Integrating Large Language Models into Decision Support Systems for Water Network Management

1th Filippo Piombino
University of Palermo
Palermo, Italy

2st Antonino Pagano
University of Palermo and CNIT
Palermo, Italy
antonino.pagano@unipa.it

3rd Fabrizio Giuliano
University of Palermo and CNIT
Palermo, Italy
fabrizio.giuliano@unipa.it

4thGabriele Restuccia
University of Palermo and CNIT
Palermo, Italy
gabriele.restuccia@unipa.it

5nd Domenico Garlisi
University of Palermo and CNIT
Palermo, Italy
domenico.garlisi@unipa.it

***Abstract*—The management of Water Distribution Systems (WDS) requires advanced analytical tools capable of processing large volumes of heterogeneous data while supporting operators in complex decision-making processes. This paper proposes an agent-based decision-support framework that integrates Large Language Models (LLMs), graph-based machine learning, and time-series analytics to support anomaly detection and diagnosis in WDS. The architecture is centered on an LLM-driven orchestrator agent capable of interpreting natural-language requests, coordinating analytical modules, and translating model outputs into actionable insights. A Graph Neural Network (GNN) models the spatial dependencies of the network and performs node-level leak detection using graph representations that combine topology with hydraulic measurements. Experiments on a simulated network with 815 nodes and 1125 pipes show strong predictive performance, with R^2 values up to 0.936. The results highlight the potential of agentic AI to enhance the usability and interpretability of advanced analytics in water network operations.**

***Index Terms*—Large Language Models, Water Distribution Systems, Graph Neural Networks, Decision Support Systems, Leak Detection, Intelligent Agents.**

I. Introduction

Water Distribution Systems (WDS) are critical components of modern urban infrastructure. However, rapid urbanization, climate change, and the progressive degradation of aging assets increasingly challenge their reliable operation. One major consequence is Non-Revenue Water (NRW), defined as treated water lost due to leaks, pipe bursts, and operational inefficiencies [1]. Besides its economic impact, NRW can pose environmental and public health risks, as pressure anomalies may allow contaminants to enter the drinking water supply. Therefore, timely detection and localization of anomalies in WDS are essential to ensure service reliability and infrastructure resilience. Modern WDSs are increasingly evolving into cyber-physical systems through the widespread adoption of IoT sensors, enabling continuous monitoring and the collection of large volumes of heterogeneous data [2]. Despite this increased availability of data, operational awareness remains limited. Control room environments still rely on static dashboards, fragmented data sources, and rule-based alarms that often generate false positives and provide limited contextual understanding. Consequently, operators work in environments rich in data but poor in actionable information. Recent advances in machine learning have introduced predictive capabilities for WDS monitoring [2], [3], yet many solutions remain isolated and difficult to interpret. In time-critical situations, this fragmentation increases cognitive load and delays decision-making. To address these limitations, we propose an intelligent decision-support architecture that integrates cognitive reasoning with spatial-temporal learning. The proposed framework is centered on a Large Language Model (LLM) acting as a conversational orchestrator that interprets natural language queries, maintains contextual awareness, and coordinates analytical tasks across specialized modules. To capture the spatial dependencies of water networks, the architecture integrates a Graph Neural Network (GNN), which analyzes network topology together with hydraulic measurements to detect anomalous behaviors and potential leak locations. The LLM then interprets the probabilistic outputs of the model and combines them with time-series analyses to provide concise and actionable insights for system operators.

The main contributions of this paper are threefold:
i) Agentic Decision-Support Framework: We introduce a novel architecture in which a Large Language Model acts as a central orchestrator to manage heterogeneous WDS data streams, reducing operator cognitive load.
ii) Integrated Spatial-Temporal Analytics: We develop a tool-calling mechanism integrating a GNN-based anomaly detection model with time-series analytics to jointly capture network topology and temporal dynamics.
iii) Explainable Infrastructure Intelligence: We demonstrate how a conversational interface connects deep learn-

---

This work was supported by the Fondo Finalizzato alla Ricerca di Ateneo (FFR), University of Palermo, Year 2026, DR Rep. No. 1903/2026.
ISBN 978-3-903176-82-9

ing models with human-in-the-loop operations, translating anomaly probabilities into actionable insights.

