ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XII-4/W2-2026
10th International Conference on Smart Data and Smart Cities (SDSC), 29 September–1 October 2026, Sofia, Bulgaria

Bridging the Physical and Virtual: Automated Thematic Classification of IoT Sensors for Urban Digital Twin Catalogs

Jeffrey Limnardy[1]*, Joseph Gitahi[1], Thomas H. Kolbe[1]

[1] Chair of Geoinformatics, Technical University of Munich, 80333 Munich, Germany
(jeffrey.limnardy,joseph.gitahi, thomas.kolbe)@tum.de

Keywords: IoT, Metadata Catalogs, Urban Digital Twins, Automation, Metadata Management, Framework.

Abstract

Urban Digital Twins (UDTs) rely on IoT sensors to bridge physical and virtual domains, but deployments with thousands of sensors face significant challenges in device information discoverability. Metadata catalogs address this by registering and describing sensors with standardized metadata, enabling stakeholders to efficiently discover and access available sensing services. However, current sensor registration to metadata catalogs relies on time-consuming, error-prone manual processes, creating stale entries when services change over time. Overcoming these manual registration challenges requires automated approaches that can maintain high-quality metadata and organize sensors into effective groups for enhanced discoverability. This paper introduces the WRENCH framework, a modular end-to-end framework for sensor device registration. The framework enables automated IoT sensor registration into metadata catalogs using Harvesters, Groupers, MetadataEnrichers, and Catalogers as key components. It provides scheduling and state management for continuous data harvesting and utilizes large language models to generate descriptive titles, improving the discoverability of catalog entries. Our framework features a novel clustering algorithm for IoT devices. KINETIC is implemented as a Grouper within the framework and utilizes various natural language processing techniques, including keyword extraction, co-occurrence networks, and community detection, to cluster sensor information documents. Unlike traditional methods, such as LDA (Latent Dirichlet Allocation), that fail with heterogeneous sensor data, KINETIC identifies underrepresented clusters while maintaining semantic coherence, demonstrating superior performance. The framework addresses central challenges in urban IoT metadata management and facilitates the growing utilization of digital twinning technologies through continuous and automated data exploration and registration.

1. Introduction

Urban Digital Twins (UDTs) provide virtual models of physical environments by integrating diverse data sources, including 3D city models, sensors, and simulation data, significantly improving urban planning and decision-making across domains such as environmental monitoring, transportation, and energy management. With 85 percent of IoT platforms expected to incorporate some form of digital twinning by 2025 (Hakiri et al., 2024), the importance of robust frameworks for managing diverse IoT data sources continues to grow.

Internet of Things (IoT) serves as a key enabler for UDTs, bridging the physical and virtual realms through identification, modeling, connectivity, and cyber-physicality. The data obtained from tangible items, sensors, IoT devices, and various origins form the fundamental basis of digital twins, providing details about the present condition, actions, and efficacy of the entity.

However, IoT data discoverability remains one of the most critical yet underaddressed challenges in digital transformation. IoT generates continuous, heterogeneous streams from distributed devices requiring contextual metadata that traditional data catalogs cannot effectively manage (Zhang et al., 2021). Sensor readings are meaningless without metadata describing sensor type, measurement unit, and environmental context (Milenkovic, 2015), yet organizations frequently collect large amounts of data they are unable to utilize.

Manual metadata management is the most significant limitation in current approaches. Existing tools such as Apache Atlas, AWS Glue Data Catalog, and Azure Purview require human intervention for metadata creation, tagging, and classification, a process that becomes not feasible where thousands of devices are continuously added. This bottleneck prevents timely data discovery and shows the need for automated solutions leveraging AI and graph-based modeling (Ma et al., 2013).

1.1 IoT Data Sources

There is a wide range of IoT data sources available in urban environments. Common examples span multiple domains, including weather and air quality sensors in the environmental domain, traffic and mobility sensors in the transportation domain, and energy consumption and production sensors in the energy domain, among many others. These data sources are often heterogeneous, with different data models, formats, and protocols. This heterogeneity poses significant challenges for integrating these diverse data sources into a unified metadata catalog.

Standards such as the OGC SensorThings API and FIWARE Context Broker/Smart Data Models provide structured approaches to representing IoT data, aiming to improve interoperability across diverse data sources.

1.2 Metadata Catalogs

Urban metadata catalogs provide a way to organize and document diverse datasets needed for UDTs through comprehensive metadata management, supporting temporal and spatial filtering, and relationship management between datasets. Several

* Corresponding author

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XII-4-W2-2026-113-2026

catalog platforms exist in the urban data domain. Proprietary solutions such as Socrata[1] offer cloud-based data publishing for city governments, though their closed-source nature limits extensibility. However, neither general-purpose open data catalogs nor catalogs used in Spatial Data Infrastructures (SDIs) are tailored for managing UDTs. The Smart District Data Infrastructure (SDDI) metadata catalog addresses this gap by extending CKAN with predefined categories (e.g., "Online Service", "Device", "Project") and urban topics (e.g., "Mobility", "Energy", "Environment") to improve discoverability, as well as supporting typed relationships between catalog entries (Knezevic et al., 2022).

