Unsupervised Anomaly Detection in Urban Water Networks Using a Hierarchical Deep Learning Model

Guillem Escriba*
Universitat Pompeu Fabra
Barcelona, Spain

David Pérez*
Universitat Pompeu Fabra
Barcelona, Spain

Nicolás Vila*
Universitat Pompeu Fabra
Barcelona, Spain

Daniel Marín
Universitat Pompeu Fabra
Barcelona, Spain

Miquel Oliver
Universitat Pompeu Fabra
Barcelona, Spain

***Abstract*—Efficient anomaly detection in urban water distribution networks is crucial for sustainable resource management. Identifying abnormal patterns in time-series data from sensors is a common task within industrial settings and automated methods can enhance efficiency and reduce costs. However, supervised learning is often impractical due to the scarcity of labeled anomalous data in settings of practical interest. Consequently, unsupervised techniques are typically used, identifying anomalies as deviations from normal behavior. This paper introduces a framework that combines Long Short-Term Memory (LSTM) and Transformer encoder-decoder networks in a hierarchical structure to predict future water usage and detect anomalies in an unsupervised manner. Our method optimizes the use of available data within practical settings, where partial information might not be available at some points, ensuring high inference efficiency and enabling near real-time anomaly detection. It is adaptable to various data configurations, making it suitable for different environments. Our approach, applied to data from the metropolitan area of Barcelona, highlights its potential to refine early anomaly detection, prevent resource loss, and promote sustainable consumption.**

***Index Terms*—Anomaly detection, machine learning, water distribution networks, LSTM networks, transformer models, unsupervised learning, smart cities, time-series analysis**

I. Introduction

Efficient management of urban water distribution networks is critical for the sustainability of urban areas. Anomalies in water usage patterns can lead to significant resource waste, financial losses, and potential disruptions in service. However, the detection of these anomalies is challenging due to the complexity and variability of time-series data collected from numerous sensors throughout the network. Traditional supervised learning approaches are often unfeasible in this context due to the limited labeled anomalous data, which makes the application of unsupervised techniques more suitable [1].

This paper introduces a novel framework to address these challenges by leveraging an ensemble of advanced machine learning architectures, specifically Long Short-Term Memory (LSTM) networks and Transformer encoder-decoder networks. Integrating these models in a hierarchical structure, our approach predicts future water usage and detects anomalies without relying on labeled data.

The proposed framework is adaptable to various data configurations, crucial in real-world scenarios where data may be incomplete or partially unavailable. This flexibility makes the method suitable for diverse urban environments and increases its applicability across different settings. To validate our approach, we applied it to water consumption data from the metropolitan area of Barcelona. The results show its potential to significantly improve early anomaly detection, preventing resource loss and promoting sustainable water consumption.

We address two commonly overlooked challenges: handling missing data and ensuring usability at scale. The model effectively utilizes all available data, and even with missing relevant variables, we can shift to a higher hierarchical level to still perform accurate inference. This approach maintains feasibility in large water management systems. This unsupervised method optimizes data utilization and ensures high inference accuracy, enabling near real-time anomaly detection. By leveraging an ensemble of two deep learning models, our contribution strengthens the robustness and reliability of water management systems.

In the following sections, we will get into the specifics of our framework, discuss the integration of LSTM and Transformer models, and present our experimental findings. Our goal is to highlight the practical benefits of this approach for smart city applications, particularly in advancing the efficiency and sustainability of urban water distribution networks.

II. Related Work

Anomaly detection methods can generally be classified into two main categories: traditional machine learning methods and deep learning (DL) methods. Traditional machine learning methods, such as the One-class SVM (Support Vector Machine) [2], Isolation Forest [3], and random forest-based models [4], have been widely used. However, these models often have clear limitations as they typically do not consider

[*] These authors contributed equally

the temporal dependencies of the data and struggle to perform well with large amounts of multivariate data [5].

In recent years, the advent of deep learning has led to the development of more advanced and sometimes more effective methods for anomaly detection [5]. Examples include LSTM networks, Autoencoders, and Transformer architectures [6]–[8]. These architectures can capture complex patterns and dependencies in data, making them suitable for detecting anomalies in time-series data.

A. Anomaly Detection in Water Systems

Within the water sector, the concept of anomaly can differ based on the characteristics of the data and the specific objectives, ranging from pure water analytics and water quality control to identifying shifts in household consumption patterns. Each use case requires different anomaly detection methods.

1) Traditional Methods in Water Analytics: Most work in water analytics utilizes clustering techniques to detect outliers in the data. Anomaly detection approaches based on nearest neighbors rely on local data density derived from the k-nearest neighbor algorithm. Regular data instances tend to form clusters within dense neighborhoods, while anomalies are often located a significant distance away from these clusters [9], [10].

Breunig et al. [9] proposed the Local Outlier Factor (LOF), which assigns an outlier score based on the ratio of the average density of a point's k-nearest neighbors to the point's own local density. This method has been adapted in various forms for different applications, including water analytics.

For instance, research has been directed toward detecting anomalies in water distribution or water quality over time [11]. Fagiani et al. [12] compared several unsupervised methods, such as Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), and One-class SVM, for leakage detection in water and natural gas grids. Similarly, other studies have used machine learning to identify malfunctioning meters, leaks, and fraud in water systems [13], [14]. Some approaches employ hybrid strategies, combining Support Vector Machines (SVMs) and Decision Trees to discover sensor defects and classify anomalies [14].

