Intelligent Systems with Applications 26 (2025) 200504

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

An effective approach for early fuel leakage detection with enhanced explainability

Ruimin Chu [a,*], Li Chik [b], Yiliao Song [c], Jeffrey Chan [a], Xiaodong Li [a]

[a] School of Computing Technologies, RMIT university, Melbourne, 3000, Australia
[b] Titan Cloud Software, Carrum Downs, 3201, Australia
[c] School of Computer and Mathematical Sciences, University of Adelaide, Adelaide, 5005, Australia

ARTICLE INFO

Keywords:
Explainable AI (XAI)
Fuel leakage detection
Online change point detection
Neuro fuzzy system
Deep learning

ABSTRACT

Leakage detection at service stations with underground storage tanks containing hazardous products, such as fuel, is a critical task. Early detection is important to halt the spread of leaks, which can pose significant economic and ecological impacts on the surrounding community. Existing fuel leakage detection methods typically rely on statistical analysis of low-granularity inventory data, leading to delayed detection. Moreover, explainability, a crucial factor for practitioners to validate detection outcomes, remains unexplored in this domain. To address these limitations, we propose an **EX**plainable **F**uel **L**eakage **D**etection approach called EXFLD, which performs online fuel leakage detection and provides intuitive explanations for detection validation. EXFLD incorporates a high-performance deep learning model for accurate online fuel leakage detection and an inherently interpretable model to generate intuitive textual explanations to assist practitioners in result validation. Unlike existing explainable artificial intelligence methods that often use deep learning models which can be hard to interpret, EXFLD is a human-centric system designed to provide clear and understandable insights to support decision-making. Through case studies, we demonstrate that EXFLD can provide intuitive and meaningful textual explanations that humans can easily understand. Additionally, we show that incorporating semantic constraints during training in the ANFIS model enhances the semantic interpretability of these explanations by improving the coverage and distinguishability of membership functions. Experimental evaluations using a dataset collected from real-world sites in Australia, encompassing 167 tank instances, demonstrate that EXFLD achieves competitive performance compared to baseline methods, with an F2-score of 0.7969. This dual focus on accuracy and human-centric explainability marks a significant advancement in fuel leakage detection, potentially facilitating broader adoption.

1. Introduction

Fuel service stations with petroleum products stored in Underground Storage Tanks (USTs) must be protected against leakage. The timely detection of leakage is crucial for preventing the spread of contamination and protecting the surrounding environment and human health. Despite best efforts, spills and leaks from USTs are sometimes unavoidable. As of Sep 2023, more than 573,000 confirmed cases of petroleum and hazardous substance leaks from USTs have been reported in the USA, with 4354 of those confirmed leaks between Oct 2022 and Sep 2023 (United States Environmental Protection Agency, 2023). USTs containing hazardous fluids pose significant risks to the environment, leading to the loss of plant and animal life, as well as the spread of various diseases within human populations (Environmental Protection Agency Victoria, 2018).

Operators of USTs are mandated to use at least one of the approved leak detection methods in compliance with regulations. According to the United States Environmental Protection Agency (United States Environmental Protection Agency, 2024), there are three main categories of leak detection methods: interstitial method, which conducts interstitial monitoring with secondary containment such as vacuum or liquids between walls; internal method that relies on Statistical Inventory Reconciliation (SIR), and external method that monitors external sources such as vapours and liquids. This work focuses on the SIR-based method (United States Environmental Protection Agency, 2019b), which is a data-driven approach that conducts statistical analysis on inventory log data collected through Automatic Tank Gauges (ATGs). SIR can perform leak detection for USTs automatically and

* Corresponding author.
E-mail addresses: s3912230@student.rmit.edu.au (R. Chu), li.chik@titancloud.com (L. Chik), lia.song@adelaide.edu.au (Y. Song), jeffrey.chan@rmit.edu.au (J. Chan), xiaodong.li@rmit.edu.au (X. Li).

https://doi.org/10.1016/j.iswa.2025.200504
Received 7 November 2024; Received in revised form 4 March 2025; Accepted 9 March 2025
Available online 17 March 2025

remotely, which makes it a low-cost method that eliminates the need for technicians to travel to sites. This is especially beneficial for sites located in remote areas. Previous research on SIR (Alayón, Sigut, Arnay, & Toledo, 2020; Gorawski, Gorawska, & Pasterak, 2017; Li, Shui, Luo, Chen, & Li, 2011) has introduced autonomous solutions, such as using classifiers or trend detection methods, for fuel leakage detection.

However, existing SIR-based solutions have two main limitations: they cannot perform early detection or provide sufficient explanations. Firstly, most existing works (Gill, Keating, & Baron, 2006; Gorawski et al., 2017; Keating & Mason, 2000; Li et al., 2011; Sigut, Alayón, & Hernández, 2014) conduct offline analysis on a month of tank inventory data, leading to detection delays as decisions can only be made after collecting this amount of data. Only a few recent studies consider the early detection aspect and propose to analyse data over days (Alayón et al., 2020) or conduct real-time analysis for early detection (Chu, Chik, Song, Chan, & Li, 2024). The other limitation is that these works either use black-box models or regression methods that provide none or minimal explanations, i.e. the estimated leak rate. In practice, false alarms are unavoidable. If getting too many false alarms, practitioners may become desensitised and less responsive to genuine alarms. On the other hand, without adequate explanations, practitioners cannot understand what causes the model to determine if there is a leakage and whether they should trust the result. Thus, there is an urgent need to develop a method that provides explanations, enabling practitioners to effectively comprehend detection outcomes and validate results while being accurate in detection to ensure as few false alarms as possible.

In recent years, eXplainable AI (XAI) has garnered significant attention. The goal is to develop explainable Machine Learning (ML) solutions that foster human comprehension, trust, and effective management of emerging AI systems (Arrieta et al., 2020; Gunning et al., 2019). A potential solution to address the limitations identified in the previous paragraph is to use an explainable real-time Deep Learning (DL) detection model that provides both high performance and explainability. However, most state-of-the-art XAI DL-based methods applied to time series problems are too complex for humans to understand (Rojat et al., 2021). This complexity may arise from the inherently unintuitive nature of time series data, unlike images or natural language, which can be grasped easily (Siddiqui, Mercier, Munir, Dengel, & Ahmed, 2019; Theissler, Spinnato, Schlegel, & Guidotti, 2022). While attention-based or attribute-based explanations through means of visualisations, shapelets or prototypes are widely used in XAI methods for time series analysis, these artefacts may not indeed be easily understood by users, particularly when the pure signal is not directly comprehensible (Rojat et al., 2021; Theissler et al., 2022). Current XAI solutions primarily focus on *enabling* explainability, e.g. allowing the extraction of relevant information from complex ML models. However, they often fall short in offering explanations that are intuitive and *meet the needs* of users, i.e. clarifying whether their explanations are suitable for the target users in the domain-specific context. Addressing this gap is crucial for enhancing the practical utility and user satisfaction of XAI systems in our problem.

In a nutshell, we aim to tackle two major research questions for data-driven fuel leak detection in USTs:

1. *How can we design a fuel leak detection method that provides explanations effectively explaining fuel leakage phenomena to practitioners?* Specifically, we seek a local explanation with great interpretability for individual predictions, making it easily understandable for practitioners who may not be familiar with the internal decision-making processes of ML algorithms.
2. *How can we achieve great accuracy performance for early leak detection while maintaining high explainability?* It is acknowledged that accuracy/performance and explainability are often in conflict with ML techniques (Adadi & Berrada, 2018; Fernandez, Herrera, Cordon, del Jesus, & Marcelloni, 2019). Enhancing explainability without compromising accuracy remains a challenge.

To address these questions, in this paper, we propose an **EX**plainable **F**uel **L**eakage **D**etection approach, EXFLD, which comprises two specialised modules to excel in their respective areas rather than compromising one property for the other with a single model. Specifically, EXFLD consists of an *online detection module* employing a sophisticated DL model to achieve accurate early fuel leakage detection and an *explanation module* to provide intuitive explanations. The explanation module utilises an inherently interpretable model structured based on domain knowledge to ensure high interpretability. These two modules work together with the detection module continuously monitoring the input data streams to perform online leakage detection, while the explanation module is activated when needed—primarily upon a leak detection or user request. When activated, the explanation module references and processes historical data to construct meaningful, domain-specific features, ultimately generating intuitive textual explanations to elucidate changes in the data properties.