1.3 Problem Statement

Metadata catalog maintainers currently rely on manual registration processes to register sensor data into the catalog. These processes are not only time-consuming and error-prone, but also require constant maintenance to keep catalog entries up to date when services evolve. Catalog maintainers must systematically explore entire service endpoints to identify sensor groups and extract relevant metadata.

Additionally, automatically identifying and grouping related devices within extensive sensor services presents a significant challenge. Urban IoT services often contain hundreds or thousands of individual sensors that must be organized into meaningful clusters based on function, location, or measured parameters. Without proper device clustering, metadata catalogs become cluttered with individual sensor entries, making discovery difficult.

This paper presents a method that automates sensor device registration into urban metadata catalogs by continuously harvesting IoT services (such as OGC SensorThings API) and clustering devices into meaningful groups enriched with descriptive and extracted metadata (titles, descriptions, spatial and temporal extent). A proof-of-concept implementation using sensor data from existing city-managed IoT platforms demonstrates the practical application of this approach.

2. WRENCH Framework

The WRENCH Framework automates the task of registering sensor services into a metadata catalog while leveraging its advantages in terms of metadata discoverability. It is a toolkit for building automated continuous data processing pipelines customized to process sensor metadata.

The framework follows a general data flow as depicted in Figure 1. The `Harvester` retrieves data from an IoT data source, such as a SensorThings API endpoint, a weather data provider endpoint, or any other sources, and creates a collection of all devices. These devices follow two paths: they are passed directly to the `MetadataEnricher` and sent to the `Grouper` for thematic clustering.

The `Grouper` runs a thematic clustering algorithm to group similar devices upon receiving the devices from the `Harvester`. These groups are then forwarded to the `MetadataEnricher`, which processes both the entire device collection received from the `Harvester` and the clustered groups from the `Grouper` to calculate both the service-level metadata and group-level metadata. At the last step, the `Cataloger` receives both metadata types and sends the results to the metadata catalog.

Each block in the diagram represents a component which can be used in the framework to build and run registration pipelines. WRENCH provides a generic pipeline out of the box which follows the data flow depicted in the diagram. This architecture generates service-wide and granular group-specific metadata, enriching the metadata catalog with various data organization and discovery levels.

2.1 Data Model

To align and facilitate extensible interfaces, WRENCH uses the data model shown in Figure 2 to ensure compatibility between components within the framework.

The few classes established in WRENCH represent the framework conceptually. The framework defines three core classes: Device, Group, and CommonMetadata. A Device represents a "Thing" instance which may contain any number of physical or virtual sensors. A Group represents a collection of devices clustered according to specific criteria. A device can belong to multiple groups if it thematically fits into them. Lastly, CommonMetadata stores metadata information for device collections at two scopes: service-level metadata encompassing the entire service, or group-level metadata specific to individual groups.

2.2 Base Classes

WRENCH provides several abstract base classes. These abstract base classes serve as a contract for implementations, enabling different implementations to be easily interchangeable. There are four main abstract base classes: `BaseHarvester`, `BaseGrouper`, `BaseMetadataEnricher`, and `BaseCataloger`.

Pipelines in WRENCH compose these base classes as modular building blocks. This architecture enables users to substitute individual components while maintaining the rest of the pipeline unchanged. For example, replacing the `BaseHarvester` implementation to extract data from different sources. The same modularity applies to all components, facilitating extensibility and customization for diverse use cases.

2.3 Components

Components are building blocks of a registration pipeline. All components must implement the `Component` interface consisting of a `run` method, which will be invoked by the pipeline. WRENCH provides several base components necessary for a registration pipeline, and these base components can be inherited to create implementation or data source specific components. Additionally, custom components can also be created and used to compose a pipeline.

2.4 Scheduling and State Management

WRENCH supports scheduled pipeline execution via `IntervalScheduler`, which reruns pipelines after a fixed duration, and `CronScheduler`, which accepts a cron expression for more fine-grained scheduling.

Pipelines have a `PipelineStateManager` which tracks component states and versions. The manager initializes states for