2) Deep Learning Methods in Water Systems: The application of deep learning for anomaly detection in water management systems is a developing area with promising potential. Existing research primarily focuses on utilizing LSTM architectures to identify anomalies within water quality data and management systems [15], [16]. LSTMs are particularly well-suited for this task due to their ability to capture temporal dependencies and patterns inherent in sequential data, which is prevalent in water consumption measurements. Another approach explored in this domain involves autoencoder-based architectures. Here, anomaly detection relies on the discrepancy between the reconstructed and original data sequences. The underlying assumption is that the model successfully encodes typical data patterns [17].

While we identified one instance of transformers being applied within water systems [18], it deviates significantly from our proposed approach. Their work utilizes transformers for classification tasks, which fundamentally differs from our objective, which would be more similar to the autoencoder setting.

III. Description of our data and Formal definition of the problem

A. Dataset

In this study, we have used data supplied by 'Aigües de Barcelona'. This data has been collected daily through digital water meters. Due to privacy restrictions, only aggregated data from several meters was accessed. That is, each daily consumption corresponds to the aggregation of several meters. The data from the meters is classified according to location, day, the number of meters, and daily consumption (in liters per day).

To accurately identify and interpret anomalies in our data, it is essential to consider external factors not captured in our original dataset. These factors can arise from events or conditions within the urban setting. Therefore, we integrated additional relevant open data sources to offer a broader context. Specifically, we incorporated weather and socioeconomic data to strengthen the model's ability to distinguish true anomalies from those influenced by external factors. Additionally, to help the model understand temporal patterns, we added columns for the day of the week, day of the year, and a global day count. These features were crucial for capturing patterns over time. By including these elements, we ensured a more comprehensive analysis and more accurate identification of true anomalous patterns.

Our dataset spans from January 1, 2019, to December 31, 2022, covering a four-year period with daily observations. In this study, we analyze these observations as distinct time-series. Each series is characterized by geolocation (`Postcode`), a binary indicator denoting whether the meter is designated for commercial or industrial use (`Use`), and the corresponding `Economic activity` associated with the meter. Thus, each tuple `(PostCode, Use, Economic Activity)` will correspond to a unique time-series resulting in a total amount of combinations of over 23,000 different time-series in our use case.

B. Data pre-processing

Data preprocessing is essential for preparing our dataset for the model, as illustrated in the pipeline in Fig. 1. One challenge was the varying number of water meters over time, which we addressed by normalizing the consumption data per meter. We also encountered multiple records for the same date, which we consolidated by summing the meters and retaining the highest consumption value. This approach ensured anomalies were not averaged out. Negative consumption values, which were unusual, were replaced with averages from the surrounding days to maintain data integrity.

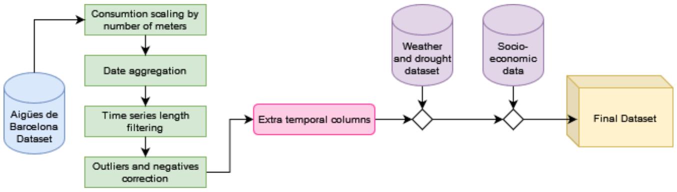
Consumtion scaling by number of meters
Date aggregation
Time series length filtering
Outliers and negatives correction
Aigües de Barcelona Dataset
Extra temporal columns
Weather and drought dataset
Socio-economic data
Final Dataset

Fig. 1: **Preprocessing of the data**.

Outlier handling involved removing values that were more than 30 times the interquartile range (IQR) and replacing them with surrounding averages. This threshold, significantly higher than the typical 1.5 times the IQR, was chosen to clearly differentiate between outliers in the data and genuine anomalies in consumption. Missing single-day data was similarly filled to ensure continuity.

C. Formal definition of the problem

The detection of anomalies in temporal data can be formalized as follows. Consider a collection of temporal data, $X_n = [\vec{x}_{n1}, \ldots, \vec{x}_{nT}]$, where $t \in \{1, \ldots, T\}$ is the temporal index, $\vec{x}_{nt} \in \mathbb{R}^m$ is an m-dimensional vector representing the variables collected on day t, and T is the total length of the series. On the other hand, $n \in \{1, \ldots, N\}$ represents the corresponding time-series, and N is the total number of time-series in the data. In our case, the maximum value of T will be 1461 (4 years of daily data), but it could vary if data were missing in some time-series or if some meters were incorporated later than others. Our goal is to find a variable $\mathbf{A}_n = (a_{n1}, \ldots, a_{nT})$, where $a_{nt} \in \{0, 1\}$, and each entry represents whether the data is anomalous or not. For this paper, we will consider an anomaly to be a data point or a series of data points that deviate significantly from the established pattern or distribution of normal consumption behavior. In this context, $a_{nt} = 0$ indicates normal consumption, while $a_{nt} = 1$ indicates anomalous consumption.

In the unsupervised case, Y is initially unknown. Our approach involves estimating the value of y directly from the data x. While it is possible to incorporate prior knowledge or labeled data to extend the hierarchical model training to a supervised framework, this paper will focus exclusively on the unsupervised scenario.

IV. Methodology

Traditional machine learning algorithms often struggle with anomaly detection in complex multidimensional time-series, and have difficulties to fully leverage the potential of large datasets. In this paper, we introduce a deep learning model designed to handle large volumes of data effectively. Our model integrates transformers and LSTMs, supported by specific data engineering techniques within their training phase, as well as other additional components that will be described in detail in this section.