For EXFLD, we employ Temporal Fusion Transformer (TFT) (Lim, Arık, Loeff, & Pfister, 2021), which is a high-performance attention-based multi-horizon forecasting model, in the *online detection module*. It detects changes in data properties that indicate a leak by evaluating the dissimilarity score derived from the TFT's estimated and anticipated results, signalling a leak when this score exceeds a set threshold. Additionally, TFT can provide visualised explanations to end users. We employ Adaptive Neuro-Fuzzy Inference System (ANFIS) (Jang, 1993), which is a fuzzy rule-based system, in the *explanation module* to provide human-understandable textual explanations. Its decision-making process is articulated through IF-THEN rules, and linguistic labels are used to describe the fuzzy variables converted from input features, which can be used to generate textual explanations for direct human interpretation. Through case studies on real-world fuel data with simulated leakage, we demonstrate how EXFLD provides intuitive and effective explanations to elucidate the leakage. Additionally, experimental results on the fuel leakage dataset confirm that EXFLD achieves competitive performance compared to baselines in terms of accuracy. The contributions of the paper include:

- To the best of our knowledge, no work has explored the explainability of its method in the domain of fuel leakage detection. This is the first work to attempt to propose a method that provides explainability in this field.
- We propose an explainable fuel leakage detection approach, EXFLD, which incorporates both a DL model for its high accuracy performance and an inherently interpretable model to generate intuitive textual explanations. This addresses the challenge of jointly achieving a high level of explainability and accuracy, particularly for a time series problem.
- We demonstrate EXFLD's explainability for the fuel leakage detection problem through case studies, offering intuitive textual explanations that use linguistic terms to describe variations in fuel variances across different periods. We further show how semantic constraints in ANFIS training enhance the interpretability of local explanations by ensuring broad coverage and distinguishability of membership functions.
- The experimental results confirm the effectiveness of EXFLD in fuel leak detection, which outperforms other comparable baselines on our fuel leakage data in terms of detection accuracy.

The remainder of this paper is structured as follows: We review the related literature in Section 2. In Section 3, we present the problem formulation and detail the structure of EXFLD. The interpretability and experimental outcomes of EXFLD are shown in Section 4. Finally, conclusions and future work are presented in Section 5.

2. Background and related work

In this section, we first review existing SIR-based approaches for fuel leakage detection. Next, we review relevant techniques for fuel leakage

detection problems, including change point detection and forecasting with exogenous series. Finally, we review relevant XAI works in the context of our study.

2.1. Fuel leakage detection with statistical inventory reconciliation methods

One of the primary leak detection methods for USTs is SIR, which is a volumetric and statistical-based approach (United States Environmental Protection Agency, 2019a). SIR generally utilises a month of inventory records for statistical analysis of discrepancies. There are two primary approaches in the literature for SIR methods. The first type typically aims to estimate leak rates using regression-based models (Gill et al., 2006; Keating & Mason, 2000; Li et al., 2011). For example, in the work of Keating and Mason (2000), regression-based models are constructed where intercepts denote the sizes of leaks. Due to the nature of regression models, these methods generally only provide an estimated leak rate upon detection. In the TUBE algorithm (Gorawski et al., 2017), a trend interpretation stage is incorporated to provide qualitative results such as tight, leak, inconclusive, invalid, and the estimated leak rate. The other group of works employs classifiers (Alayón et al., 2020; Sigut et al., 2014; Toledo, Arnay, Hernández, Sigut, & Alayón, 2024), using feature vectors extracted from daily inventory data (e.g. cumulative variance, variance over sales) to classify days as either "with leaks" or "without leaks". The early detection aspect is studied in Alayón et al. (2020) by formulating the problem as labelling feature sets of various operating days as "with leaks" or "without leaks". However, the study is not conducted in an online setting, and its accuracy performance depends on the data from a greater number of operating days. The experiments of Alayón et al. (2020), Sigut et al. (2014), Toledo et al. (2024) primarily focus on feature selection and determining the optimal classifier-feature combinations, but there is a lack of transparency in the classifier's decision-making process. Recently, the authors of Chu et al. (2024) have explored online Change Point Detection (CPD) on real-time fuel variance sequential data, enabling the early detection of fuel leakage with a detection delay of less than 7 days.

Although a few existing works consider the aspect of early detection, no works so far have addressed explainability. In particular, existing methods provide limited or no insights into the reasoning behind leakage detection, posing barriers to practitioners' validation and trust in these predictions. Methods with explainability or interpretability have the potential to overcome this issue and thus can serve as a crucial step towards establishing trustworthy AI in this field.

2.2. Fuel leakage detection using CPD with multi-stream inputs

Change point detection (CPD) refers to the problem of identifying abrupt changes in time series data where its properties have changed (Kawahara & Sugiyama, 2012). As a fuel leakage would lead to a change in the distribution of fuel variance, which is the key variable that is commonly used in SIR-based methods, CPD is a suitable method for fuel leakage detection. CPD approaches can be categorised based on criteria such as offline or online deployment (some studies refer to this as retrospective and sequential) and univariate or multivariate. Online algorithms operate in real-time, inferring change points as new data arrives, while offline algorithms conduct the detection retrospectively based on the entire historical dataset (van den Burg & Williams, 2022; Truong, Oudre, & Vayatis, 2020). Univariate change point detection involves identifying change points in a single sequence of independent observations (Wang, Yu, & Rinaldo, 2020). While multivariate CPD may be more complicated with different scenarios, most works focus on cases where the change point affects all coordinates of the series (homogeneous series) (Cho, 2016; Matteson & James, 2014) or for an unknown subset of the coordinates (sparse/heterogeneous series) (Cho & Fryzlewicz, 2015; Guo, Gao, & Lu, 2022; Wang & Samworth, 2018). More recent works (Alanqary, Alomar, & Shah, 2021; Knoblauch & Damoulas, 2018) focus on changes in spatio-temporal models, detecting changes in both spatial and temporal structures.

As mentioned, leakage can be detected based on the key variable, fuel variance. On the other hand, fuel variance can also be influenced by exogenous variables such as tank temperature. Including these exogenous variables when analysing the fuel variance would be helpful, thus resulting in multi-channel inputs. However, applying the aforementioned mainstream multivariate CPD methods to our problem may lead to the detection of change points in exogenous sequences, which are irrelevant to leakage (e.g. leakage should not be directly related to changes in temperature), resulting in false alarms.

Meanwhile, time series forecasting with exogenous variables has received much research attention in the last two decades. The nonlinear autoregressive exogenous (NARX) model (Lin, Horne, Tino, & Giles, 1996) was introduced to predict the current value of a target time series not only based on its previous values but also on exogenous series. Various approaches have been explored in this field (Gao & Er, 2005; Liu, Gong, Yang, & Chen, 2020; Lu, Han, Sun, & Yang, 2024; Qin et al., 2017). However, this problem setup and its applications in anomaly detection or CPD remain rare. In ML-based CPD, dissimilarity scores can be obtained through predictive errors. This makes forecasting with exogenous variables algorithms a good candidate, aligning with our problem characteristics. Specifically, this type of algorithm considers the information from exogenous variables when predicting the fuel variances. A change point is detected when the prediction deviates from the ground truth, indicating a change in the data distribution in the target sequence of fuel variance.

Among various algorithms, we specifically consider a state-of-the-art algorithm, TFT (Lim et al., 2021), for its compatibility with our problem and advantages. First, TFT enables multi-horizon forecasting with high performance. It can make estimations at multiple steps in the future and thus facilitate the detection of permanent changes rather than single outliers. To enhance performance, components including static covariate encoders, gating mechanisms and variable selection networks are integrated into the architecture. These constituents enable the model to accommodate not only exogenous series but also static variables, provide adaptive network complexity and choose relevant input variables at each time step. Finally, the model facilitates interpretability through visualisation, aiding in identifying important variables and observing temporal patterns. Most models are not able to provide all the functions mentioned above. Multiple recent works have utilised TFT for various practical, real-world applications, such as wind speed forecasting (Wu, Wang, & Zeng, 2022) and load forecasting (Li, Tan, Zhang, Miao, & He, 2023), particularly due to the model's ability to enhance interpretability, which reinforces our choice of this model.