II. Related Work and Background

The use of LLMs in critical infrastructures like WDS is an emerging research area with strong potential to enhance human-machine interaction, accessibility to technical data, and operational efficiency. In particular, LLMs are being applied as natural language interfaces to complex hydraulic models and simulators, such as the LLM-EPANET framework, which allows users to interact with sophisticated water supply models via text without requiring deep technical knowledge [4]. Recent studies highlight diverse applications of LLMs in critical infrastructure, including automated analysis of heterogeneous databases [5], explanations for computer vision decisions on damaged infrastructure [6], and optimization of pump scheduling for energy savings [7]. In complex systems like WDS, advanced LLM architectures employ techniques such as Retrieval-Augmented Generation (RAG), controlled execution, and problem decomposition, often within multi-agent frameworks, similar to applications in wastewater treatment [8]. Adapting LLMs to domain-specific tasks can be achieved through fine-tuning, as with WaterGPT [9], but generalist models can also perform effectively when combined with prompt engineering, task decomposition, and interactions with simulators or external knowledge bases. These strategies balance flexibility, operational speed, and task-specific performance.

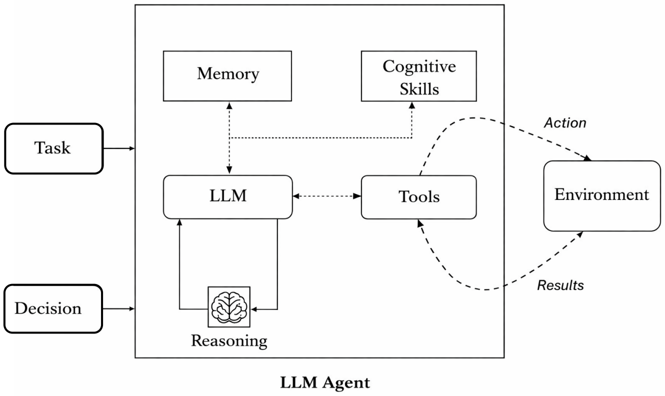


Fig. 1: Architecture of an intelligent virtual agent integrating a Large Language Model with external analytical tools and systems.

Agentic AI refers to systems in which an LLM orchestrates reasoning, memory, and tool use to perform goal-directed tasks. The LLM interprets user requests, determines the necessary operations, leverages external resources, and integrates their outputs, making this approach especially useful in technical settings that require access to diverse data sources and analytical tools. As shown in Fig. 1, an LLM-based agent processes incoming tasks through structured reasoning and contextual memory, then interacts with external tools through an iterative feedback loop. This workflow allows the system to combine natural-language interaction with grounded computational actions, such as querying databases, executing hydraulic analyses, or activating machine-learning models. In practice, agent-based systems are commonly implemented using frameworks that support the orchestration of LLM reasoning and external tools. LangChain is a widely used framework that supports prompt management, conversational memory, and interaction with external data sources or computational modules. LLM-based agents using LangChain often follow the ReAct (Reasoning and Acting) paradigm, alternating between reasoning steps and actions via external tools, incorporating observations from each action into subsequent reasoning, and ultimately producing a final response. This enables the integration of natural-language reasoning with structured interactions in external environments.

III. Analytical Module Design

The analytical module constructs a unified representation of the WDS by combining structural information from the network topology with temporal information from sensor measurements. This integration provides the input to the anomaly-detection model and enables the system to capture both hydraulic connectivity and short-term operational dynamics. The process is organized into three main steps

1) Spatial Topology Extraction (Neo4j Pipeline): the WDS is represented as a directed graph $G = (V, E)$ describing its hydraulic structure. Nodes V correspond to network elements such as junctions, reservoirs, tanks, and control devices, whereas edges E represent the connecting pipes. Each node is associated with features describing static and dynamic properties, including geographic location, pressure, and local demand. Each edge encodes operational attributes such as flow rate, flow velocity, and valve status. This graph structure is retrieved from a Neo4j database that stores the network topology. The extraction process provides the node-index mapping, the edge list, and the structural attributes required to build the graph representation used by the learning module.