[1] `https://dev.socrata.com/`

https://doi.org/10.5194/isprs-annals-XII-4-W2-2026-113-2026

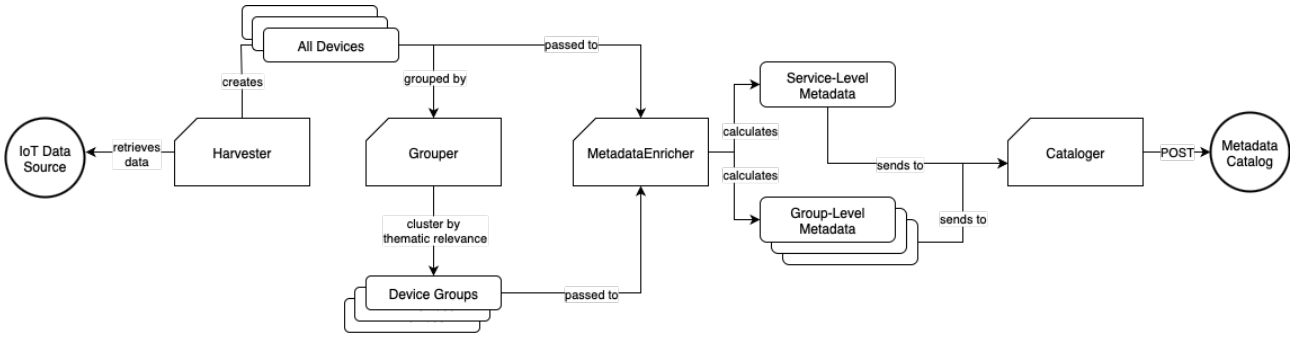

Figure 1. Framework Data Flow

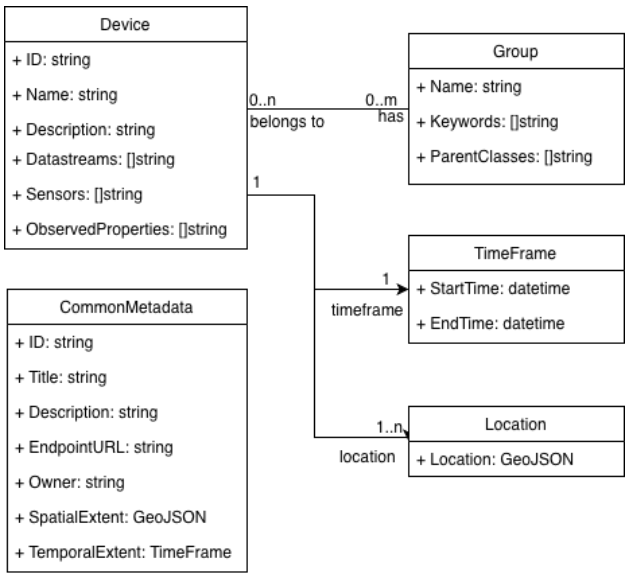

Figure 2. Data Model

new components or uses the latest existing state for comparison with current states. Component states are committed only upon successful pipeline completion or early termination. Failed pipeline runs result in complete state rollback. This transaction-like mechanism prevents state inconsistencies between different components.

2.5 State Persistence

Persistent state management is handled through the `ResultStore` abstraction, with two implementations: `FileStore`, which persists pipeline states as JSON files for fault tolerance, and `InMemoryStore`, which stores states in memory for faster but non-persistent development use.

3. KINETIC Clustering Algorithm

Part of the registration process is creating device groups that reflect the contents of each data source, improving their visibility and discoverability in the metadata catalog. WRENCH's Grouper component clusters devices into thematic groups closely following the 15 SDDI catalog topics shown in Figure 3, excluding "Information Technology". These groups enable users to search and filter data by thematic interest.

Cities collecting IoT data typically aim to publish sensor data to enable the development of Urban Digital Twins. The sensors

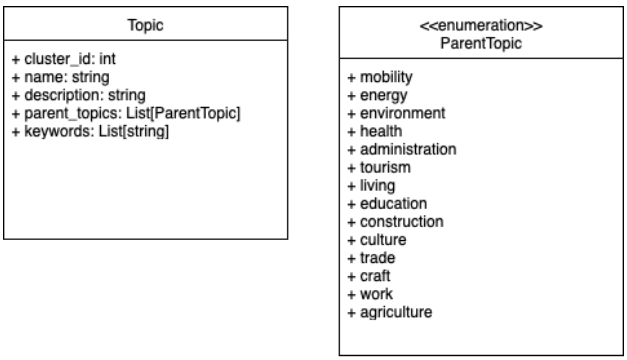

Figure 3. The Topic data model for LLM Structured Output

available in these services will likely be related to one of the topics mentioned in the list above. It is also important to note that this IoT data may come from physical devices or simulations.

Various methods can be used to group sensor devices. These include the conventional LDA (Latent Dirichlet Allocation) (Blei et al., 2003), newer NTM (Neural Topic Modeling) models like BERTopic (Grootendorst, 2022), or even graph-based topic modeling like CWUTM (Topic model based on co-occurrence word networks for unbalanced short text datasets) (Ma et al., 2023).

Sensor metadata in IoT deployments often exhibit imbalanced topic distributions, with large dominant clusters alongside small underrepresented ones — for instance, the Hamburg IoT service contains nearly 1,000 EV charging station devices but only a single renewable energy monitoring device. Clustering such skewed distributions is challenging for conventional methods: LDA struggles with short texts (Mazarura and de Waal, 2016, Zou and Song, 2016), while both LDA and BERTopic suffer from a hyperparameter trade-off where settings optimized for major topics neglect minority clusters and vice versa.

3.1 Methodology

KINETIC is a graph-based topic model that leverages keyword co-occurrence networks and community detection to partition the co-occurrence network into communities or clusters. It uses a mix of Sentence Transformer embeddings (Reimers and Gurevych, 2019) and term frequency to calculate similarities between documents and clusters. These classified clusters are then passed on to a large language model, which generates appropriate names for each cluster. The full process is depicted in Figure 4