A. Hierarchical Training

One of the fundamental parts of the implementation of our proposed solution lies in its training method: a hierarchical training scheme. Unlike conventional training routines, in hierarchical training, a series of sequential stages are established through which the data will pass during the training phase.

We define as stage each of the consecutive trainings that result in a new model fine-tuned from a previous one. In each subsequent stage, a progressively smaller subset of data is used for each previous model, resulting in increasingly specialized models. The stages are defined by a set of attributes that describe each individual time-series: (`Use, Economic Activity, PostCode`). These attributes are chosen after a data engineering process that ensures the following condition: at each stage, the previously trained model is refined by dividing the complete dataset into unique, non-overlapping subsets, ensuring that each time-series belongs to only one subset. In this particular use case, the training stages, illustrated in Fig. 2, are defined as follows:

- Stage 1: This is the most general stage where the models are trained with the whole dataset without any further differentiation.
- Stage 2: Using the previously trained model, it is fine-tuned with 2 different subsets. These new subsets are created by dividing the samples according to the attribute `Use`, differentiating between Industrial and Commercial uses. As a result, we obtain 2 new specialized models.
- Stage 3: Similarly to the previous stage, we use the Use-specific models and we fine-tune them. However, now the subsets are smaller, defined by (`Use, Economic Activity`). This stage results in even more specialized models, which capture the different trends across similar time-series.
- Stage 4: Finally, in the last stage, we specialize the previously trained models for each unique time-series (water-meter) defined by the following unique subset (or key): (`Use, Economic Activity, PostCode`). Thus,

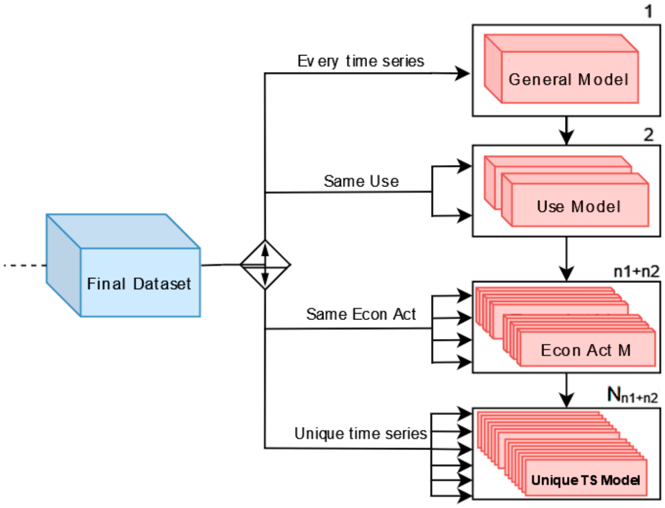

Fig. 2: **Hierarchical Training.** n1: number of economic activities of Commercial Use, n2: number of economic activities of Industrial Use, N: number of postcodes

we have a unique model for each time-series, enabling the detection of patterns and nuances in all of them.

The number of training stages depends on the number of variables used to create the subsets - the number of available data for each time-series -, with a minimum of two stages: a general and a specialized one. The number of stages is defined by the number of variables that can lead to unique time-series differentiation, in our case (`Use,Economic Activity,PostCode`). In the case of temporal variables, although it is possible to create subdivisions such as monthly or seasonal, these could restrict the model's capacity and its interaction with other stages because these subsets are restricting the temporal span of each subset. Thus, the amount of temporal data used in the model's training is reduced. Therefore, the selection and number of stages must balance the specialization and the generalization capacity of the model.

In hierarchical training method, there are two main approaches to handling the models generated at each stage during inference:

- Discard Previous Models: In this approach, the model only performs inference with the models resulting from the last stage. This increases specialization but limits inference to previously trained data and may increase the risk of overfitting. However, it reduces execution time for inference.
- Retain Previous Models: This approach uses a model to integrate the results of all stages, improving accuracy and allowing inferences on new time-series that have not yet been seen using the more general stages. The higher accuracy is at the cost of longer execution time in the inference phase. However, each stage can be executed in parallel, reducing the computational cost and making it as feasible as the previous approach for a production environment.

The advantages and justification of the hierarchical training scheme are based on the following reasons:

1) Preservation of Generalization: Although it specializes in the final stages, the model maintains a hierarchical structure that preserves the generalizations of previous models, achieving a balance between adaptability and robustness.
2) Variability of time-series: Significant differences between time-series from different sectors or areas justify a specialized approach to capture the particularities of each group.
3) Continuous Data Growth: As data continuously increases, the model's accuracy improves over time, taking advantage of the use of Deep Learning models which require sizeable amounts of training data.
4) Focus on Anomaly Detection: A specialized model can be more effective at identifying deviations or anomalous behaviors.

The combination of a hierarchical approach with specialization allows the model to efficiently address the complexity and variability of time-series, as well as maintain generalization when necessary.

B. Model architecture

An overview of our model's architecture is illustrated in Fig. 3, showcasing the different models obtained from the hierarchical training approach, including the combination of Transformer and Multi-LSTM outputs. Starting from each time-series X_n, the input of the model after its respective preprocessing would be a tensor $X \in \mathbb{R}^{N\times T\times M}$, and its output would be another tensor $Y_\in \mathbb{R}^{N\times T\times 1}$.

a) Transformer Anomaly Detection (TranAD) [8]: The first component of our model is TranAD, a transformer-based model [19], which is widely used in natural language processing (NLP) and Computer Vision. TranAD leverages the transformer architecture to detect anomalies in time-series data.

