first creates a semantic segment that represents the series projected on a single dimension, then knowledge nodes describing the features are attached to each segment. The structure is managed by *semts: KnowledgeGroup*, each individual feature being represented by *semts: SegmentDataCharacteristic*; the compute method is preserved by *semts: KnowledgeGenerationMethod*.

4. Case Study for Open Data Analysis

4.1 Dataset Preparation

The case study aims to illustrate how the flow projected within the proposed tool transforms a time series into a semantic structure focused on knowledge representation. The example gives a demonstrative instantiation of the ontological model with an explicit configuration, highlighting the use of the ontology classes, properties and the semantic relationships that results between the elements resulted from the formalization of the time series.

The dataset choice has focused on sustainability-related data provided by a recognized source; the search concluded with the finding of a dataset regarding the sustainable development goals (SDGs) from the United Nations (UN) organization. We considered this data suitable to KM analysis since SDGs are used internationally for reporting the state of the sustainability development efforts by each country, so it needs to be standardized to perform correct assessments of comparison and interpretation. Looking forward, this transformation of data into a knowledge-based format allows its visualisation as a graph and querying to identify behaviours in achieving sustainability goals.

From the full database – *SDR2025-data.xlsx* (Sachs et al, 2025), we explored a part of the *Backdated SDK Index* data (columns A to V). The records of each country appear as an annually observed entity, and each record corresponds to a temporal observation described by a set of dimensions. We selected a part of this data set (rows 3627 to 3651), i.e., the series regarding the evolution of the SDG indexes corresponding to Romania, each row representing an annual observation. Before feeding the data to the tool, we extracted the referred piece and converted it into a CSV with the structure presented in Table 2, along with the column's structural role.

Table 2: Input data structure, columns description and structural role in the ontology representation

Tabular columns	**id**	**country**	**year**	**indexreg**	**sdgi_s**	**goal1, ..., goal17**
Description	Country code	Country name	Measurements time	Continent identifier	SDG index score	SDG measurements
Structural role	Entity identifier	Entity label	Temporal index	Series dimensions		

By passing this structure through the ontological tool, it gets reinterpreted as a semantic map and each component gain a certain role. Generally, for any time series, an entity identifier, a label and a temporal index are defined, and the remaining columns are handled as dimensions of the series.

Specifically, the tool performs a generic interpretation scheme with a custom user specification, and certain roles are established, guiding the tool how to interpret the initial structure. In this case, the build configuration (Figure 2) provides entity (*id*), label (*country*), time (*year*), granularity type and whether to include features; the remaining columns (*indexreg, sdgi_s, goal1 – goal17*) are interpreted according to a generic mapping rule, as analytical dimensions. The tool is not depended on fixed column names, but on structural roles that are configured or inferred according to a set of processing rules, so this logic could be reused.

```
config = BuildConfig(
    base_iri="https://example.org/kg/semts/sdr2025/",
    entity_col="id",
    label_col="Country",
    time_col="year",
    time_mode="year",
    granularity="row",
    omit_missing=True,
    include_features=True,
    feature_set="extended",
)
```

Figure 2: Example run build configuration

4.2 Results and Discussion

The entire series associated with Romania (Figure 3) is instantiated as a node of type *semts: TimeSeriesSegment*, representing the main segment of the model and functioning as an aggregation point for segments, dimensions and derived segments. For each temporal index (year, in this case), the tool creates a distinct time segment of the same class type, connected to the main node via *semts: isPartOf*, to express the hierarchy between the whole series and its time observations. Each time segment is indexed via semts: segmentIndex, pointing to an OWL-Time entity and it is indirectly controlled by the configuration (specification of the time index attribute column and the time mode).


```json
{
  "@id": "https://example.org/kg/semts/sdr2025/segment/ROU-series",
  "@type": "semts:TimeSeriesSegment",
  "semts:title": "Time series for Romania (ROU)",
  "semts:segmentIndex": {
    "@id": "https://example.org/kg/semts/sdr2025/time/interval/ROU/2000-01-01T00-00-00Z__2025-01-01T00-00-00Z"
  },
  "semts:segmentDimension": [
    { "@id": "https://example.org/kg/semts/sdr2025/dimension/indexreg" },
    { "@id": "https://example.org/kg/semts/sdr2025/dimension/sdgi_s" },
    { "@id": "https://example.org/kg/semts/sdr2025/dimension/goal1" }
  ],
  "semts:segmentKnowledge": {
    "@id": "https://example.org/kg/semts/sdr2025/knowledge-group/series-metadata/ROU"
  }
}
```


Figure 3: Example output section representing the entity with some of the dimensions

The analytical indicators are converted into instances of *semts: DataDimension* class, i.e. each indicator is not only studied as a column name, but as a semantic dimension of the series. The link between the complete series and its dimensions is forged but the *semts: segmentDimension* property.