2) Time-Series Feature Aggregation (TimescaleDB Pipeline): to complement the topological information, the system retrieves historical and real-time sensor data from a TimescaleDB database. Since these measurements are sampled at high frequency, they are aggregated over fixed temporal windows, such as 24-hour intervals, to obtain compact and stable feature representations. For each monitored variable, the mean, standard deviation, minimum, and maximum are computed within each window. This aggregation step transforms raw time-series data into summary features that retain the main temporal characteristics of the network state while keeping the representation computationally manageable.

3) Snapshot Construction and Labeling: the outputs of the two pipelines are merged into a sequence of graph snapshots, each representing the state of the WDS over a given temporal window. In each snapshot, node and edge features combine structural attributes with aggregated hydraulic measurements, producing a representation suitable for supervised learning. Binary labels are assigned at node level using physical indicators associated with potential water loss, including structural area information, local flow conditions, and nodal demand. Each snapshot is therefore treated as an independent labeled graph instance for anomaly detection. Although the resulting formulation is static at the snapshot level, the aggregated features implicitly encode temporal behavior over the observation window. This design captures relevant short-term dynamics without introducing explicit spatio-temporal graph architectures, thereby reducing computational complexity and supporting fast inference for real-time decision-support.

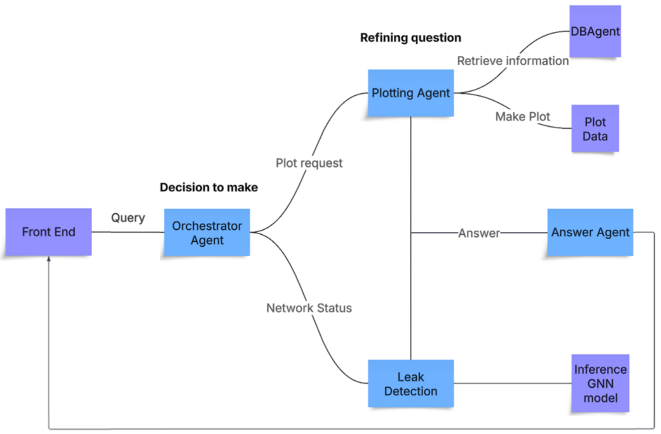


Fig. 2: Logical architecture of the conversational agent for water network management.

IV. ORCHESTRATOR AGENT DESIGN

The proposed architecture is centered on an Orchestrator Agent that coordinates the interaction between the human operator and the analytical components of the system. Its role is to interpret natural-language requests, select the appropriate computational pathway, invoke the required tools, and return responses that combine model outputs with contextual explanations. In this way, the agent provides a unified access layer to the monitoring, analysis, and anomaly-detection capabilities of the WDS platform. The control logic is encapsulated in the `WaterNetworkOrchestratorAgent` class, which manages conversational context through a memory buffer of up to 20 previous exchanges and coordinates the invocation of the main analytical tools. At the architectural level, the agent operates as a hierarchical controller. A user query issued through the front-end interface is first analyzed to identify its operational intent. Requests concerning anomalies, pressure drops, or suspected leaks are routed to the leak-detection workflow, whereas requests involving historical inspection, trend analysis, or system diagnostics are directed to the data-analysis workflow. This separation allows the conversational interface to support distinct analytical tasks while preserving a coherent interaction model for the operator.

A. Tool-Oriented Orchestration

The orchestrator interacts with the analytical modules through a modular tool layer. In the current implementation, the main tools are `execute_leak_detection`, which activates the GNN inference pipeline, and `execute_plotting`, which generates visual analyses of hydraulic and operational variables. This design isolates the reasoning process from the execution of specialized computations, improving modularity and enabling controlled interaction with heterogeneous subsystems. The complete workflow, illustrated in Fig. 2, begins when the operator submits a natural-language request. The agent processes the query, determines which tool is required, invokes the corresponding module, and incorporates the returned result into its response. This response may include a direct answer, a synthesized interpretation of model outputs, or a visual representation of the requested information, depending on the operational context.