The transformer in TranAD functions as an encoder-decoder model comprising 2 encoders and 2 decoders. The first encoder processes a complete input sequence X_n and generates a 'Focus Score'. In the second encoder, the input series is divided into W temporal windows, with S as the window size, denoted as $\tilde{X_{ni}} = [\vec{x}_{n(1+i)}, \ldots, \vec{x}_{n(S+i)}]$ where $i \in \{0, 1, \ldots, W\}$. This generates an encoded representation of each input window, which is then used by two decoders to reconstruct the input sequence.

Training TranAD involves an adversarial procedure with two phases. In the first phase, the model generates an approximate reconstruction of the input window. The deviation of this reconstruction, referred to as the 'Focus Score', helps the attention mechanism in the transformer's encoder to better extract temporal trends, particularly in subsequences with high deviations. In the second phase, these deviations condition the model's output, leading to the final reconstructed sequence Y_n^{Tran}. One of the main advantages of TranAD is its efficiency; training can be up to 99% faster compared to other similar models (see Table 5 in [8]).

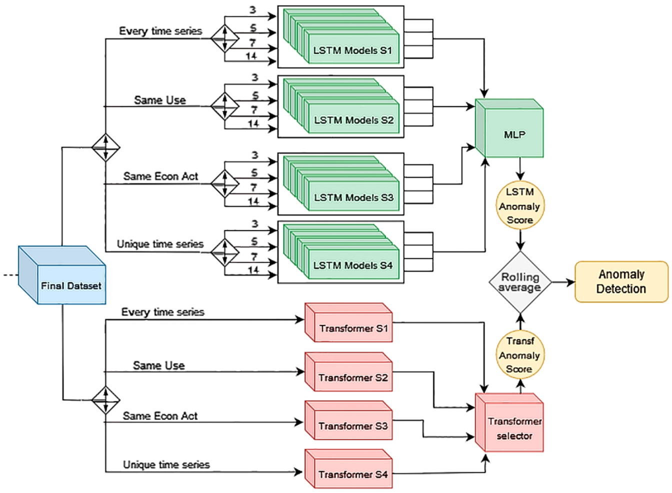

Fig. 3: **Model architecture**

Additionally, TranAD performs well even with limited training data, achieving comparable performance to other deep learning models while using a significantly smaller fraction of the data (see Table 3 in [8]). For a more comprehensive understanding of TranAD's performance and methodology, refer to the original work by [8].

We train four transformer models based on the hierarchical training procedure described earlier. For inference, we use the most specialized model that matches the data available. Specifically, if we have data for (`PostCode, Use, Economic Activity`) we use the most specialized model. However, if (`Economic Activity`) data is missing, we use the model from the previous stage of our hierarchical training process. This process is illustrated by the transformer stages and selector in Fig. 3.

b) Multi-LSTM: The Multi-LSTM, an advanced implementation of conventional LSTMs, has been designed to detect and model patterns across different temporal windows in parallel. This capability makes it an essential tool in the model, where it works in conjunction with TranAD, the other main neural network of the system. LSTMs, a variant of recurrent neural networks, are widely recognized for their ability to learn long-term dependencies in data sequences, making them ideal for applications such as natural language processing and time-series prediction.

However, conventional LSTMs have limitations, especially in contexts where it is necessary to analyze multiple time scales or sequences with complex and varied patterns. This is where the Multi-LSTM comes into play, which extends the capabilities of standard LSTMs by allowing the simultaneous analysis of multiple temporal windows. This feature is particularly valuable for capturing patterns that occur at different time scales, improving the accuracy and robustness of the model in complex tasks. LSTMs operate using temporal windows in which the S previous days are used to predict the value of day $S+1$. Moreover, in our model, both the number of LSTMs, defined by the constant L, and the size of the temporal windows $S_1, S_2, \ldots, S_L$, are fully parameterizable, thus allowing better adaptability to each specific case.

Starting from the input tensor X, certain modifications are required to adapt it to the Multi-LSTM. To begin, we select the variables that are useful for prediction, which do not necessarily have to be the entire set m nor the same subset m^{TranAD} used by the Transformer. This subset will be called m^{LSTM}. Due to their architecture, LSTM networks operate on temporal windows of size S. For each time series, overlapping subsequences are considered. The number of such subsequences for a single time series n of length T_n is the total number of windows of size S that fit in the time series. This number is given by $T_n - S + 1$. Let X_{ns} be a subsequence of size S from the same time-series n formed by vectors of dimensionality $p = |m^{\text{LSTM}}|$, then, its dimensionality will be $X_{ns} \in \mathbb{R}^{p \times S}$. Thus, the total number of subsequences for the set of time-series N will be: $G = (\sum_n (T_n - S + 1))$, in essence the sum of the number of subsequences of each time-series. In this way, a tensor of dimensionality $X^{LSTM} \in \mathbb{R}^{G \times p \times S}$ would enter the LSTM. However, in the case of Multi-LSTM, since it captures multiple temporal windows at the same time, a tensor of size $X^{LSTM} \in \mathbb{R}^{L \times G \times p \times S}$, where L is the number of different windows for each of the LSTMs, which is a tuneable parameter that can increase or decrease the amount of LSTM, will enter at each stage. Thus, the output at each stage of the Multi-LSTM will be a tensor $Y'^{\text{LSTM}} \in \mathbb{R}^{L \times N \times T \times 1}$, which will subsequently be transformed into $Y^{\text{LSTM}} \in \mathbb{R}^{N \times T \times 1}$ using the MLP neural network, which we will discuss next.