The annual observations encoded using row granularity strategy are exposed in Figure 4; this approach treat each temporal segment being assigned to a single knowledge node of type *semts: SegmentDataCharacteristic*, connected via *semts: segmentKnowledge* to its corresponding segment. This node holds the observation's values, expressed via *semts: hasValue* to a *semts: EmbeddedValue* and it stores them as a JSON object serialized in *semts: ValueString*.


```json
{
  "@id": "ROU-t-2010-row10",
  "@type": "semts:TimeSeriesSegment",
  "semts:segmentKnowledge": {
    "@type": "semts:SegmentDataCharacteristic",
    "semts:hasValue": {
      "@type": "semts:EmbeddedValue",
      "semts:valueString": "{\"goal1\":\"77.279\",...}"
    }
  }
}
```

```json
{
  "@id": "ROU-t-2000-row0-dim-goal1",
  "@type": "semts:TimeSeriesSegment",
  "semts:segmentDimension": { "@id": "goal1" },
  "semts:segmentKnowledge": {
    "@type": "semts:SegmentDataCharacteristic",
    "semts:hasValue": {
      "@type": "semts:EmbeddedValue",
      "semts:valueString": "77.279"
    }
  }
}
```


Figure 4: Compact section of an observation represented with row (left) / cell (right) granularity

In the cell granularity strategy (Figure 4), the temporal observation is no longer treated as a single compact package, but it is decomposed into individual values, each represented separately in the representation. Thus, each series dimension generates a distinct semantic unit associated with the time segment. A tabular record is represented by a semantic segment associated with a dimension via *semts: segmentDimension*, connected to the corresponding temporal segment and it receives a knowledge node of type *semts: SegmentDataCharacteristic*. The individual value is then attached via semts: hasValue to a *semts: EmbeddedValue* and the *semts: valueString* contains the explicit value.

Thus, one offers a descriptive analysis that is held in the same place with the original data, becoming a part of the same semantic structure, so the results mix both the time observations and derived features, facilitating unitarily reuse. The features sets characterize a univariate part of the series, creating a semantic segment that represents the projection of the complete series onto that dimension. A *semts: KnowledgeGroup* is attached and the feature families are organized under, each representing a *semts: SegmentDataCharacteristic* and caring the determined value via *semts: EmbeddedValue* (Figure 5).

Figure 5: Analytical features integration: feature organizing structure (left) and individual feature with value and provenance method (right)

In the expanded form of the representation, the features provide a wider set of feature families, capturing different properties of the time series. These derived properties are grouped into thematic groups, as follows: stats (statistics), acf (autocorrelation), entropy, frequency, heterogeneity, memory, pacf (partial autocorrelation), data shape, stl (seasonal and trend decomposition) and statistical tests, each bringing an analysis perspective. At the individual level, each family expands; there are basic groups, for example the statics family that carries descriptors such as: mean, median, distribution, missing or null values (Figure 6), but also time series analysis specifics, such as: entropy, seasonality or trend.