B. Leak-Detection Module

For requests related to network health assessment, the orchestrator invokes a GNN-based leak-detection model that operates on graph representations of the WDS. The agent can select a temporal snapshot of the network, submit it to the model, and interpret the resulting node-level predictions in a form suitable for the operator. The overall inference pipeline is summarized in Fig. 3. The GNN architecture comprises three stages: a feature encoder, a graph convolution module, and a regressor. The encoder transforms static and dynamic node attributes into a latent

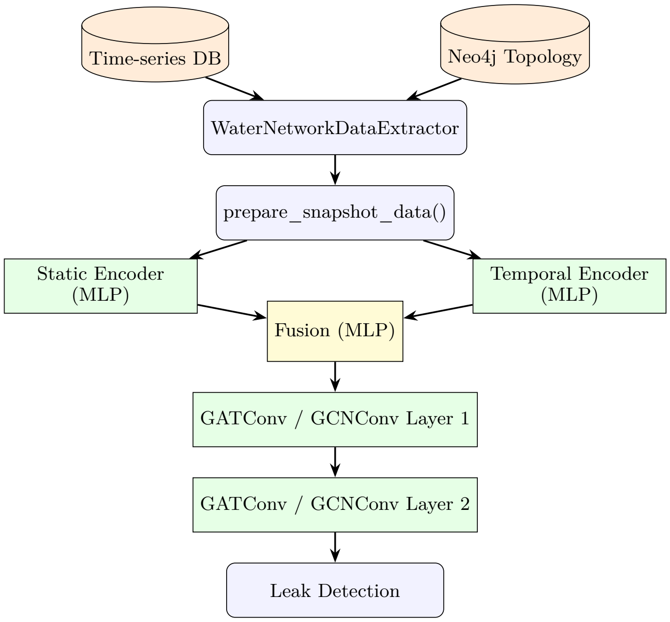


Fig. 3: Diagram of GNN architecture for leak detection.

TABLE I: GNN architecture variants.

Variant	Layer-1	Layer-2	Output
GAT (default)	GATConv(128, 128, heads=4)	GATConv(512, 128, heads=1)	128
GCN	GCNConv(128, 128)	GCNConv(128, 128)	128

representation using two symmetric Multilayer Perceptrons (MLPs), whose outputs are concatenated into a 128-dimensional embedding. The graph convolution module then propagates information across the network topology through two graph layers. In the default configuration, Graph Attention Convolution (`GATConv`) layers are used to learn edge-dependent attention coefficients, allowing the model to modulate the influence of neighboring nodes according to the local hydraulic context. A Graph Convolutional Network variant (`GCNConv`) is also available for scenarios in which lower computational cost is preferred. The layers are also summarized in table I. The final stage sets a regression module that maps the learned node embeddings to a continuous scalar value representing the estimated magnitude of the leak or hydraulic anomaly at each node. The module consists of three fully connected layers with ReLU activations, Batch Normalization, and Dropout, which stabilize training and reduce overfitting by regularizing the learned representations.

C. *Time-Series Query and Visualization Module*

The time-series query and visualization module enables natural-language access to historical and real-time measurements collected from the WDS. It is activated when the operator request concerns temporal trends, historical inspection, or comparative diagnostics. In the current implementation, these data are stored in a PostgreSQL database, where hourly node-level observations include variables such as pressure, demand, and piezometric head.

To support this interaction, the system uses an LLM-based query component that translates natural-language requests into executable SQL statements. The generation process is guided by a structured prompt encoding the database schema, admissible query patterns, security constraints, and domain-specific rules. These rules enforce the correct use of temporal filters, node identifiers, aggregation operations, and query limits, while preventing unsafe statements. Before execution, each generated query is validated to ensure both syntactic correctness and compliance with the imposed constraints. This validation step verifies the presence of the required SQL clauses, the explicit specification of temporal conditions, and the absence of malformed or unauthorized operations. If a query does not satisfy these conditions, it is reformulated automatically.