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XII-4-W2-2026-113-2026

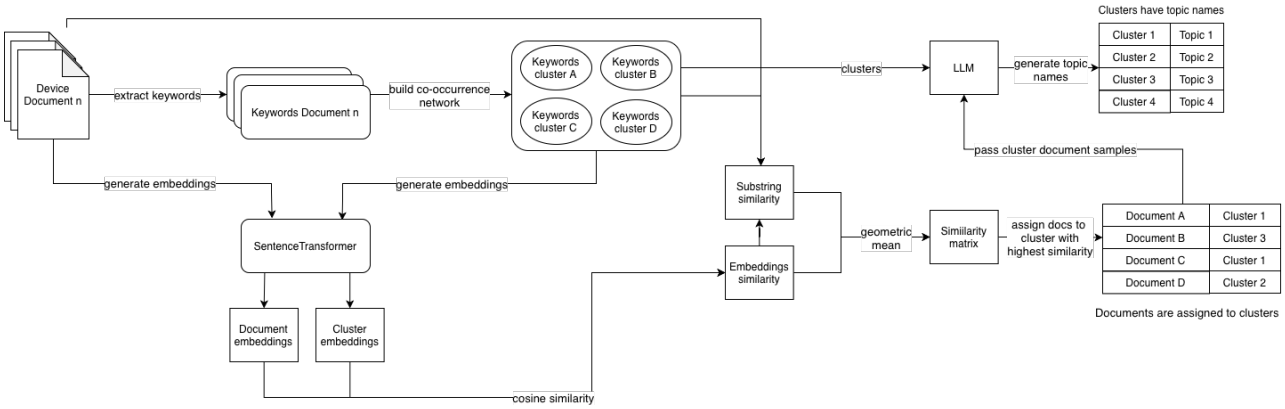

Figure 4. Our Clustering Methodology

Keyword Extraction The first step of our clustering method involves extracting keywords from the device documents. Device documents are stringified Device data models with selected fields (Name, Description, and Datastreams). These documents are passed to a keyword extraction method such as KeyBERT (Grootendorst, 2020) or YAKE! (Campos et al., 2020).

The emergent keywords form the foundation for thematic clustering, because they reflect the semantic meaning of each device. Keywords should appropriately capture the diverse domains in the IoT service to create meaningful clusters.

Keyword Clustering After extracting keywords from each document, these are clustered to form keyword groups representing topics. The clustering method involves constructing and partitioning a keyword co-occurrence network into different communities or clusters. Keywords are represented as network nodes and their co-occurrences as weighted edges.

Two keywords co-occur when they appear in the same document. Edge weights correspond to co-occurrence count, with higher weights indicating frequent co-occurrences. The co-occurrence network can be partitioned using community detection methods such as the Louvain method, which is widely used to detect non-overlapping community structures of large networks using modularity optimization (Blondel et al., 2008). Furthermore, the Louvain method can be configured to consider weighted edges, utilizing the co-occurrence frequency information to build communities within the network. The method also takes a resolution parameter, referring to the time described in (Lambiotte et al., 2014), which can control the size of detected communities. Higher values favor smaller communities, while lower values produce larger communities.

The detected keyword communities represent the topics available in a sensor service. These communities provide a foundation for thematic organization but require further processing to enable device assignment. To achieve this, each community must be represented in a continuous embedding space where semantic relationships can be quantified and devices can be systematically assigned to their most relevant thematic groups.

Embedding Generation Embeddings are generated for the keyword clusters and device documents to semantically classify devices into keyword clusters. They must be embedded using the same model to ensure consistent representation within the same embedding space.

The `multilingual-e5-large-instruct` model is selected for its strong multilingual performance and low memory footprint (Wang et al., 2024). Query prompts guide the embedding generation to provide context for documents and clusters.

Similarity Calculation and Classification The classification of devices into the formed keyword clusters uses similarity calculation. This calculation considers two different similarity parameters, semantic and substring similarity. The generated embeddings for device documents and keyword clusters are used to measure conceptual relatedness, and the similarity between these embeddings can be calculated using cosine similarity, providing a similarity score. The cosine similarity between their embedding vectors $\mathbf{d}_i, \mathbf{c}_j \in R^d$ is defined as:

$$S_{semantic}(i,j) = \frac{\mathbf{d}_i \cdot \mathbf{c}_j}{\|\mathbf{d}_i\|\|\mathbf{c}_j\|} \tag{1}$$

where $\mathbf{d}_i$ represents the embedding vector for the i-th document, and $\mathbf{c}_j$ represents the embedding vector for the j-th cluster.

Secondly, counts of cluster keyword occurrences represent substring similarity between a document and a cluster. The substring occurrences are normalized by the keyword count of each cluster to keep similarity scores comparable across different cluster sizes. The substring similarity between document i and cluster j is defined as:

$$S_{\text{substring}}(i,j) = \frac{1}{|K_j|} \sum_{w \in K_j} \text{count}(w, d_i) \tag{2}$$

where K_j represents the set of keywords for cluster j, $|K_j|$ denotes the number of keywords in cluster j, and count (w, d_i) represents the frequency of keyword w in document d_i.

The geometric mean of both substring and semantic similarity is then calculated to produce a unified similarity score. This dual approach balances semantic understanding and explicit keyword matching. The combined similarity between document i and cluster j is defined as:

$$S_{\text{combined}}(i,j) = \sqrt{\frac{S_{\text{semantic}}(i,j)^2 + S_{\text{substring}}(i,j)^2}{2}} \tag{3}$$

where $S_{\text{semantic}}(i, j)$ represents the semantic similarity from Equation 1, and $S_{\text{substring}}(i, j)$ represents the substring similarity from Equation 2.