c) MLP: The last neural network composing the model is a MLP (Multi-Layer Perceptron). Its function is to evaluate and weigh the results of each of the different stages and neural networks. Its architecture is much simpler than that of the rest of the components because the amount of data to process and the complexity of the patterns is much less. Essentially, what the MLP does is grouping a series of predictions obtained from different neural networks into a single and more accurate one, taking the generalizations from the earliest stages with the specializations of the latest when necessary. In this way, all the models from previous stages can be combined into one, thus ensuring generalization and preventing overfitting as well as maintaining high accuracy and specialization for each time series. The input tensor of the MLP is the output of each of the E stages of the Multi-LSTM, that is, $X_{MLP} \in \mathbb{R}^{(E \cdot L) \times N \times T \times 1}$ and produces an output $Y_{MLP} \in \mathbb{R}^{N \times T \times 1}$.

C. Mutual Complementation and Advantages

The model integrates two deep learning architectures, TransAD for precise anomaly detection and Multi-LSTM for handling temporal dependencies, to improve performance in anomaly detection tasks. This ensemble approach provides a robust analysis, making the model adaptable to different data types while reducing overfitting risks. The complementary nature of these architectures ensures that the model benefits from their respective strengths, resulting in a more balanced and effective anomaly detection system. Furthermore, the hierarchical training approach facilitates the incorporation of new time-series data, even with limited data, ensuring efficient and accurate inference.

D. Anomaly Detection

To identify anomalous data within our model, we implement a formula that integrates the predictions of both models, emphasizing temporal coherence. This methodology ensures both methodological soundness and reliable anomaly detection. Our tool offers flexibility by allowing users to adjust numerical criteria to their preferences while maintaining a well-defined and consistent process.

To describe our methodology, we utilize the previously defined X_n, excluding data unrelated to water consumption, as it is not pertinent to this section. Thus, we consider x_{nt}^{cons} as the value representing the liters of water consumed by each meter on day t in sequence n. Similarly, we define $Y_n^{\text{MLP}} = (y_{n1}^{\text{MLP}}, y_{n2}^{\text{MLP}}, \ldots, y_{nT}^{\text{MLP}})$ and $Y_n^{\text{Tran}} = (y_{n1}^{\text{Tran}}, y_{n2}^{\text{Tran}}, \ldots, y_{nT}^{\text{Tran}})$, where $t \in \{1, \ldots, T\}$ is the previously defined temporal index,

$y_{nt}^{\mathrm{MLP}} \in \mathbb{R}$ is the water consumption predicted by the MLP for sequence n on day t, and $y_{nt}^{\mathrm{Tran}} \in \mathbb{R}$ is the prediction by the Transformer model. The initial step in this process involves the conventional procedure for detecting numerical anomalies, which entails calculating the difference between the actual consumption values and our predictions. Consequently, we define $\mathbf{L}_n^f = (l_{n1}^f, l_{n2}^f, \ldots, l_{nT}^f)$ where $l_{nt}^f = |x_{nt}^{\mathrm{cons}} - y_{nt}^f|$ and f corresponds to either MLP or Tran. While various methods exist for calculating the discrepancy between sequences and their predictions, we selected the L1 loss due to the peculiar characteristics of consumption data. The primary reason for this choice is the robustness of L1 loss to outliers, which improves our ability to identify them, unlike other functions such as L2 loss that are highly sensitive to such values [20].

To detect anomalies based on temporal context, we define a hyper-parameter $\mathcal{K} \in \{0, \lfloor \frac{T-1}{2} \rfloor\}$, where T is the length of our sequences. This parameter facilitates the calculation of a moving average of the error, thus capturing anomalies extended in time. $\mathcal{K}$ represents the number of days before and after considered in this calculation, allowing the identification of anomalies over longer periods as $\mathcal{K}$ increases. Accordingly, we obtain $\mathbf{E}_n^f = (e_{n1}^f, e_{n2}^f, \ldots, e_{nT}^f)$ where $e_{nt}^f = \frac{\sum_{i=t-\mathcal{K}}^{t+\mathcal{K}} l_{ni}^f}{2\mathcal{K}+1}$, and l_{ni}^f is the value of l^f on day i in sequence n, appropriately bounding both ends, thus readjusting the divisor term when fewer values are considered due to these limits.

Before combining the results of both models, we independently detect anomalous values according to each model. In this step, we use statistical percentiles, introducing another hyperparameter, $\mathcal{P} \in (0, 100)$, which can be configured by the user. This value, $\mathcal{P}$, represents the percentile threshold, indicating the proportion of data deemed statistically anomalous. For instance, if we set $\mathcal{P} = 1$, we identify approximately 1% of the data as anomalous for each model by evaluating the moving average of the error function between the model's prediction and the actual consumption value. Therefore, we obtain $\mathbf{A}_n^f = (a_{n1}^f, a_{n2}^f, \ldots, a_{nT}^f)$ where $a_{nt}^f \in \{0, 1\}$. A value of 1 indicates anomalous consumption on day t according to the described criterion, while a value of 0 indicates normal consumption.