The final output includes the validated query together with its interpretation and the extracted query parameters. This design improves transparency and traceability, since operators can inspect how their request has been translated before the corresponding data are analyzed or visualized. It also allows users without direct knowledge of SQL to access database contents through domain-relevant expressions, including references to hydraulic variables, temporal intervals, and node-level comparisons. Within the overall architecture, this module acts as the interface between conversational requests and the underlying time-series database. In addition to retrieving data, it supports the generation of visual summaries and comparative views, extending the capabilities of the orchestrator from anomaly detection to broader exploratory and diagnostic analysis.

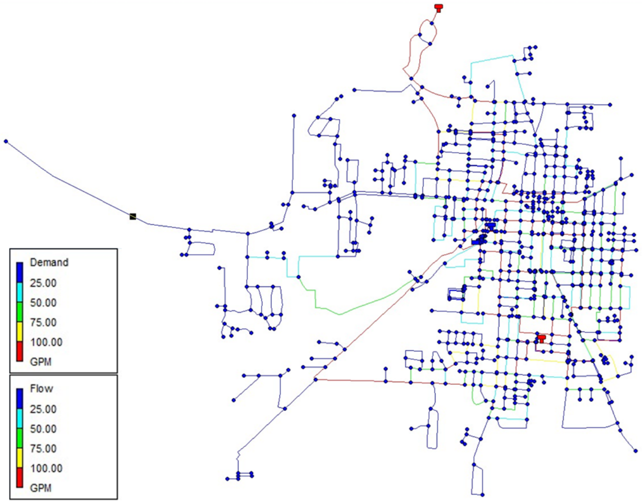


Fig. 4: Topology of the water network (815 nodes).

V. Experimental Configuration

This section summarizes the experimental setting used to evaluate the proposed framework, including the agent implementation, the GNN training procedure, and the simulated WDS dataset. The conversational agent is implemented with LangChain and follows the ReAct paradigm for iterative reasoning and tool invocation. The underlying reasoning model is Claude 3.5 Sonnet, accessed through the Anthropic API. To improve reliability in a critical-infrastructure setting, the model is configured with conservative generation parameters.

A. *Simulated WDS Dataset*

The experimental evaluation is based on simulated hydraulic data designed to reproduce realistic operating conditions of the water network under different leak regimes. Four scenarios are considered to assess model behavior across increasing anomaly severity. The *no-loss* scenario represents normal operation and serves as the reference condition. The *minimal loss* scenario introduces small leakages that are difficult to distinguish from regular consumption patterns. The *average loss* scenario represents intermediate anomalous conditions, whereas the *maximum loss* scenario includes pronounced leak events that generate clear hydraulic deviations.

B. *Network Description and Hydraulic Patterns*

The simulated network (Fig. 4) comprises 815 nodes and 1125 arcs representing primary and secondary pipes. This topology reflects a large-scale urban distribution system with a highly interconnected structure. Figures 5 and 6 show the average daily profiles of consumption, pressure, and piezometric head, together with their variability bands ($\pm 1\sigma$). A normalized comparison highlights the relative temporal dynamics of the three indicators. These profiles summarize the hydraulic behavior of the simulated system and support the interpretation of the leak scenarios. Correlation analysis reveals weak negative relationships between consumption and the other variables, with coefficients of -0.139 for consumption versus pressure and -0.345 for consumption versus piezometric head. Pressure and piezometric head show a positive correlation (0.391), consistent with the expected hydraulic behavior of the network. Based on these patterns, three relevant operating conditions are identified: (i) peak-demand periods, defined by consumption > 0.0003 L/s; (ii) low-pressure periods, defined by pressure < 45.27 m; and (iii) critical periods, where high demand and low pressure occur simultaneously.