LLM Topic Generation Large language models (LLMs) assist our clustering process by creating appropriate names for each cluster. These named clusters are stored in the Topic data model. Passing the data model enforces the LLM to output a structured data set with expected values.

The Topic data model provides several essential fields. The `cluster_id` identifies which cluster this topic corresponds to, `name` and `description` are both used to name and describe the clusters. `ParentTopic` is an enumeration of the SDDI-defined thematic groups as described in Section 3. These groups guide the creation of the topic names and descriptions, while enforcing the groups to at least be related to one of the SDDI-defined thematic groups so that they can be appropriately assigned to their thematic group when registering into the SDDI catalog (Knezevic et al., 2022). The "Information Technology" category is excluded due to the LLM's tendency to consistently classify all sensor data into this category, regardless of the actual data type or domain.

To generate comprehensive and representative topic names and descriptions, the cluster keywords are taken, along with some sample data that belongs to the cluster. Representative sample data are selected by choosing devices with unique datastream descriptions, since similar devices share the same datastream characteristics. Since the classification of devices into clusters happens before the topic generation with LLM, these classified devices can be used as sample data. This information is injected into the user prompt and used in the generation.

The model `llama3.3:70b-instruct-q4_K_M` is used for testing as it has been the best available model within our internal Ollama instance at the time of development and has a context length of 128k tokens. It is based on the multilingual Llama 3.3 model with 70b parameters, pretrained and fine-tuned for following instructions with 4-bit quantization for computational efficiency (Grattafiori et al., 2024).

This final step in the clustering flow produces the final output of the KINETIC grouper, a set of topics, with name, description, and representative keywords, along with devices classified to each topic.

Cache Since LLM-generated topic names are non-deterministic, repeated runs risk producing inconsistent or duplicate cluster names. To address this, KINETIC maintains a cache of clusters, their embeddings, and generated topics from previous runs, ensuring consistency while also reducing overhead. The cache also allows manual intervention, enabling human review and improvement of unsatisfactory clusters. It is only invalidated when a significant number of new outliers are detected, triggering a full cluster regeneration.

4. Experiments

To determine how WRENCH performs compared to manual registration, qualitative and quantitative evaluations are performed to assess the automated registration pipeline's efficiency, performance, and quality.

4.1 Data Sources

The evaluation data used comes from the OGC SensorThings API services provided by Hamburg, Osnabrück, and Munich. These servers enable urban digital twin and smart city. Hence, the IoT data in these platforms are heterogeneous and span multiple domains. The data are primarily in German with some English content. Since the data within these servers continuously changes and evolves, the following exploration reflects the server state at the time of writing.

Hamburg IoT Server The Hamburg IoT server[2] has 4953 things. The devices in the server mainly come from the mobility domain. It also has a single Thing from the energy domain measuring the production and usage of renewable energy infrastructure. The distribution of the Things by device type is displayed in the pie chart in Figure 5.

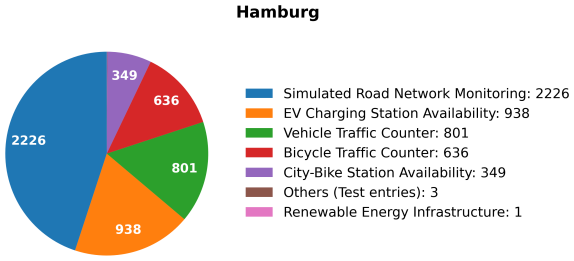

Figure 5. Thing distribution in Hamburg IoT server by category

Osnabrück IoT Server The Osnabrück IoT server[3] has 478 Things. The server is more heterogeneous than the Hamburg IoT server and spans the energy, environment, and mobility domains.

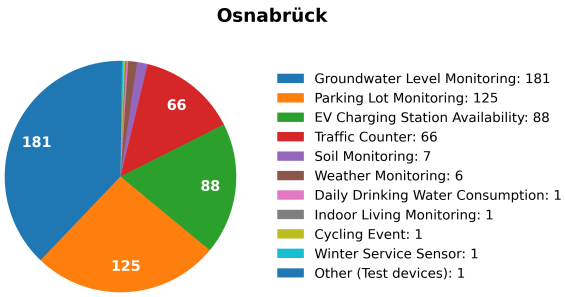

Figure 6. Thing distribution in Osnabrück IoT server by category

The distribution of Things in the Osnabrück IoT server is even more fragmented than the Hamburg IoT server. It contains individual categories with a single Thing, such as "Indoor Living Monitoring" and "Winter Service Sensor", while also having categories with a high number of Things, like "Groundwater Level Monitoring" and "Traffic Counter".

Munich IoT Server The Munich IoT server has 1397 Things as of 23 July 2025. The server primarily focuses on the mobility and environment domain, featuring diverse sensor devices. Since the Chair of Geoinformatics at TUM maintains the server, the data is better organized and more consistently structured than other data sources that rely on multiple contributors or less centralized management.

[2] `https://iot.hamburg.de/v1.1`
[3] `https://daten-api.osnabrueck.de/v1.1`

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XII-4-W2-2026-113-2026 | © Author(s) 2026. CC BY 4.0 License.