2.3. Explainable AI

In this section, we first review current XAI approaches for time series and leakage issues. Next, we discuss ANFIS, a transparent rule-based model enabling intuitive explanations, which we choose to adopt for our problem.

2.3.1. XAI in multivariate time series problems

In recent years, there have been an increasing number of works incorporating explainability into time series problems. BeatGAN (Zhou, Liu, Hooi, Cheng, & Ye, 2019) is a GAN-based method proposed to detect anomalies for multivariate time series data. It uses heatmaps with different colours to indicate anomalous scores for visualisation, allowing users to pinpoint the time ticks of anomalies. A model-agnostic technique of series saliency maps is proposed in Pan, Hu, and Chen (2021) to consider the time and feature dimensions coherently. The authors present the interpretability of the method by visualising the learned mask components and comparing them to the original time series data. In Deng and Hooi (2021), structure learning with GNN and attention weights are utilised to provide explainability for the

detected anomalies. They present the interpretability of the model through sensor embeddings to represent the similarity between the sensor behaviours, learned graph edges to indicate sensor relationships, and attention weights to highlight the importance of each node's neighbours. Actionable Forecasting Network (AFN) (Jagirdar, Talwadker, Pareek, Agrawal, & Mukherjee, 2024) is a deep neural network proposed for multivariate time series forecasting problems while enabling explainability. The reasoning of forecasting is provided in the form of heatmaps, which support the identification of significant time steps and features. A recent work (Kacprzyk, Liu, & van der Schaar, 2024) introduces TIMEVIEW, a transparent time series forecasting model that is complemented with an interactive visualisation tool for interpretability. This tool facilitates the examination of both high-level and low-level features of predicted trajectories, as well as their changes in response to inputs.

Throughout the review, we have observed a growing trend towards enhancing explainability in DL-based models. Explanations are generally presented through visualisations, shapelets or prototypes (Theissler et al., 2022). However, unlike images or natural language, these artefacts are not directly interpretable (Rojat et al., 2021) and their interpretability for the target audience is not regularly assessed (Theissler et al., 2022). These methods are often evaluated in terms of generality across different problems but lack insightful analysis specific to issues, making it challenging for people without a strong background to understand. While general applicability is desirable, some scenarios may benefit more from domain-specific explanations when presented to end users.

In summary, while recent works have made progress in incorporating explainability into time series problems, many approaches prioritise generality over practical interpretability. This leaves a major gap in providing domain-specific, intuitive explanations that align with practitioners' needs. Addressing this issue is essential for fostering trust and enabling effective use of explainable AI in real-world applications.

2.3.2. XAI in leakage problems

In this section, we review some recent applications of XAI in leakage-related problems. Xu et al. (2022) propose an explainable ensemble tree model for detecting water pipe leakages based on vibration signals. Shapley Additive Explanation (SHAP) method is employed to provide interpretations of the XGBoost model, aiding in identifying typical features crucial for distinguishing various leakage states. In Liu et al. (2023), the authors introduce a transfer learning-based explainable diagnosis method based on two-dimensional class activation maps and dynamic time warping for fault diagnosis at oil-gas treatment stations. The method provides auxiliary tools for fault reasoning by pinpointing the variables that have a greater impact on the fault condition at specific periods. In Gemeinhardt and Sharma (2023), deep feature modelling is applied to perform leakage detection and localisation based on both distributed acoustic sensor and distributed temperature sensor data. The image segmentation of predicted leaks highlights the exact segment of data that triggered the alarm, facilitating human analysis.

However, explanations through visualisations such as SHAP or saliency maps do not necessarily build users' trust or help users validate the model's predictions. SHAP highlights feature importance, showing the most typical identification features for each leakage state. While this can uncover new insights into the problem, it may not be useful for practitioners in validating results or conveying the model's trustworthiness. Class activation maps that identify critical time segments face the same issue: users may not trust the results because they struggle to understand these segments and the rationale.

Other works (Mounce, Boxall, & Machell, 2010; Silva, Veloso, & Gama, 2023) consider generating descriptive rules to support root cause analysis. For example, in Silva et al. (2023), the authors utilise a DL model with an explainability layer to detect failures in air production units, including issues such as oil leaks. The explainability layer involves a Chebyshev sampling strategy (Aminian, Ribeiro, & Gama, 2021) and the Adaptive Model Rules (AMRules) (Duarte, Gama, & Bifet, 2016) to generate descriptive rules. While these descriptive rules provide a clearer rationale compared to previous methods, they often rely on real numbers (e.g. "If H1 sensor is at or below 8.8 bar"). Such explanations remain insufficient for naive users to understand whether this value is abnormal. Moreover, in scenarios where the normal/abnormal criteria must be established based on different entity properties or specific contexts, this explanation becomes even less helpful. For example, an explanation like "If the temperature is at 25 degrees" might still leave users unsure whether this temperature is high or not, as the level of this temperature can vary based on different countries or regions.

2.3.3. Inherently interpretable models with better explainability

It is acknowledged that transparent ML models that are inherently interpretable, such as decision trees and (fuzzy) rule-based learning models, have higher interpretability compared to black-box models with complex structures (Arrieta et al., 2020; Gunning et al., 2019; Theissler et al., 2022). Among these transparent ML models, fuzzy rule-based systems can empower more understandable models as they enable textual-based explanation by using linguistic terms to represent variables. Traditionally, fuzzy inference systems are constructed based on expert knowledge; however, in most real-world situations, it is improbable that an expert can understand the complete behaviour of the data. This is also true in our case, as experts do not possess comprehensive knowledge of the behaviours of all the tanks or sites. Neuro-fuzzy modelling approaches have been introduced to address this limitation by utilising empirical data to derive fuzzy rules by training adaptive networks. Their design approximates human reasoning, allows understanding of the network's inferred outcomes and effectively manages imprecision and uncertainty.

In this work, we use Adaptive Network-based Fuzzy Inference System (ANFIS) (Jang, 1993) to generate intuitive explanations that support practitioners in validating the detection of leakage and enhancing trust in the system. ANFIS is a fuzzy inference system implemented in the framework of adaptive networks. It has been used for explainability in various real-world scenarios, such as elucidating decisions made by unmanned aerial vehicle systems (Keneni et al., 2019), explaining the decision for cancer diagnosis (Nguyen, Kavuri, Park, & Lee, 2022), real-world regression problems (Pramod & Pillai, 2021) and predicting performance of solar ground source heat pump systems (Hikmet Esen & Ozsolak, 2017). There is also work that uses ANFIS for leak detection, such as Cristello, Dang, Hugo, and Park (2024), but this research primarily leverages the model's transparency to facilitate analysis rather than providing explanations alongside the detection results. Through a hybrid learning procedure, ANFIS refines fuzzy IF-THEN rules to determine optimal values that best describe the input–output relationship of a complex system. Its fuzzification layer maps real-numbered inputs to degrees of membership in fuzzy sets defined by Membership Functions (MFs). These fuzzy sets can then be assigned with linguistic labels, enabling natural knowledge representation. For example, if the input variable is inventory height, the fuzzy sets could be "Low", "Medium" and "High" with the MFs determining the extent to which a specific inventory height belongs to each category. Thus, ANFIS can generate textual explanations that justify why a fuel leakage may be detected using selected linguistic terms that practitioners can easily understand.

Based on the literature discussed in the previous sections, Table 1 summarises the current state of research and research gaps. As shown in the table, no existing paper addresses the combination of real-time fuel leakage detection with multi-stream input handling and comprehensive leakage result explanation, which forms the core motivation for our research.

Table 1
Summary of literature review and research gaps.