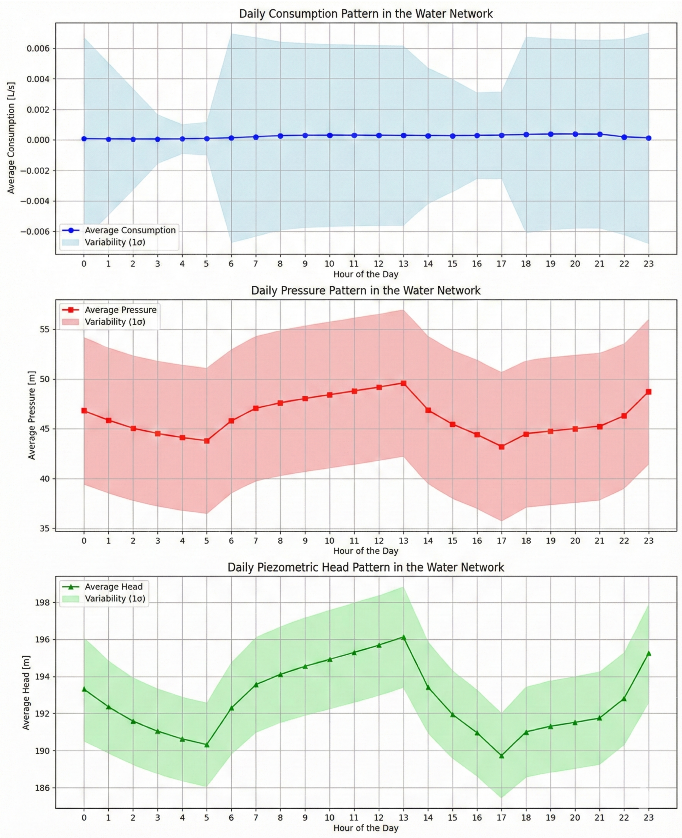


Fig. 5: Average daily trends of consumption, pressure, and piezometric head with mean and standard deviation.

VI. GNN Performance in Water Leak Detection

The proposed GNN was evaluated across the four scenarios described in the previous subsection: one scenario without leaks and three scenarios with leaks of varying severity. The training dataset was highly unbalanced, with only five out of 815 nodes experiencing leaks. The model's performance was assessed using standard metrics, including the Mean Absolute Error (MAE), Mean Squared Error (MSE), and the coefficient of determination (R^2).Despite the complexity and imbalance of the scenarios, the GNN demonstrated excellent predictive capabilities in estimating key hydraulic variables associated with leak events, including leak area, leak flow rate, and anomalous demand, with the results for each target summarized in Table II.

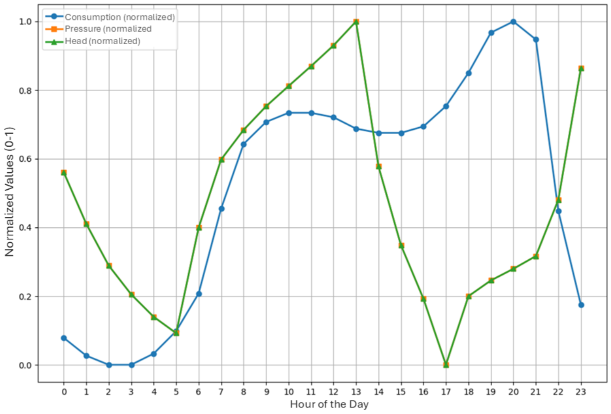


Fig. 6: Normalized comparison between daily consumption, pressure, and piezometric height patterns.

TABLE II: GNN performance for leak detection.