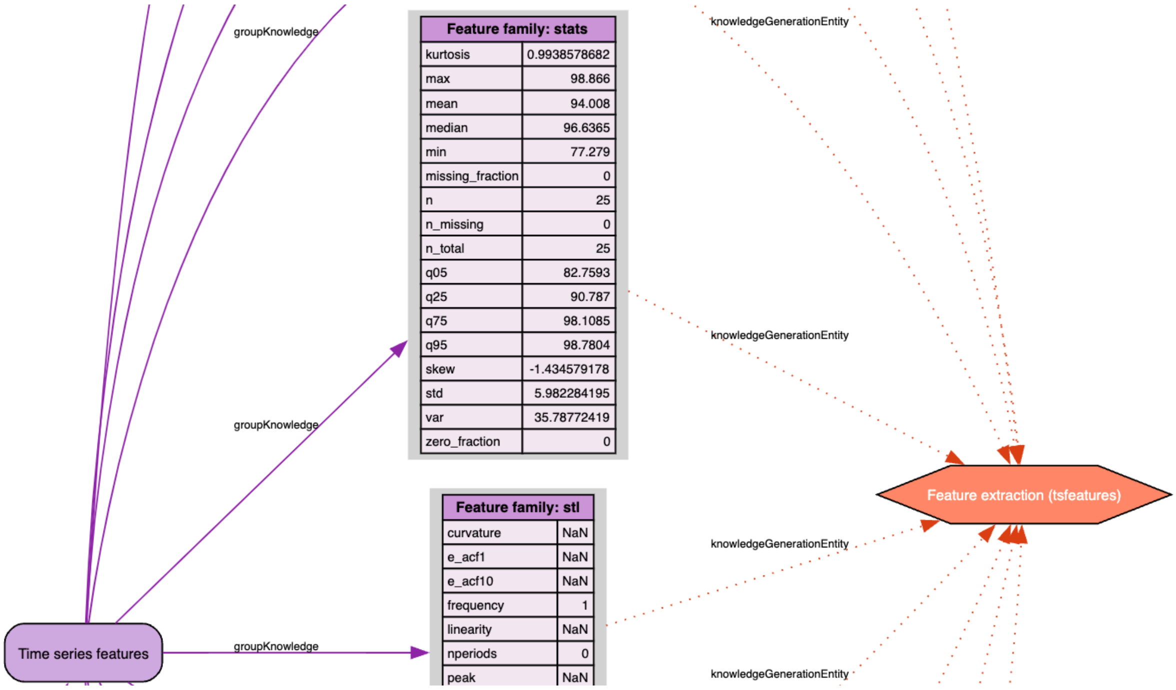

Figure 6: Visual representation of a part of a knowledge graph resulted from the time series analysis

Representing the time series in JSON-LD format allows the generation of a semantical knowledge graph, in which relationships between the entity, time segments, dimensions, observations and features are visually highlighted. This approach enables a visual manner to explore the structure of the encoded knowledge (Figure 6).

This logic is not specific to the example of time series chosen for the scope of this paper, and the tool is only instantiated with a custom configuration, according to the data columns. The same procedure could be applied to any temporal dataset. In the resulting form, the time series is no longer just a collection of organized values, but it develops into a group of semantic artefacts in which data and derived knowledge reside together.

5. Conclusion

The proposed solution was designed to transform open time series provided as tabular representation into an enhanced form of semantic knowledge. The approach originates from the idea that simple public access to data is not enough to support knowledge distribution, but one needs an organized structure that preserves the relationships between observed entities, time indexes, measured dimensions and descriptive features derived from observations analysis. This linked structure approach allows attaching knowledge to data, meaning the analytical metadata to specify time intervals or dimensions.

The paper contribution includes a mapping process that iterates multiple steps: read and detect the structure of the input data, identify semantic meaning of the columns (entity, time, dimensions), standardize values, construct identifiers, model the series contents and its time segments according to the ontology specifications in a hierarchical structure. The ontology does not act as a vocabulary labelling unit, and it is conceived as a reproductible scheme for organizing the knowledge. SemTS was used as a conceptual scheme for representing the time series data, instantiating elements that give them meaning. By encoding observation, features and provenance into this schema, additional knowledge is then obtained by querying, comparing and visualizing the resulted semantic graph.

The paper included a case study based on open data for sustainability development and showed a fragment of the knowledge graph extracted from this analysis. This approach allows a universal form of representation, ready for integration. From a KM perspective, the specificity of the solution lays in the fact that it turns data into knowledge assets: knowledge objects (observations, derived characteristics and provenance) with defined relationships, which can be interrogated to gain knowledge insights. The solution urges to support KM concerns like knowledge organization and explanation, context preservation, addressing the availability open data insight.

Ethical Declaration: Ethical clearance was not required for the development of this research. The data analysis was conducted exclusively on publicly available online resources.

AI Declaration: AI assistance tools were used in the research process to enhance the search for suitable literature, to check grammar and spelling, and to review parts of the code.