Work	Fuel leakage detection	Handle multi-stream inputs	Model interpretation	Leakage result explanation
SIR-based with statistical methods (Gill et al., 2006; Gorawski et al., 2017; Keating & Mason, 2000; Li et al., 2011)	✔, Offline	✗	✔, but limited	✗
SIR-based with ML methods (Alayón et al., 2020; Sigut et al., 2014; Toledo et al., 2024)	✔, Offline	✗	✗	✗
Fuel leakage detection via online CPD (Chu et al., 2024)	✔, Real-time	✗	✗	✗
XAI for multivariate time series problems (Deng & Hooi, 2021; Jagirdar et al., 2024; Kacprzyk et al., 2024; Pan et al., 2021; Zhou et al., 2019)	✗	✔	✔	✔, Mostly graph-based and unintuitive
XAI for leakage problems (Gemeinhardt & Sharma, 2023; Liu et al., 2023; Silva et al., 2023; Xu et al., 2022)	✗	✔	✔	✔, Limited and unintuitive
Ours	✔, Real-time	✔	✔	✔, Both graph-based and textual explanations

3. Methodology

In this section, we first formulate the problem of online fuel leakage detection and explanation. Next, we provide an overview of EXFLD. Then, we explain the structure and key components of TFT and ANFIS and how they are employed in EXFLD to conduct leak detection and provide explanations.

3.1. Problem formulation

In this paper, we aim to detect fuel leakage events as early as possible and explain this detection based on sequentially observed inventory data and static fuel tank information. The explanations are designed not for the model designer but mainly for fuel site practitioners, providing non-technical descriptions that help them infer why the leakage is detected. No existing methods can address the detection and the required explanation task at the same time. We formulate this problem as a two-step task where the fuel leakage detection is formulated as an online CPD problem while the explanation is a rule-based modelling process that outputs text-based rules. To formulate this two-step problem, we define that each tank entity i has known static features s_i (tank id, tank maximum height, tank maximum volume), sequential exogenous inputs $x_{i,0:T}$ (time of day, inventory height, inventory volume, tank temperature) from period 0 to T and target series, i.e. fuel variance $y_{i,0:T}$ as the input of the system. The fuel leakage detection aims to output the change point κ such that before κ, fuel variance samples are i.i.d. with a distribution P, and after κ, samples are i.i.d. with a different distribution Q. Additionally, explanations for detection results are also produced to explain why leakage is identified, i.e. to provide plots comparing prediction against anticipated results, attention weight patterns that indicate important past timesteps contribute to the prediction decision, the importance ranking of variables and textual explanations in the forms of IF-THEN rules which describes the degree of meaningful features for the detection results.

3.2. EXFLD overview

To address the online fuel leakage detection and explanation challenges, we propose EXFLD depicted in Fig. 1. First, we preprocess the data and train TFT with normal inventory data to predict target future variances based on past reference data and exogenous variables. ANFIS is trained with both normal and simulated leakage data, to differentiate normal and leakage cases and optimise MFs. During online detection (Algorithm 1), the incoming data stream is preprocessed and passed to the trained TFT to obtain predictive future variances, which are then compared to the actual variances by the discriminator to obtain the dissimilarity scores. A leakage is identified when the score exceeds the threshold. The explanation module with ANFIS (Algorithm 2) is activated primarily when a leakage is detected or when the user requests. The module collects and processes the past 30 days of inventory data to generate meaningful features such that an intuitive textual explanation can be produced. Finally, these textual explanations, along with the visualised explanations generated by TFT, are provided to end-users for confirmation of variance changes indicative of a leakage.

Algorithm 1 Online Detection with TFT.

Input: static feature s_i, sequential exogenous series $x_{i,0:T}$ and fuel variance series $y_{i,0:T}$
Output: Change points that indicate fuel leakage
while there is a data point $t+\tau$ to proceed **do**
  Preprocess incoming data (Outlier removal & Normalisation) ▷ refer to Section 3.3
  Compute estimated fuel variance $f(y_{i,t-k:t}, x_{i,t-k:t+\tau}, s_i)$ with trained TFT ▷ refer to Eq. (8)
  Compute dissimilarity score $f_s(\hat{y}_{i,t:t+\tau}, y_{i,t:t+\tau})$ ▷ refer to Eq. (9)
  Calculate threshold ψ ▷ refer to Eq. (10)
  if Dissimilarity score > Dynamic threshold **then**
    Leakage detected
  end if
end while

3.3. Data pre-processing

Data preprocessing is performed to alleviate the influence of noise and prepare the data for analysis. For the fuel leakage problem, factors such as measurement errors can impact the quality of the data (Gorawski et al., 2017). Outlier removal is a technique to minimise disturbances caused by outliers. As we perform detection in an online manner, where full information is not available upfront, we consider the real-time local outlier removal strategy. Singular Spectrum Analysis (SSA), used in previous studies (Chu et al., 2024; Gupta, Wadhvani, & Rasool, 2022; Lu, Kumar, Collier, Krishna, & Langston, 2018), performs data reconstruction and calculates residuals by measuring the difference between the original signal and its reconstruction. These residuals are dynamically monitored, and a dynamic threshold is established following the three-sigma rule, where any point with a residual exceeding the threshold is identified as an outlier. Detected outliers are then replaced with an imputed value, calculated as the mean of

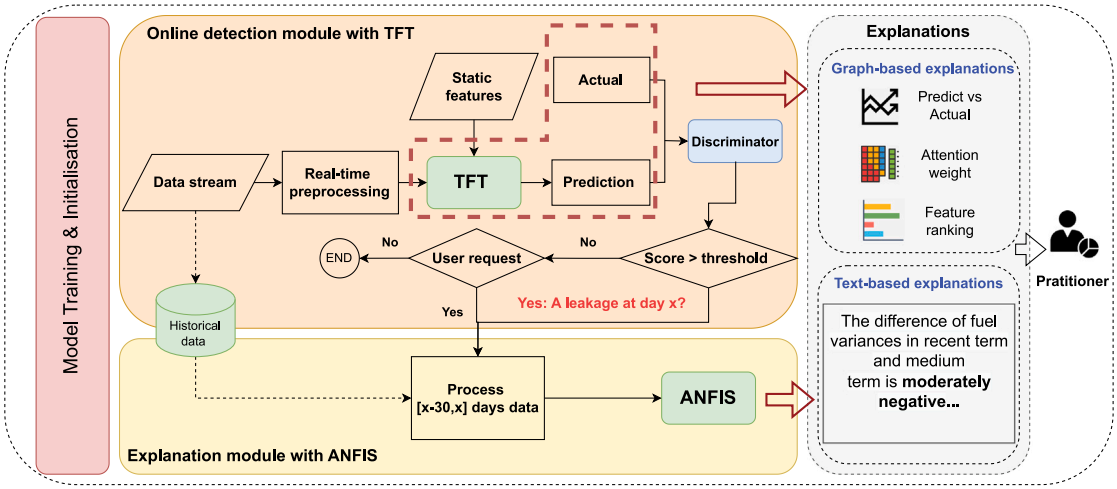


Fig. 1. Overview of EXFLD.

Algorithm 2 Explanation Module with ANFIS.

Input: Inventory data in the last 30 days
Output: Textual explanation

Step 1: Retrieve and preprocess inventory data
Retrieve inventory data from the past 30 days
Partition the data into periods of {Recent, Medium Term, Long Term} based on predefined fuzzy sets
Compute aggregated features for each period
Derive differences of fuel variance across different periods: $\Delta\left(y_{recent}, y_{medium}\right)$, $\Delta\left(y_{recent}, y_{long}\right)$
Step 2: Generate textual explanations with ANFIS
Input the features obtained from Step 1 into the trained ANFIS model
Retrieve top-firing IF-THEN rules from the ANFIS inference process
Translate the identified rule into linguistic explanations in the format:

"IF (variable) is (linguistic term for the fuzzy set) AND ... THEN (class)".

Generate detailed feature descriptions, including membership degrees:

"(variable) is (linguistic term 1) with a membership of (value 1) and (linguistic term 2) with a membership of (value 2) and ...".

Aggregate all explanations to provide the reasoning for leakage detection

the previous window of size 5. In addition to outlier removal, data normalisation is applied to inventory volume and height by dividing each by its respective maximum value, which can be known in advance, ensuring standardised data representation.

3.4. Online detection with TFT

In this section, we first explain some key components of TFT and then discuss how TFT is applied to our problem to perform leakage detection.