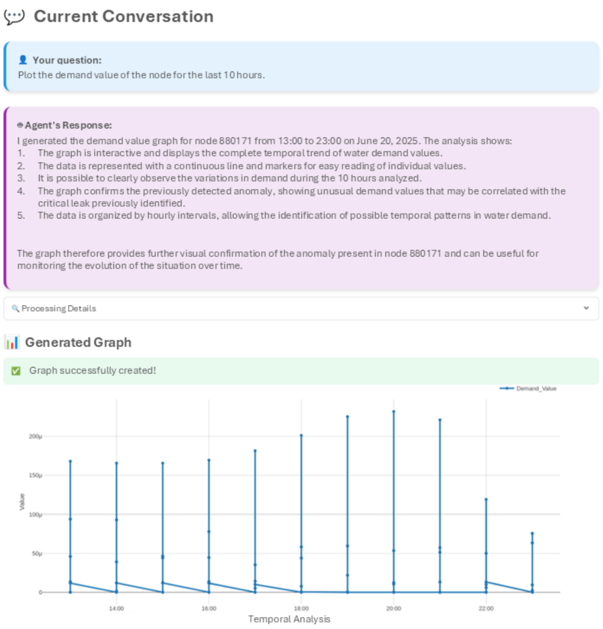
Target	**MAE**	R^2	**Loss**
Leak area	100 mm^2	0.8388	0.3088
Leak flow rate	0.0026 L/s	0.8388	0.2863
Anomalous demand	0.0002 L/s	0.9364	0.2234

Among the three targets, anomalous demand is predicted most accurately, with an R^2 of 0.9364. This high performance is further supported by a very low MAE of 0.0002 L/s, which is an order of magnitude smaller than the typical demand values that can reach up to 0.006 L/s. Leak area and leak flow rate also show solid performance, both with $R^2 = 0.8388$, indicating consistent predictive capability across the main leak-related variables. The error on leak area remains operationally acceptable. An MAE of 100 mm^2 corresponds to roughly 40–50% of the typical leak areas in the dataset, which range from 200 to 250 mm^2. Although this level of error limits precise geometric estimation, it remains sufficient for practical purposes such as rapid leak localization and approximate severity assessment.

VII. AI Agent System Interaction Examples

This section illustrates the operation of the proposed system through two representative interaction scenarios. The first example concerns time-series exploration through on-demand visualization, whereas the second focuses on node-level anomaly diagnosis through the leak-detection workflow.

a) Time-Series Visualization: Figure 7 presents an example in which the operator requests the water-demand



Current Conversation

Your question:
Plot the demand value of the node for the last 10 hours.

Agent's Response:
I generated the demand value graph for node 880171 from 13:00 to 23:00 on June 20, 2025. The analysis shows:
1. The graph is interactive and displays the complete temporal trend of water demand values.
2. The data is represented with a continuous line and markers for easy reading of individual values.
3. It is possible to clearly observe the variations in demand during the 10 hours analyzed.
4. The graph confirms the previously detected anomaly, showing unusual demand values that may be correlated with the critical leak previously identified.
5. The data is organized by hourly intervals, allowing the identification of possible temporal patterns in water demand.

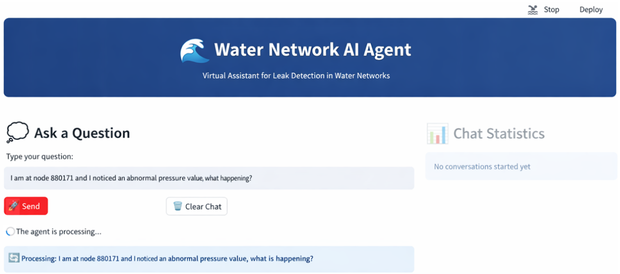
The graph therefore provides further visual confirmation of the anomaly present in node 880171 and can be useful for monitoring the evolution of the situation over time.

Generated Graph

✅ Graph successfully created!

Fig. 7: On-demand visualization of water demand at node 880171.

profile at node 880171 over the previous 10 hours. In response, the agent retrieves the relevant measurements and generates both a concise summary and a temporal plot of the selected variable. The visualization reports demand as a continuous time series with point markers, allowing direct inspection of hourly variations. This representation makes short-term fluctuations immediately visible and supports the identification of abnormal patterns. In the illustrated case, the plot shows unusually high demand values, consistent with the anomaly previously identified at the same node. The lower *Temporal Analysis* panel further emphasizes the main peaks and variations within the selected interval.