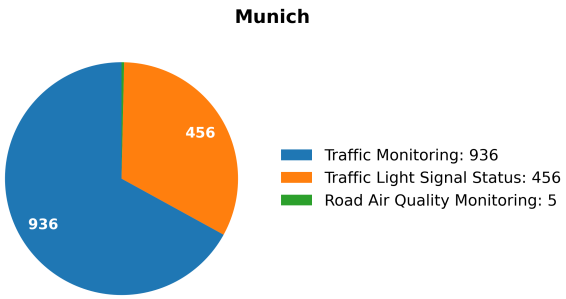

Figure 7. Thing distribution in Munich IoT server by category

Figure 7 shows the distribution of sensors by category in the Munich IoT server. The server has fewer sensor categories compared to other IoT servers observed in this paper. However, like the Hamburg and Osnabrück servers, Munich also shows an uneven distribution of categories.

4.2 Clustering Requirements

The clustering requirements derive from the source data. The data exploration shows that the number of devices for a single device category can be highly skewed; some categories may include thousands of devices, while others may have fewer than 10 devices. The clustering method must be capable of identifying different clusters of varying sizes.

Data varies across different IoT servers. The source data introduced in the previous subsection maintains data in various domains. The Hamburg IoT server focuses more on mobility data, while the Osnabrück IoT server contains data from a more diverse set of domains. These server-specific categories are latent and must be discovered by the clustering method.

Things within the same category are typically registered using shared text templates. For example, Hamburg's bicycle counters follow a *Zählfeld* naming pattern, while city-bike stations use *Stadtrad-Station* as a prefix. This lexical regularity across devices of the same type forms the foundation for keyword extraction-based clustering.

Only Thing and Datastream entities are used for clustering. Sensors and ObservedProperties are excluded as they may incorrectly link unrelated clusters, while Observations and Locations carry no semantic value for grouping purposes. This selective approach takes advantage of the template-derived lexical similarities in device names and descriptions while avoiding the clustering interference caused by shared technical components or semantically irrelevant data, ensuring accurate identification of functionally related device groups.

4.3 Clustering Accuracy

We evaluate clustering accuracy using NMI (Normalized Mutual Information), Homogeneity, Completeness, and V-Measure. Homogeneity measures whether each cluster contains only data points from a single class, while Completeness measures whether all data points of a given class belong to the same cluster. The two are often inversely related. V-Measure combines both as their harmonic mean, similar to how F-measure combines precision and recall.

The resulting scores for evaluation of classification performance in table 1 show that KINETIC performs best for both Hamburg and Osnabrück datasets, achieving near-perfect V-Measure results for the Hamburg dataset and a minor decline in accuracy for the Osnabrück dataset.

Method	NMI	Hom.	Comp.	V-Meas.
LDA Hamburg	0.999	0.998	0.999	0.999
LDA Osnabrück	0.820	0.939	0.727	0.820
LDA Munich	0.896	1.000	0.812	0.896
BERTopic Hamburg	0.907	0.971	0.850	0.907
BERTopic Osnabrück	0.674	0.854	0.556	0.674
BERTopic Munich	0.774	0.966	0.645	0.774
KINETIC Hamburg	**0.999**	**0.999**	**1.0**	**0.999**
KINETIC Osnabrück	**0.940**	**0.934**	**0.946**	**0.940**
KINETIC Munich	**1.0**	**1.0**	**1.0**	**1.0**

Table 1. Classification Performance Scores for Topic Modeling Methods

LDA method also performs nearly on par with KINETIC for the Hamburg and Munich datasets. However, its performance declines by a noticeable margin on the Osnabrück dataset. This suggests that topic diversity and cluster imbalance have a heavier impact on classical topic modeling methods such as LDA. The Osnabrück dataset, in particular, contains more minority clusters, which are more challenging to detect.

BERTopic performs the worst out of the three explored methods, although it follows a similar performance trend across the Hamburg and Osnabrück datasets. Its performance declines even more sharply than LDA on the Osnabrück dataset.

The three methods present a trade-off between topic modeling accuracy and computational efficiency. KINETIC offers higher accuracy at the cost of increased runtime. In contrast, LDA offers faster processing but is more susceptible to higher performance degradation when applied to an imbalanced dataset or one containing minority clusters.

4.4 Framework Performance

Processing time measurements were conducted on an Apple Silicon M1 processor and an 8 GB RAM system. To reduce the effects of system-level noise, e.g., background processes consuming CPU or memory resources, results are averaged over three runs. The components used in evaluating WRENCH's performance are: SensorThingsHarvester (Harvester), KINETIC (Grouper), SensorThingsMetadataEnricher (MetadataEnricher), and SDDICataloger (Cataloger).

Table 2 presents the runtime and the peak memory usage of each component in the framework. As expected, the KINETIC grouper is the most compute-intensive component, since it performs several compute-intensive calculations for embeddings generation, similarity calculation, and keyword extraction. These tasks are typically more efficient when executed on a GPU, which the test machine does not have, resulting in subpar performance. The Hamburg dataset's processing time is 1813.4 seconds (approximately 30 minutes) for 4953 items. However, using its cache can significantly improve this, reducing the processing time by over 1000 seconds (around 16 minutes).