3.4.1. Temporal fusion transformers

TFT is an attention-based DL model for multi-horizon forecasting that can handle exogenous inputs and static metadata while also facilitate interpretability. The architecture of TFT is illustrated in Fig. 2. It is composed of the following main constituents:

1. **Gating mechanisms:** There are challenges in foreseeing the relevance of variables due to unknown relationships between exogenous inputs and targets as well as in gauging the extent of non-linear processing. TFT alleviates these challenges by proposing Gated Residual Networks (GRNS), allowing the selective application of non-linear processing. The GRN accepts a primary input a and an optional context vector c and performs:

$$\begin{aligned} GRN_{\omega}(a,c) &= LayerNorm(a + GLU_{\omega}(\eta_1)), \\ \eta_1 &= W_{1,\omega}\eta_2 + b_{1,\omega}, \\ \eta_2 &= ELU(W_{2,\omega}a + W_{3,\omega}c + b_{2,\omega}), \end{aligned} \tag{1}$$

where ELU (Exponential Linear Unit) is used as an activation function, η_1 and η_2 are intermediate layers, and ω is an index to denote weight sharing. Gated Linear Units (GLUs) take the form:

$$GLU_{\omega}(\gamma) = \phi_s(W_{4,\omega}\gamma + b_{4,\omega}) \odot (W_{5,\omega}\gamma + b_{5,\omega}), \tag{2}$$

where $\phi_s(.)$ is the sigmoid activation function, $W(.)$ and $b(.)$ represent the weights and biases, $\odot$ denotes the element-wise Hadamard product.

2. **Variable Selection Networks:** Although there may be multiple available variables, not all of them are relevant to the output. To tackle this issue, TFT employs instance-wise variable selection networks for both static and time-dependent covariates, enabling the model to ignore unnecessary noisy inputs and offer insights into the importance of variables. At time t, $\xi_t^{(n)}$ represents the transformed input of the nth variable and $\Xi_t = [\xi_t^{(1)^T}, \ldots, \xi_t^{(m_\chi)^T}]^T$ denotes the flattened vector of previous inputs. As shown in Eq. (3), variable selection weights υ_{χ_t} are derived by passing Ξ_t and context vector c_s through GRN. $\tilde{\xi}_t^{(n)}$ is the feature vector for variable n, generated through non-linear processing via its own GRN. Finally, the processed features are combined after being weighted by their corresponding υ_{χ_t} as depicted in Eq. (5).

$$\upsilon_{\chi_t} = Softmax(GRN_{\upsilon_\chi}(\Xi_t, c_s)), \tag{3}$$

$$\tilde{\xi}_t^{(n)} = GRN_{\tilde{\xi}(n)}(\xi_t^{(n)}), \tag{4}$$

$$\tilde{\xi}_t = \sum_{n=1}^{m_\chi} \upsilon_{\chi_t}^{(n)} \tilde{\xi}_t^{(n)}, \tag{5}$$

3. **Static Covariate Encoders:** TFT integrates static features by using separate GRN encoders to generate different context vectors. These context vectors serve various purposes, such as selecting temporal variables, locally processing temporal features and enhancing temporal features with static information.

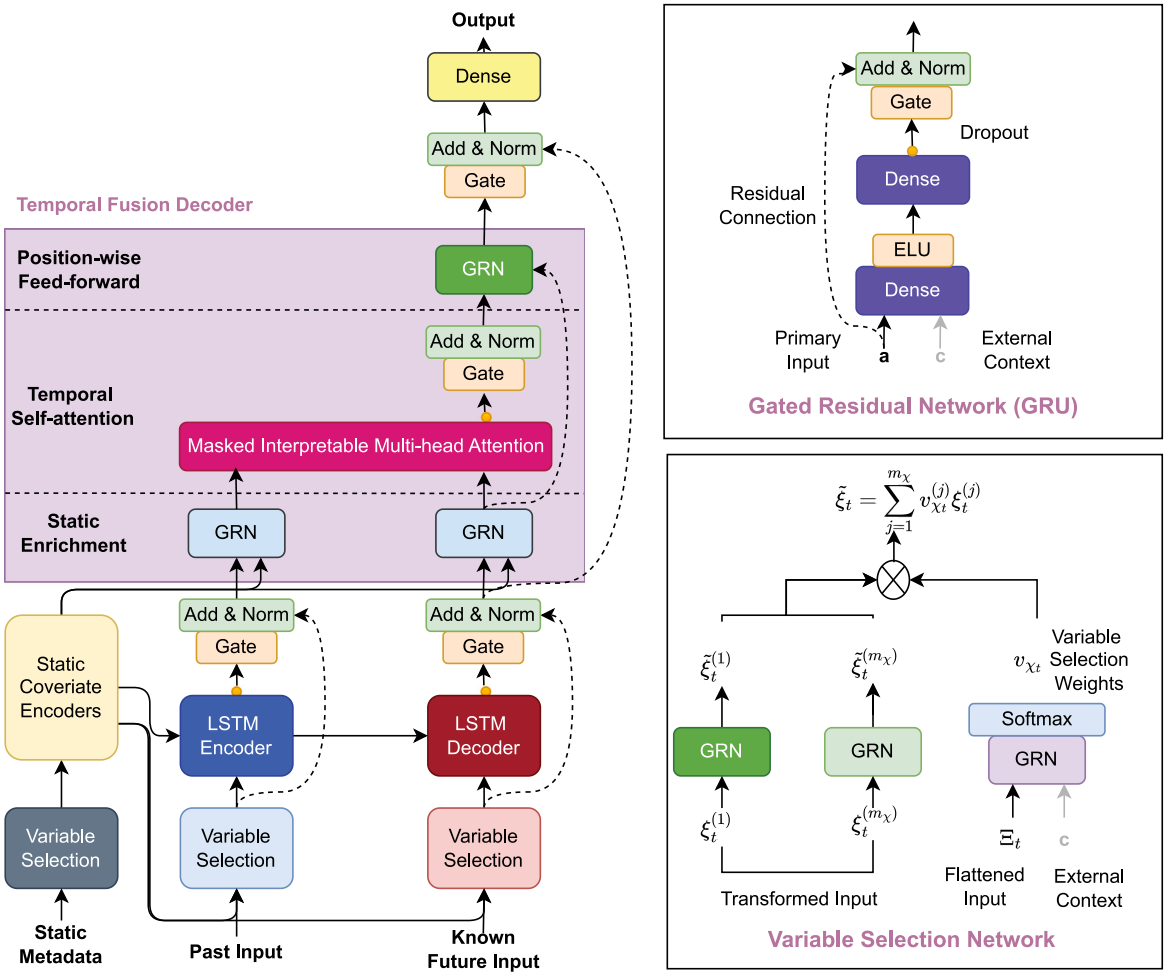


Fig. 2. The architecture of TFT.

4. **Temporal processing:** TFT learns both short- and long-term temporal relationships from time-varying inputs. To capture long-term dependencies, an interpretable multi-head attention block as shown in Eq. (6) is proposed.

$$InterpretableMultiHead(Q,K,V) = \tilde{H}W_H, \tag{6}$$

$$\begin{aligned}\tilde{H} &= \tilde{A}(Q,K)VW_V, \\ &= 1/H\sum_{h=1}^{m_H} Attention(QW_Q^{(h)}, KW_K^{(h)}, VW_V),\end{aligned} \tag{7}$$

where $W_Q^{(h)}, W_K^{(h)}$ are head-specific weights for keys K and queries Q, W_V is the weight for value V that is shared across all heads and W_H is the weight for final linear mapping.
A sequence-to-sequence layer is applied for locality enhancement. It takes inputs $\tilde{\xi}_{t-k:t}$ for the encoder and $\tilde{\xi}_{t+1:t+\tau_{max}}$ for the decoder to generate a set of uniform temporal features that are then used as inputs for the temporal fusion decoder.
5. **Quantile forecasts:** TFT uses quantile forecasts to estimate the spectrum of target values at each prediction horizon. In terms of loss function for training, it minimises the sum of quantile losses across all quantile output.