Water Network AI Agent

Virtual Assistant for Leak Detection in Water Networks

Ask a Question

Type your question:

I am at node 880171 and I noticed an abnormal pressure value, what happening?

The agent is processing...

Processing: I am at node 880171 and I noticed an abnormal pressure value, what is happening?

Chat Statistics

No conversations started yet

Fig. 8: Main interface of the AI agent system.

b) Node-Level Anomaly Diagnosis: Figure 8 shows the main interface used to submit diagnostic requests in natural language. In this example, the operator reports an abnormal pressure value at node 880171 and asks the system to identify its likely cause. After receiving the request, the orchestrator agent activates the appropriate analytical workflow, including the GNN-based leak-detection module, and returns a diagnostic response. An example of the generated response is reported below. The agent identifies a critical leak at node 880171 with a probability of 99.98%, confirms the pressure anomaly, and indicates that the event is localized because neighboring nodes exhibit low leak probability. Based on this evidence, the system recommends immediate maintenance intervention.

Example Agent Response for Leak Detection

Agent Response: A critical leak has been detected at node 880171 with a probability of 99.98%. Immediate intervention is recommended because:
- The system classifies the risk as **CRITICAL**.
- The pressure anomaly is confirmed by the analysis.
- Neighboring nodes show no significant anomalies (average leak probability ≈ 2%), indicating a localized issue.

Emergency maintenance procedures should therefore be activated for this node.

VIII. CONCLUSION

This paper presents an agent-based decision-support framework for water distribution systems (WDS) that integrates natural-language interaction with graph-based leak detection and time-series analysis. The architecture combines an LLM-based orchestrator with specialized modules for leak inference, database querying, and visualization, enabling the conversational agent to interpret operator requests, trigger the appropriate analytical tools, and provide outputs with concise operational explanations. Results show that the proposed GNN reliably estimates leak-related variables under realistic simulated conditions, performing well across leak area, leak flow rate, and anomalous demand. Experimental examples demonstrate that the agent translates these outputs into interpretable responses and visual analyses, reducing the effort required to understand heterogeneous system information. The framework highlights the potential of agentic AI as an orchestration layer for intelligent WDS management, improving both interpretability and operational usability of advanced monitoring technologies.

REFERENCES

[1] Abdelrahman M Farouk et al. Non-revenue water reduction strategies: a systematic review. *Smart and Sustainable Built Environment*, 12(1):181–199, 2023.

[2] Antonino Pagano, Domenico Garlisi, Fabrizio Giuliano, Tiziana Cattai, Redemptor Jr Laceda Taloma, and Francesca Cuomo. Introducing and evaluating swi-feed: A smart water iot framework designed for large-scale contexts. *Computer Communications*, 237:108146, 2025.

[3] Gabriele Restuccia, Domenico Garlisi, Fabrizio Giuliano, Antonino Pagano, and Ilenia Tinnirello. The synergy of digital twins and machine learning in sustainable water management: Swim. *Computer*, 58(12):46–54, 2025.

[4] Y. Goldshtein et al. Llm-epanet: A natural language interface for hydraulic simulation. *Water Resources Research*, 2025.

[5] A. Sela et al. Automatic data analysis for water networks using llms. *Journal of Hydraulic Engineering*, 2024.

[6] R. Taormina and H. van der Werf. Explaining vision models for infrastructure damage with llms. *Automation in Construction*, 2024.

[7] J. Marzouny and J. Dziedzic. Interactive pump scheduling via chatgpt. *Journal of Water Resources Planning*, 2024.

[8] L. Cheng et al. Gaia: Llm-assisted dispatching for power networks. *IEEE Transactions on Power Systems*, 2025.

[9] Yi Ren, Tianyi Zhang, Xurong Dong, Weibin Li, Zhiyang Wang, Jie He, Hanzhi Zhang, and Licheng Jiao. Watergpt: Training a large language model to become a hydrology expert. *Water*, 16(21):3075, 2024.