References

Damagatla, S.V.K., Bossi, A., Bargigia, I. and Pifferi, A. (2025) "A Data Management Plan for Open Data, readable and reusable by humans and AI", European Conferences on Biomedical Optics 2025, Technical Digest Series, Optica Publishing Group, Munchen, Germany. https://doi.org/10.1364/ECBO.2025.Tu2A.27

Davies, T., Walker, S., Rubinstein, M. and Perini, F. (2019) "The State of Open Data: Histories and Horizons", African Minds and the International Development Research Centre (IDRC), Cape Town, South Africa, pp. 12-17. https://doi.org/10.5281/zenodo.2668475

Grass, A., Beecks, C., Chala, S.A., Lange, C., Decker, S. (2023) "A Knowledge Graph for Query-Induced Analyses of Hierarchically Structured Time Series Information", New Trends in Database and Information Systems. ADBIS 2023. Communications in Computer and Information Science, Cham: Springer, Vol 1850, pp. 174-184. https://doi.org/10.1007/978-3-031-42941-5_16

Grass, A., Beecks, C., Lange-Bever, C., Collarana, D., Deshmukh, R.A., Decker, S. (2024) "The Semantic Time Series Ontology", [online], https://github.com/semts-ontology/SemTS.

Grass, A., Deshmukh, R.A., Beecks, C., Decker, S. (2025) "Towards an Ontology for Representing Time Series Knowledge: Motivation, Requirements and Concept", 37th International Conference on Advanced Information Systems Engineering (CAiSE 2025), Spinger Nature Switzerland, pp. 103-110. https://doi.org/10.1007/978-3-031-94590-8_13

Hahnel, M., Smith, G., Toogood, M., Jarvis, P., Emery, C. (2026) "The State of Open Data 2025: A Decade of Progress and Challenges (Version 6)", Digital Science, Springer Nature, Team Figshare. https://doi.org/10.6084/m9.figshare.30823079.v6

Hyndman, R.J., Athanasopoulos, G., Garza, A., Challu, C., Mergenthaler, M. and Olivares, K.G. (2025) "Forecasting: Principles and Practice, the Pythonic Way", Melbourne: OTexts, [online], https://otexts.com/fpppy

Kamalruzaman, M.S., Mohd, N. and Maaz, Z. (2025) "Mapping the Landscape of Knowledge Management Research: 2020–2025", International Journal of Research and Innovation in Social Science, Vol 10, pp. 1194-1205. https://dx.doi.org/10.47772/IJRISS.2025.90700098

Kawtar, Y.D., Hind, L. and Dalila, C. (2022) "Ontology-based Knowledge Representation for Open Government Data", International Journal of Intelligent Systems and Applications in Engineering, Vol 10, No. 4, pp. 761-766.

Mohammad, Y., Nachouki, M., Mohamed, E.A. (2025) "Knowledge Management Systems in Business Management Using Knowledge Graphs and Semantic Technologies", International Journal ok Knowledge Management, Vol 21, No. 1, pp. 1-29. https://doi.org/10.4018/IJKM.369121

Moreira, J., Bouter, C., Daniele, L., Peixoto, M. and Machado, M. (2024) "Knowledge Representation of Time Series Data: A Comparison Analysis of Standardized Ontologies", 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2024), Amsterdam, The Netherlands.

Ortiz-de-Urbina-Criado, M., Abella, A. and Garcia-Luna, D. (2023) "Open data-set identifier for open innovation and knowledge management", Journal of Knowledge Management, Vol 27, No. 10, pp. 2779–2796. https://doi.org/10.1108/JKM-07-2022-0514

Peng, C., Xia, F., Naseriparsa, M. and Osborne, F. (2023) "Knowledge Graphs: Opportunities and Challenges", arXiv preprint arXiv:2303.13948. https://arxiv.org/abs/2303.13948

Rowley, J. (2007). "The wisdom hierarchy: representations of the DIKW hierarchy", Journal of Information Science, Vol 33, No. 2, pp. 163-180. https://doi.org/10.1177/0165551506070706

Sachs, J.D., Lafortune, G., Fuller, G., Iablonovski, G. (2025) "Financing Sustainable Development to 2030 and Mid-Century. Sustainable Development Report 2025" – Full Database [dataset][online], Paris: SDSN, Dublin: Dublin University Press, https://dashboards.sdgindex.org/downloads/. https://doi.org/10.25546/111909