To further increase the rigor of our approach, the combination of the anomalies detected by both models uses an additional aggregation method with rolling windows. We obtain $\mathbf{A}_n^{\mathrm{agg}} = (a_{n1}^{\mathrm{agg}}, a_{n2}^{\mathrm{agg}}, \ldots, a_{nT}^{\mathrm{agg}})$, where $a_{nt}^{\mathrm{agg}} \in \{0, \ldots, 2(2\mathcal{K}+1)\}$. Specifically, $a_{nt}^{\mathrm{agg}} = \sum_{i=t-\mathcal{K}}^{t+\mathcal{K}} a_{ni}^{\mathrm{MLP}} + a_{ni}^{\mathrm{Tran}}$. In other words, each value corresponds to the number of anomalies detected in the interval of days $[t-\mathcal{K}, t+\mathcal{K}]$, adding both models. If we have defined $\mathcal{K} = 1$, we would have $a_{nt}^{\mathrm{agg}} \in \{0, 6\}$, where the maximum value would occur in the case of maximum possible concordance between both models in the temporal context.

Ultimately, the final identification of anomalies will depend on the priority the user gives to this concordance, which can be determined by defining the third and last hyper-parameter, $\mathcal{Q} \in \{1, \ldots, 2(2\mathcal{K}+1)\}$. From this, we obtain $\mathbf{A}_n = (a_{n1}, a_{n2}, \ldots, a_{nT})$, where $a_{nt} \in \{0, 1\}$, 0 if $a_{nt}^{\mathrm{agg}} < \mathcal{Q}$ and 1 if $a_{nt}^{\mathrm{agg}} \geq \mathcal{Q}$.

V. Experiments and Results

We conducted multiple tests, focusing on the most relevant and definitive experiments. With a percentile threshold ($\mathcal{P}$) of 1%, a temporal contiguity ($\mathcal{K}$) of 1 day, and varying the anomaly detection threshold ($\mathcal{Q}$), we identified a significant number of anomalies. The choice of the 1% percentile was motivated by its capacity to produce a balanced and representative distribution of anomalous events. Similarly, selecting a 1-day temporal contiguity allowed us to capture anomalies that exhibit short-term persistence.

With a threshold of $\mathcal{Q} = 4$, we detected a total of 52,000 anomalies, corresponding to approximately 0.55% of the total data. Reducing the threshold to $\mathcal{Q} = 3$ resulted in about 1% of the data being classified as anomalous. The choice of threshold in our anomaly detection approach significantly impacts the number and types of anomalies detected. Selecting a lower threshold (i.e., $\mathcal{Q} < 2\mathcal{K} + 2$) allows for greater variability and breadth in the detected anomalies, capturing diverse anomalous patterns. Conversely, a higher threshold (i.e., $\mathcal{Q} \geq 2\mathcal{K} + 2$) ensures greater consistency and rigor in anomaly detection, albeit with reduced variability. This flexibility, enables users to tailor the detection process to their specific needs, balancing precision and breadth in anomaly identification.

Anomalies were identified through the integration of different components from our model, as detailed in the previous section. Fig. 4 illustrates the predictions for a selected time-series. The top plot displays predictions from our MLP model, which integrates all the LSTM modules (Y_n^{MLP}), while the middle plot shows predictions from the TranAd model (Y_n^{Tran}). The bottom plot presents aggregated anomaly scores from each model ($\mathbf{A}_n^{\mathrm{MLP,agg}}$ and $\mathbf{A}_n^{\mathrm{Tran,agg}}$) alongside the combined anomaly score ($\mathbf{A}_n^{\mathrm{agg}}$).

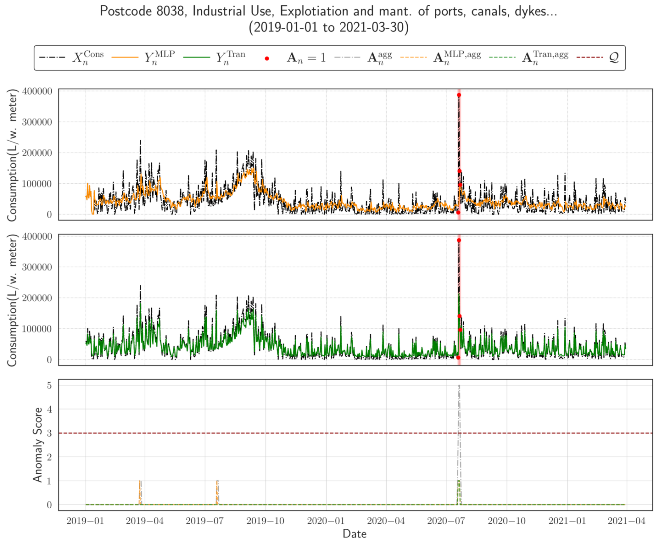

Fig. 4: **Anomaly detection results for a complete time-series.**

Anomalies are flagged ($\mathbf{A}_n = 1$) when the combined anomaly score exceeds the specified threshold ($\mathcal{Q}$). This method was applied to each tuple in our dataset, defined by `(PostCode, Use, Economic Activity)`. Although earlier deviations in the MLP predictions were noted, they did not persist over multiple days nor coincide with deviations from the TranAd model, and thus did not cause $\mathbf{A}_n^{\text{agg}}$ to exceed $\mathcal{Q}$.

Fig. 5 illustrates a shorter segment of a different time-series, highlighting several identified anomalies. In this instance, we can observe different types of anomalies. The first anomaly is detected solely due to the TranAd model's prediction deviating significantly from the actual consumption, while the MLP prediction remains closer to the actual values. Shortly after, another anomaly is identified uniquely from the MLP model, demonstrating the strategy's robustness in detecting anomalies even when only one model indicates a significant deviation.