Figure 8 illustrates how each framework component's processing time grows with increasing dataset size. The Grouper scales the most with higher dataset size, with processing time increasing exponentially from 211.7 to 1631.8 seconds for Osnabrück and Hamburg datasets respectively. The Harvester shows more linear scaling, increasing from 11.9 to 86.2 seconds

https://doi.org/10.5194/isprs-annals-XII-4-W2-2026-113-2026

Component	Dataset	Memory	Time (s)
Harvester	Hamburg (4,953)	514 MB	86.2
	Osnabrück (478)	473 MB	11.9
	Munich (1,397)	494 MB	30.5
Grouper	Hamburg	850 MB	1631.8
	Osnabrück	468 MB	211.7
	Munich	513 MB	991.0
Enricher	Hamburg	812 MB	59.6
	Osnabrück	493 MB	89.2
	Munich	527 MB	23.7
Cataloger	Hamburg	727 MB	19.8
	Osnabrück	488 MB	22.1
	Munich	497 MB	5.4
Total	Hamburg	917 MB	1813.4
	Osnabrück	534 MB	337.3
	Munich	595 MB	1121.3

Table 2. Component Performance by Dataset

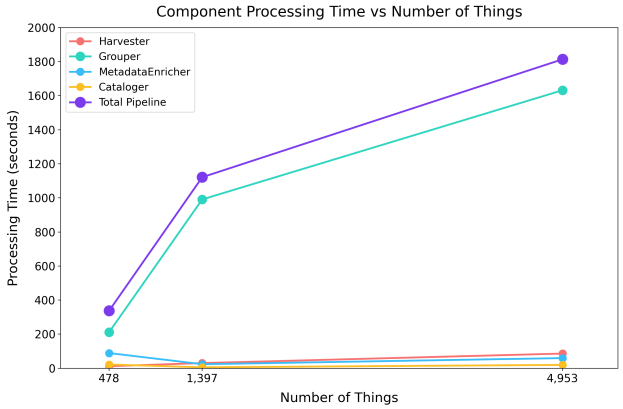

Figure 8. Component Processing Time Comparison to Dataset Size

across the same range. This performance difference reflects the complexity of each component's operations. While the Harvester performs straightforward data collection and the MetadataEnricher processes aggregated metadata, the Grouper must analyze semantic relationships between thousands of individual device descriptions, making it the primary bottleneck in the pipeline. The Total Pipeline performance closely follows the Grouper's behavior, confirming that clustering operations dominate the overall processing time.

4.5 Quality of LLM-Generated Entries

The quality of device group entries must be evaluated for completeness, coherence, and relevance. Both titles and descriptions should address: what devices measure, which server they belong to, device count, sensor types used, and whether all devices in the group are reflected.

Using KINETIC, group names and descriptions were generated for all three datasets. The generated entries demonstrate good overall quality across all datasets. Titles are clear and informative, while descriptions successfully capture device types, measurement parameters, sensor specifications, and data aggregation details. The LLM effectively incorporated relevant keywords that enhance searchability within metadata catalogs.

However, two limitations were observed. First, some devices were assigned to incorrect thematic groups when certain keywords are shared between clusters. For example, the Osnabrück IoT service contains devices named "Parkhaus…", "E-Ladestation Parkhaus…", and "Parkplatz…". While "E-Ladestation Parkhaus" should be classified under EV Charging Stations, as these devices monitor charging infrastructure within parking garages, KINETIC occasionally groups them with Parking Lot Monitoring due to overlapping terminology. Second, generated catalog descriptions lack publisher and ownership information, as this metadata is unavailable from the upstream IoT services and must be supplied manually by data providers.

4.6 Catalog Results

The pipeline generates catalog entries within an SDDI catalog test instance, utilizing built-in features such as main categories and thematic tags. The MetadataEnricher component uses an LLM to generate titles and descriptions during metadata enrichment. Figure 9 shows an example entry tagged with the "mobility" theme. The SDDI catalog's search functionality leverages this generated content to improve discoverability—users searching for related terms (e.g., "e-cargo bikes," "stadtrad," "bike-sharing") will find relevant entries even when these terms only appear in generated descriptions.

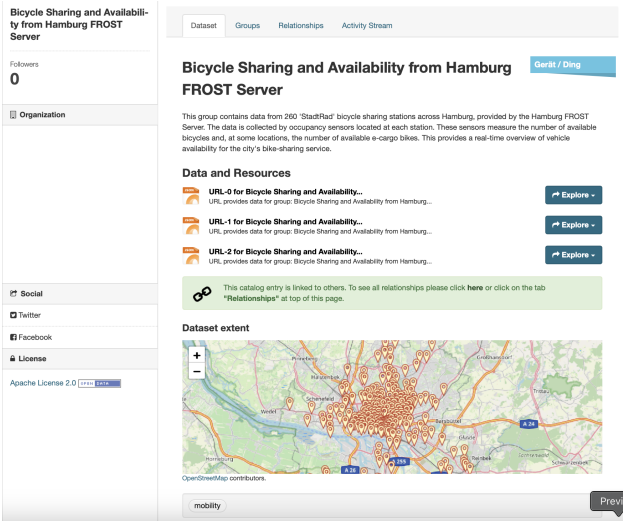

Figure 9. Generated Group Entry in SDDI Catalog from Pipeline Execution

The "Data and Resources" section links directly to filtered SensorThings API endpoints displaying only the Things relevant to each entry, with multiple URLs provided when necessary to avoid exceeding filter length limits. Each entry also contains spatial and temporal metadata. Spatial coverage is shown on an embedded map via a GeoJSON expression of device locations, while the framework calculates validity period start and end dates—if the end date falls within a day of the current date, the dataset is considered active and the field is left empty. Together, these attributes allow users to filter entries by time frame or geographic boundary. Information that cannot be discovered from the data source, such as author, maintainer, language, and version, must be provided by the user.