3.4.2. Leakage detection with TFT

To forecast fuel variances over multiple future steps, for each tank entity i, TFT takes inputs that include: a set of static features s_i (tank id, tank maximum height, tank maximum volume), sequential exogenous inputs $x_{i,t-k:t+\tau}$ (time of day, inventory height, inventory volume, tank temperature) and the past fuel variance $y_{i,t-k:t}$. τ is the prediction length and k is the size of the lookback window. Based on these inputs, TFT forecasts the future variances:

$$\hat{y}_{i,t:t+\tau} = f(y_{i,t-k:t}, x_{i,t-k:t+\tau}, s_i), \tag{8}$$

The dissimilarity score $f_s(\hat{y}_{i,t:t+\tau}, y_{i,t:t+\tau})$ at time t is computed using the dissimilarity estimator f_s defined below:

$$f_s(\hat{y}_{i,t:t+\tau}, y_{i,t:t+\tau}) = \overline{\hat{y}_{i,t:t+\tau}} - \overline{y_{i,t:t+\tau}}, \tag{9}$$

which computes the difference between the mean of the predicted fuel variances and the mean of the actual fuel variances. It detects a leakage at time t if the dissimilarity scores continuously exceed the thresholds ψ for a step of ς after t, to identify those persistent changes. The threshold ψ is adaptively updated based on the scores observed so far. Specifically, it is estimated as:

$$\psi = \alpha F(S,p), \tag{10}$$

where S is the set of scores observed so far, F represents the quantile function with p denoting the probability value and α being the scalar.

3.5. Textual explanation with ANFIS

In this section, we first explain the architecture of ANFIS and then discuss how to create fuzzified temporal features to ensure the meaningfulness of the IF-THEN rules. Finally, we introduce the semantic constraints used during training to ensure good interpretability of ANFIS.

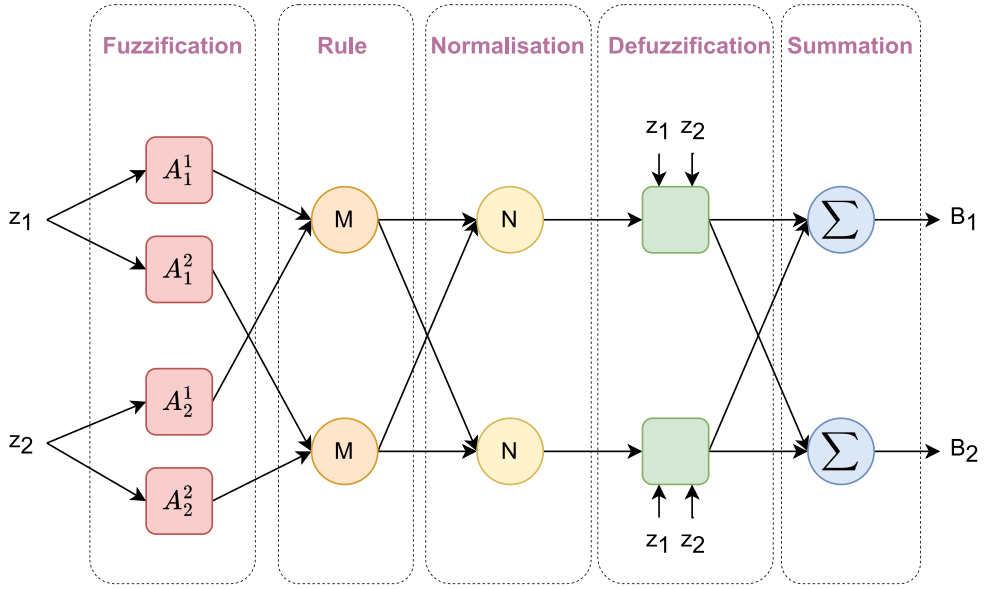


Fig. 3. The architecture of ANFIS.

3.5.1. Adaptive neuro-fuzzy inference system

ANFIS is a neuro-fuzzy inference system whose architecture consists of five layers. Fig. 3 illustrates the structure of ANFIS. Training ANFIS involves optimising the premise and consequence parameters. The five layers include:

1. **Fuzzification layer:** It produces the membership degree for the linguistic label associated with each node based on input values using the chosen MF. For example, when using the Gaussian MF:

$$Gaussian(z, \mu_j, \sigma_j) = e^{-\frac{1}{2}(\frac{z-\mu_j}{\sigma_j})^2}, \tag{11}$$

   where z is the input to node j, μ_j and σ_j are the centre and the width of the Gaussian function, which are also the premise parameters for the node.
2. **Rule layer:** It computes the product of incoming membership values. The result, which is also the output, represents the firing strength of a rule.
3. **Normalisation layer:** It computes normalised firing strengths associated with each rule r, which is the ratio of the rule r's firing strength to the sum of all rules' firing strengths.
4. **Defuzzification layer:** It computes the output for each rule based on its normalised firing strength. We use zero-order Takagi–Sugeno fuzzy models, meaning that a singleton value is used here to represent the certainty degree for each output class.
5. **Summation layer:** It aggregates the outputs from all rules to produce the final classification.

ANFIS establishes a series of fuzzy IF-THEN rules using suitable MFs in the antecedents to produce values in the consequent parts. An example of a Takagi–Sugeno fuzzy rule for prediction is:

$$\text{IF } (z_1 \text{ is } A_1^r) \text{ AND } (z_2 \text{ is } A_2^r) \text{ THEN } (f_1 \text{ is } B_1) \text{ AND } (f_2 \text{ is } B_2) \tag{12}$$

where A_1^r, A_2^r are the fuzzy sets associated with input features z_1, z_2 for rth rule, B_1, B_2 are the degree values expressing the likelihood of outcome class f_1, f_2. The final classification in the consequent can be ultimately determined based on which class has the highest likelihood.

3.5.2. ANFIS for textual explanation

ANFIS is implemented to perform classification using tailored features extracted from 30 days of data to determine whether a leakage has occurred. The learnt IF-THEN rules can then be used as explanations for fuel leakage analysis. The inputs z to ANFIS need to be carefully set up to ensure the generated rules are genuinely beneficial to practitioners.

When a user needs to confirm if a leakage has occurred at day l, data from the past 30 days $[l-30, l]$ is retrieved. We select 30 days because SIR methods generally analyse this amount of data. A time-based fuzzy set {Recent, Medium Term, Long Term} is defined following the approach of Bhatia and Hagras (2022), representing look-back periods. Recent covers the most recent 7 days, Medium Term covers days 5 to 19, and Long Term covers days 15 to 30, which is set up based on experience. Next, time-based fuzzification is performed on the input time series to create features representing aggregated time series values over a specified period. For example, y_{recent} represents the fuel variance value of the recent period, computed using the most recent 7 days' fuel variance based on the predefined time-based fuzzy set. The time series inputs include fuel variance, temperature and inventory height. Given the objective of explaining fuel leakage, which involves comparing fuel variance across different periods, we further derive $\Delta\left(y_{recent}, y_{medium}\right), \Delta\left(y_{recent}, y_{long}\right)$. These values denote the difference in fuel variance between the recent and the medium term and between the recent and the long term, respectively. A large negative difference in fuel variance of the recent period and fuel variance of the medium or long term essentially indicates a fuel leakage.

For textual explanations, the input features can be described in linguistic terms (e.g. Low, Medium, High) with ANFIS. This is accomplished through the fuzzification layer, which utilises predefined linguistic labels and learned MFs to translate crisp input values into degrees of match with linguistic terms. Additionally, the top-firing IF-THEN rules can be retrieved to provide a textual representation of the reasoning behind the model's predictions.