Further along the timeline, a cluster of anomalies is detected due to prolonged differences between both models and the actual consumption. This results in the moving average of $\mathbf{A}_n^{\text{MLP,agg}}$ and $\mathbf{A}_n^{\text{Tran,agg}}$ reaching a high combined anomaly score ($\mathbf{A}_n^{\text{agg}}$), highlighting the temporal consistency of these anomalies.

The variety of detected anomalies demonstrates the reliability and precision of our model, which effectively identifies anomalies through model concordance or temporal persistence, both serving as relevant indicators. A key strength of this approach is its adaptability; we can adjust the threshold ($\mathcal{Q}$) to balance between strict concordance and individual model deviations. For example, setting $\mathcal{Q} \geq 2\mathcal{K}+2$ ensures that only the most consistent anomalies are detected, requiring strict agreement between models.This approach leads to insightful results, as the distribution of anomalies by use, economic activity, or geographic area can be very helpful for understanding the behavior of water network leakages.

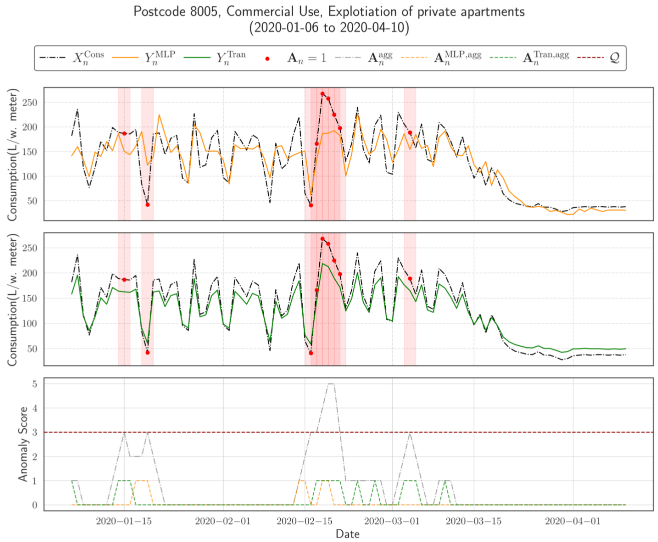

Fig. 5: **Anomaly detection results for a time-series segment.**

Specifically, the distribution of anomalies by industrial or commercial use is detailed in Table I, while Table II illustrates the distribution of anomalies across the top 10 economic activities with the most anomalies.

Within our dataset, 80% of the data pertained to commercial use, while 20% was related to industrial use. Given this distribution, it is anticipated that commercial settings would exhibit a higher number of anomalies compared to industrial settings. However, the actual frequency of anomalies does not scale proportionally with this distribution. Instead, commercial settings display a higher susceptibility to anomalies, with a significant proportion of both anomalies and water loss occurring in these areas. Industrial time-series, despite being less frequent, generally involve larger amounts of water. This is evident in Figure 4, which shows a time-series related to ports and canals exploitation and maintenance, reaching up to 400,000 liters. In contrast, Figure 5 represents private apartments with a peak of around 250 liters.

Among the economic activities, those with high operational demands, such as vehicle storage and fork manufacturing, exhibit the highest number of anomalies and associated water loss. These activities do not necessarily consume the most water but require constant water consumption, making them more prone to irregularities. Furthermore, the table highlights the top economic activities by the number of anomalies detected, not by the volume of water lost. It is probable that industrial activities, which use larger amounts of water, have fewer anomalies but more significant water loss per anomaly.

The geographic distribution of the anomalies is illustrated in Fig. 6, which presents a detailed map of Barcelona where neighborhood color intensity indicates the number of identified anomalies within the corresponding postcodes.

A significant insight from these results is the concentration of up to 5% of the total anomalies in a suburb of Barcelona, despite its water consumption not equating to that proportion of the total water consumption in the metropolitan area. This disproportionate anomaly concentration suggests potential is-

TABLE I. Distribution of Anomalies by Use

Use	Anomalies		Liters Lost (L)
	Number	**Percentage (%)**	
Commercial	88,258	89.0928	652,171.07
Industrial	10,805	10.9072	91,958.35

TABLE II. Distribution of Anomalies Across Economic Activities

Economic Activity	Anomalies		Liters Lost (L)
	Number	**Percentage (%)**	
Vehicle Storage and Parking	947	0.956	4,108.95
Fork Manufacturing	917	0.926	5,412.82
Electronic Exploitation	907	0.916	2,962.14
Hotel and Motel Services	886	0.894	4,223.82
Hairdressing Services	812	0.820	5,010.52
Residential and Social Services	757	0.764	4,368.44
Communication EMA Rate C1A	704	0.711	3,544.98
Other NCAA Services	691	0.698	3,616.26
Closed Premises	671	0.677	4,187.54
Industrial Premises and Rentals	665	0.671	4,642.68