Relationships between group-level and service-level entries are visualized as a graph in the "Relationship" tab, enabling navigation between entries. Service-level entries such as the Hamburg IoT server typically have many group entries pointing to them,

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XII-4-W2-2026-113-2026

while a group entry such as "Bicycle Sharing" points to a single service-level entry.

5. Conclusion

Automating sensor registration into metadata catalogs while maintaining semantic quality and discoverability presents significant challenges. This paper addresses these challenges by introducing the WRENCH framework, which enables full automation of IoT data registration through a modular architecture of interchangeable components: Harvesters, Groupers, MetadataEnrichers, and Catalogers. This design supports diverse data sources and allows simultaneous registration to multiple metadata catalogs with scheduled updates.

The KINETIC clustering method strengthens the framework by integrating keyword extraction, co-occurrence-based graph analysis, embeddings, and community detection to identify coherent sensor groups in heterogeneous and imbalanced urban IoT datasets. KINETIC outperforms LDA and BERTopic across all metrics, particularly on imbalanced data with minority clusters. Moreover, LLM integration generates descriptive metadata that improves catalog discoverability.

This structured workflow provides practical benefits to stakeholders by reducing the manual effort required to register, maintain, and update metadata for urban IoT sensors in catalogs. For maintainers and data providers, it enables consistent and scalable metadata creation through automated harvesting, grouping, enrichment, and registration. For municipalities, digital twin operators, planners, researchers, and application developers, it improves the discoverability and usability of sensor resources across domains such as mobility, environment, energy, and infrastructure.

The current implementation has limitations. KINETIC requires significantly more computational resources than traditional methods, though this yields superior clustering quality. LLM dependency introduces latency and cost considerations for large-scale deployments. WRENCH currently provides four core components (SensorThingsHarvester, KINETIC, SensorThingsMetadataEnricher, SDDICataloger), limiting immediate applicability to other IoT ecosystems.

Future work should focus on expanding component diversity, optimizing scalability, and validating usability through user studies with domain experts and catalog administrators. Such studies would measure task completion time, metadata quality, and user satisfaction compared to manual processes. To foster community development and broader adoption, our framework is open-sourced at `https://github.com/urbansense/wrench`.

References

Blei, D. M., Ng, A. Y., Jordan, M. I., 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, 993–1022.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E., 2008. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., Jatowt, A., 2020. YAKE! Keyword Extraction from Single Documents using Multiple Local Features. *Information Sciences*, 509, 257-289.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al, 2024. The Llama 3 Herd of Models. doi.org/10.48550/arXiv.2407.21783.

Grootendorst, M., 2020. KeyBERT: Minimal Keyword Extraction with BERT. doi.org/10.5281/zenodo.4461265.

Grootendorst, M., 2022. BERTopic: Neural Topic Modeling with a Class-based TF-IDF Procedure. doi.org/10.48550/arXiv.2203.05794.

Hakiri, A., Gokhale, A., Yahia, S. B., Mellouli, N., 2024. A Comprehensive Survey on Digital Twin for Future Networks and Emerging Internet of Things Industry. *Computer Networks*, 244, 110350.

Knezevic, M., Donaubauer, A., Moshrefzadeh, M., Kolbe, T., 2022. Managing Urban Digital Twins with an Extended Catalog Service. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W3-2022, 119-126.

Lambiotte, R., Delvenne, J.-C., Barahona, M., 2014. Random Walks, Markov Processes and the Multiscale Modular Organization of Complex Networks. *IEEE Transactions on Network Science and Engineering*, 1(2), 76-90.

Ma, C., Du, J., Liang, M., Guan, Z., 2023. Topic Model Based on Co-occurrence Word Networks for Unbalanced Short Text Datasets. doi.org/10.48550/arXiv.2311.02566.

Ma, M., Wang, P., Chu, C.-H., 2013. Data management for internet of things: Challenges, approaches and opportunities. *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 1144–1151.

Mazarura, J., de Waal, A., 2016. A comparison of the performance of latent dirichlet allocation and the dirichlet multinomial mixture model on short text. *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, 1–6.

Milenkovic, M., 2015. A Case for Interoperable IoT Sensor Data and Meta-data Formats: The Internet of Things (Ubiquity symposium). *Ubiquity*, 2015(November). https://doi.org/10.1145/2822643.

Reimers, N., Gurevych, I., 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. K. Inui, J. Jiang, V. Ng, X. Wan (eds), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, 3982–3992.

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., Wei, F., 2024. Multilingual E5 Text Embeddings: A Technical Report. doi.org/10.48550/arXiv.2402.05672.

Zhang, L., Jeong, D., Lee, S., 2021. Data Quality Management in the Internet of Things. *Sensors*, 21(17).

Zou, L., Song, W. W., 2016. Lda-tm: A two-step approach to twitter topic data clustering. *2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 342–347.