3.5.3. Semantic constraint for interpretability improvement

During the experiments, we have observed that the obtained IF-THEN rules might not meet our expectations in terms of interpretability. While the interpretability of fuzzy models is often presumed to be inherent, this is not always the case, especially when employing adaptive learning techniques to optimise the fuzzy inference procedures for complex systems (Zhou & Gan, 2008). This is due to the contradicting goals of accuracy and interpretability and research has been conducted to improve model interpretability in data-driven fuzzy modelling. Factors such as the number, coverage, normality and distinguishability

of MFs should be considered when designing the system. Hence, two semantic constraints suggested by de Oliveira (1999) are added to the objective function as shown in Eq. (13). J_G represents the performance measure function and we use the cross-entropy loss function. λ_1 and λ_2 are the penalty factors.

$$J = J_G + \lambda_1 J_1 + \lambda_2 J_2, \tag{13}$$

J_1 represents the distinguishability constraint to ensure each MF is distinct enough from others:

$$J_1 = \frac{1}{2}\sum_{m=1}^{N}(M_q(z^m) - 1)^2 I(M_q(z^m) - 1), \tag{14}$$

where z^m is the mth sample, $M_q(\cdot)$ is the sigma-count measure of the membership degrees, defined by

$$\begin{aligned} M_q(z^m) &= \sqrt[q]{\sum_{j=1}^{n}(\upsilon_j^q(z^m))}, \\ I(x) &= \begin{cases} 1, x > 0 \\ 0, x \le 0 \end{cases}, \end{aligned} \tag{15}$$

where υ_j is the jth MF, and q controls the strength of the distinguishability. When q = 1, it exhibits a strong constraint and it becomes weaker as q increases.

J_2 represents the coverage constraint to ensure that the generated MFs cover the entire universe of discourse of a variable:

$$J_2 = \frac{1}{2}\sum_{m=1}^{N}(z^m - \hat{z}^m)^2, \tag{16}$$

$$\hat{z}^m = \frac{\sum_{j=1}^{n}\upsilon_j(z^m)d_j}{\sum_{j=1}^{n}\upsilon_j(z^m)}, \tag{17}$$

where d_j is the centre of the MF of node j.

4. Experiments

In this section, we first introduce the real-world fuel data collected from the service stations that we use to evaluate EXFLD and the configurations of the algorithms. Next, we conduct case studies and experiments to evaluate EXFLD by investigating the following questions:

1. What interpretable results does EXFLD provide regarding fuel leakage, and how effectively do they explain the leakage phenomena?
2. How does incorporating semantic constraints during the training of ANFIS enhance the interpretability of the generated textual explanations?
3. How does EXFLD perform in terms of accuracy and detection delay for leak detection compared to other state-of-the-art online CPD baselines? Can high accuracy and short detection delay be achieved simultaneously?

4.1. Dataset

The dataset we used is sourced from service stations across various states in Australia between 2020 and 2023. Overall, we collect 167 tank samples from 67 different sites, most of which have more than one tank. This dataset comprises inventory data recorded at 30-min intervals, transaction data detailing sales to customers, and delivery data recording restocking to tanks. We define the key variable commonly used by SIR methods, fuel variance, in Eq. (18). The variance at interval T, $\mathrm{Var}(T)$, is determined by the difference between the measured closing volume ($V_{\mathrm{close}}(T)$) and theoretical inventory volume, which can be computed using the opening volume $V_{\mathrm{Open}}(T)$, sales volume over the interval $V_{\mathrm{sales}}(T)$ and delivery volume $V_{\mathrm{delivery}}(T)$ as follows:

$$\mathrm{Var}(T) = V_{\mathrm{close}}(T) - (V_{\mathrm{Open}}(T) - V_{\mathrm{sales}}(T) + V_{\mathrm{delivery}}(T)). \tag{18}$$

The real-world dataset contains various types of noise caused by known phenomena (e.g. temperature-related thermal expansion effects) or unexpected events (e.g. theft). To mitigate the impact of temperature-related noise, we apply temperature compensation, which standardises the volume to 15 degrees Celsius. Additionally, the fuel variance data is segmented into idle periods (when neither transactions nor delivery occurred in the 30-min interval), transaction periods (with transactions occurring in the interval) and delivery periods (restock). We retain only the idle period data as it is less affected by disturbances from other sources of fuel errors.

We reserve the first-year data of each tank sequence for training. TFT is trained on preprocessed normal inventory log data from all tanks to learn how to predict future variances based on past variances, exogenous variables and static variables. For ANFIS, the training data are divided into 30-day segments. For each segment, a duplicated copy with inserted leakage is created, where the leak rate is set at 0.2 gallons per hour (gph) and introduced in the last 4–7 days. Input–output pairs are then created, where the input consists of extracted features from the 30-day segment and the output is labelled as either normal or leakage. We assume that tanks within the same site share some similarities, and one ANFIS model is trained for each site. During training, ANFIS learns to differentiate between normal and leakage cases based on the individual site's data characteristics.

The remaining part of the sequence is used for test evaluation. Due to the scarcity of real fuel leakage cases, we use datasets with induced leaks for experiments, which is a common practice for existing SIR-based studies including (Alayón et al., 2020; Gorawski et al., 2017; Sigut et al., 2014; Toledo et al., 2024). A tank leakage is induced at an average of 0.2 gph in each sequence, adhering to the standard test procedure of the United States Environmental Protection Agency (United States Environmental Protection Agency, 2019a, 2019b). Each simulated leak begins no earlier than 6 months into the test sequence and persists for approximately 1 month, after which tank operations are presumed to cease, meaning no data will be available thereafter.

Following the guidelines (United States Environmental Protection Agency, 2019a, 2019b), we conduct leakage simulation with adjustments to match our data sampling rate. For each sequence, its leak rate lr is first established by randomly sampling from a uniform distribution ranging between 0.2*(1%–30%) gph and 0.2*(1 + 30%) gph. It is assumed that the leakage occurs at the bottom of the tank and persists throughout the entire leakage period, which is a common assumption from previous works (Alayón et al., 2020; Gorawski et al., 2017; Sigut et al., 2014; Toledo et al., 2024). Then, for each interval T during the leakage period, the fuel variance value is adjusted by subtracting the leakage volume computed based on the corresponding leak rate lr, as follows:

$$\mathrm{Var}_{\mathrm{adj}}(T) = \mathrm{Var}(T) - 0.5 \times lr\sqrt{\frac{h_{pl}}{h_{max}}},$$

where $\mathrm{Var}_{\mathrm{adj}}(T)$ is the adjusted fuel variance value, h_{pl} is the current product level height and h_{max} denotes the maximum inventory height. Finally, to ensure consistency, other relevant variables, including inventory volume and inventory height, are also recalculated to account for the induced leakage volume.

4.2. Setup

EXFLD is implemented in Python 3.8.10, with PyTorch-forecasting 1.0.0 (Beitner, 2020) used to implement TFT and the implementation of ANFIS is based on FOX (Pasquadibisceglie, Castellano, Appice, & Malerba, 2021). The source code can be found at: https://github.com/ruimin-chu/EXFLD. The experiments are run in an environment of AMD EPYC 7502 CPU processor and NVIDIA A100 GPU. Both models are trained for 50 epochs with early stopping. Some parameters are informed by domain knowledge to align with real-world operational requirements (i.e. lookback window size, step size), while others are

Table 2
Hyperparameter setup.

TFT hyperparameters	ANFIS hyperparameters	Leakage detection setup
Forecast horizon (τ): 72	# of fuzzy sets: 3	Quantile probability (p): 0.985
Lookback window size (k): 240	Membership function: Gaussian	Threshold scalar (α): 1.75
Step size (ς): 16	Feature set: { $\Delta(y_{recent}, y_{medium})$,	
Hidden units: 43	$\Delta(y_{recent}, y_{long})$ }	
# of LSTM layer: 1		
# of attention heads: 3		
Hidden continuous size: 27		
Learning rate: 0.02		
Dropout rate: 0.25		
Clipping gradient: 0.05		