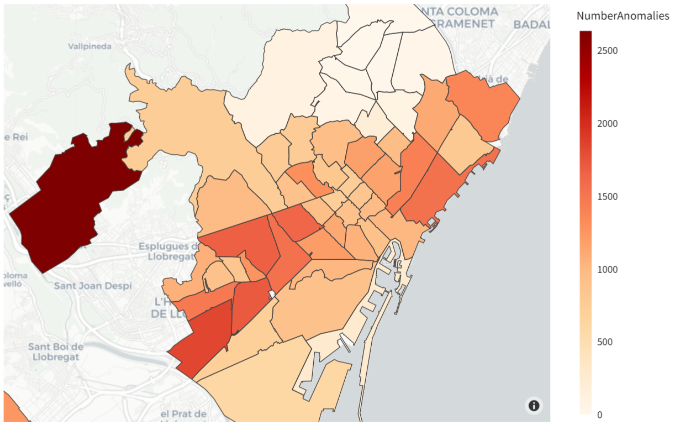

Fig. 6: **Geographic Distribution of Anomalies in Barcelona.**

sues specific to this suburb that warrant further investigation. Additionally, the results reveal that only 10 postcode regions account for up to 80% of the total overconsumption due to identified anomalies. This finding highlights the effectiveness of our approach in identifying critical areas within the urban water network.

VI. Conclusions

The proposed framework represents a notable advancement in water resource management. By integrating LSTM networks with Transformer encoder-decoder architectures, the proposed model effectively addresses the complexities of detecting anomalies in variable time-series data. Its adaptability to different data configurations ensures strong performance, even with incomplete or partially missing data.

The study in Barcelona revealed that anomalies were spatially concentrated, with one suburb responsible for 5% of total anomalies and 10 postcodes accounting for 80% of overconsumption anomalies. The hierarchical model enables near real-time anomaly detection, beneficial for large-scale urban water networks, and supports multiple layers of parallelization, optimizing inference efficiency. It remains reliable despite data gaps and captures both short and long-term water usage trends. Its customizability allows fine-tuning for different urban settings by adjusting stages, window sizes, and the number of LSTM models for better accuracy and contextual understanding.

Future research could incorporate diverse datasets, such as socioeconomic and environmental data, to enhance the model's anomaly detection accuracy. Investigating various configurations for the hierarchical model could improve its performance and generalizability. Additionally, field tests in various urban environments could further validate the framework's robustness and effectiveness.

In conclusion, this framework presents an effective unsupervised anomaly detection solution for urban water networks, with the potential for broad adoption and impact on sustainable water management.

VII. Acknowledgment

The authors want to thank Aigües de Barcelona for providing the datasets for this work. Additionally, we would like to extend our gratitude to our colleague, Oriol Soler, for his invaluable contributions and dedication throughout the project.

References

[1] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi, "Deep learning for time series anomaly detection: A survey," *ACM Comput. Surv.*, aug 2024. Just Accepted.

[2] K.-L. Li, H.-K. Huang, S.-F. Tian, and W. Xu, "Improving one-class svm for anomaly detection," vol. 5, pp. 3077–3081 Vol.5, 2003.

[3] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," pp. 413–422, 2008.

[4] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, "Robust random cut forest based anomaly detection on streams," in *Proceedings of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proceedings of Machine Learning Research*, (New York, New York, USA), pp. 2712–2721, PMLR, 20–22 Jun 2016.

[5] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, vol. 54, p. 1–38, Mar. 2021.

[6] S. Lin, R. Clark, R. Birke, S. Schonborn, N. Trigoni, and S. Roberts, "Anomaly detection for time series using vae-lstm hybrid model," pp. 4322–4326, 05 2020.

[7] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[8] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," 2022.

[9] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, (New York, NY, USA), p. 93–104, Association for Computing Machinery, 2000.

[10] V. Hautamaki, I. Karkkainen, and P. Franti, "Outlier detection using k-nearest neighbour graph," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3, pp. 430–433 Vol.3, 2004.

[11] M. Raciti, J. Cucurull, and S. Nadjm-Tehrani, *Anomaly Detection in Water Management Systems*, pp. 98–119. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[12] M. Fagiani, S. Squartini, L. Gabrielli, M. Severini, and F. Piazza, "A statistical framework for automatic leakage detection in smart water and gas grids," *Energies*, vol. 9, no. 9, 2016.

[13] E. Kermany, H. Mazzawi, D. Baras, Y. Naveh, and H. Michaelis, "Analysis of advanced meter infrastructure data of water consumption in apartment buildings," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, (New York, NY, USA), p. 1159–1167, Association for Computing Machinery, 2013.

[14] L. A. Passos Júnior, C. C. Oba Ramos, D. Rodrigues, D. R. Pereira, A. N. de Souza, K. A. Pontara da Costa, and J. P. Papa, "Unsupervised non-technical losses identification through optimum-path forest," *Electric Power Systems Research*, vol. 140, pp. 413–423, 2016.

[15] *Detecting Anomalies in Water Quality Monitoring Using Deep Learning*, vol. Day 2 Wed, March 06, 2024 of *SPE Water Lifecycle Management Conference and Exhibition*, 03 2024.

[16] E. El-Shafeiy, M. Alsabaan, M. Ibrahem, and H. Elwahsh, "Real-time anomaly detection for water quality sensor monitoring based on multivariate deep learning technique," *Sensors*, vol. 23, p. 8613, 10 2023.

[17] S. Russo, A. Disch, F. Blumensaat, and K. Villez, "Anomaly detection using deep autoencoders for in-situ wastewater systems monitoring data," 2020.

[18] S. Qin, Y. Lang, and K.-P. Chow, *Traceable Transformer-Based Anomaly Detection for a Water Treatment System*, pp. 219–234. 10 2023.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[20] C. Ding and B. Jiang, "L1-norm error function robustness and outlier regularization," 2017.