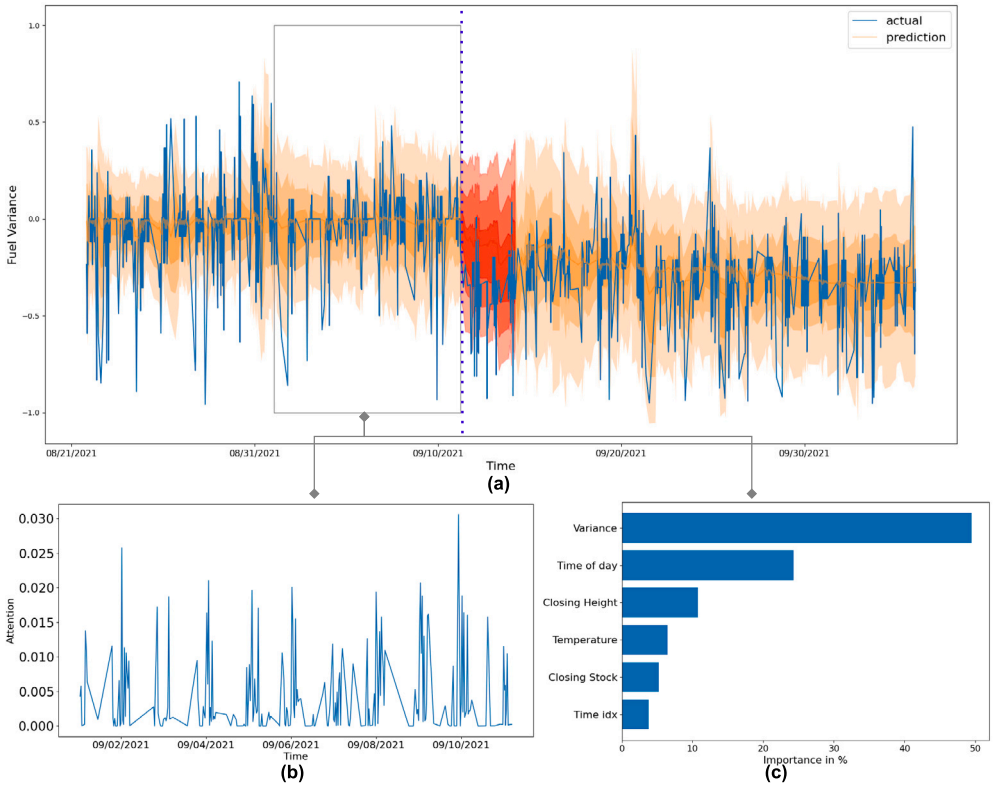


Fig. 4. Case study of visualised explanation.

fine-tuned based on optimisation results. The hyperparameter setup is detailed in Table 2. Specifically, the TFT hyperparameters are tuned using Optuna, while the ANFIS hyperparameters are chosen based on grid search. Further sensitivity analysis will be discussed in Section 4.5.4.

4.3. Case studies on interpretability

In this section, we aim to answer the first question introduced at the start of Section 4. Through case studies, we evaluate the visualised explanations obtained from the detection module and the textual explanations obtained from the explanation module, focusing on their effectiveness in explaining fuel leakage.

Graph-based Explanation:

We present the visualised interpretable results generated with TFT in Fig. 4. This includes the plot of the actual fuel variance trend versus prediction (Fig. 4(a)), instance-wise attention weight pattern (Fig. 4(b)) and instance-wise feature importance ranking (Fig. 4(c)). Fig. 4(a) provides a comparison between the actual fuel variance trend (depicted by the blue line) and the quantile prediction made by TFT (depicted by the orange lines). This visualisation is useful for understanding the detected change points. In the highlighted red region, where the leakage is known to occur, the prediction deviates noticeably from the actual variance, signalling abnormalities. This discrepancy assists practitioners in validating detection results by offering a visualised quantitative comparison between observed and predicted behaviours.

Fig. 4(b) illustrates the attention weight pattern, showing the relative importance of past timesteps for the prediction at the timestep indicated by the purple dotted line in Fig. 4(a). This visualisation reveals which historical data points contribute most significantly to the model's local forecast. Fig. 4(c) displays the feature importance ranking for the timestep where leakage occurs, identifying which features contribute the most to the prediction at that moment. Though they enable interpretability that provides users with insight into how TFT processes data to make forecasts, their utility for validating the detection results is very limited. These interpretations can be used to examine TFT's forecasting decisions but do not explain the detection of a change point, which is the key interest in our problem. Additionally, the information disclosed is limited. For example, the highest spike in attention weight pattern in Fig. 4(b) around 09/10/2021 indicates that the fuel variance at this point contributes the most to the future prediction for the step at the purple dot line. However, this information does not provide practitioners with a direct way to verify whether leakage occurred at

(a) Local explanations for a timepoint with leakage.

(b) Local explanations for a timepoint with no leakage.

Fig. 5. Textual explanations by ANFIS.

that specific time point, as it does not establish a direct causal link to the event.

Based on the above discussion, we can summarise that although TFT offers visualised explanations that aid in validation to some extent, some of which merely present their findings rather than enabling sufficient validation for our study. Relying solely on these visualised explanations is insufficient for our problem.

Textual Explanation: In Fig. 5, we present two examples of textual explanations generated by ANFIS: one for a leakage case (Fig. 5(a), where the red region denotes the leakage period) and the other for a normal case (Fig. 5(b)). Each explanation consists of two main parts. The top firing rule is first displayed along with its firing strength, allowing users to gauge the confidence of the model's prediction. This is followed by the input feature descriptions, where the top two matching linguistic terms and their respective degrees of membership are shown to reflect confidence and also account for uncertainty. These feature descriptions are derived from the trained MFs based on the input variable values as a result of fuzzification. We note that the sum of membership values for an input value x may not equal 1, as fuzzy set theory, unlike traditional probability theory, does not impose the constraint of the summation axiom on membership values for a given element (Han, Kamber, & Pei, 2006).

As shown in the examples, the textual explanations are plausible. For the leakage case in Fig. 5(a), based on the provided top firing rule, a user can learn that the differences in fuel variances between the recent term and the medium or long terms being considered "very negative" cause the model to determine that there is a leakage. This explanation aligns with expert knowledge of leakage cases, thereby enhancing the credibility of the model's decision. Additionally, the feature descriptions in the bottom half of the textual explanation reveal that these differences are primarily considered "very negative" with a membership value of 98%, with a slight tendency towards "moderately negative" with a membership value of 4%. These descriptions allow users to gain further insights into the model's interpretation and confidence levels for the input features. For the normal case, the model suggests that having no significant difference in fuel variance across different periods is normal for the tank. This again aligns with domain knowledge regarding the expected description of differences in fuel variances for a non-leakage case, where the fuel variance is expected to be stable.

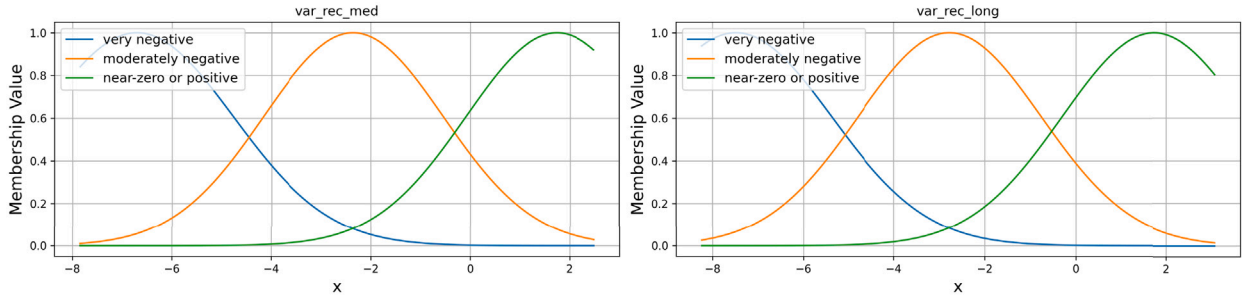
Textual explanations complement visualised explanations by transforming complex data into human-understandable knowledge, which is especially valuable in our context. While visualised explanations, such as comparisons between predicted and actual fuel variances, enable users to spot discrepancies, they often fall short in helping practitioners assess the magnitude or significance of these differences. This limitation is particularly pronounced when dealing with data from multiple sites with varying baseline values, as manual analysis to determine the degree of difference can be time-consuming and impractical. Textual explanations mitigate this problem by describing the degree of difference using linguistic terms, which are intuitive and tailored to each site's unique characteristics. By training an ANFIS on individual site data, the model can learn MFs that adapt to each site's characteristics. Furthermore, during the model design process, we consulted with industry practitioners who expressed preferences for textual explanations as they are easier to interpret. Since these practitioners are not experts in DL algorithms, they find it challenging to extract meaningful insights from graphs or determine severity based on interpretable results from TFT. This feedback further reinforces our decision to incorporate an explanation module to generate textual explanations.

4.4. Semantic constraint on improving interpretability

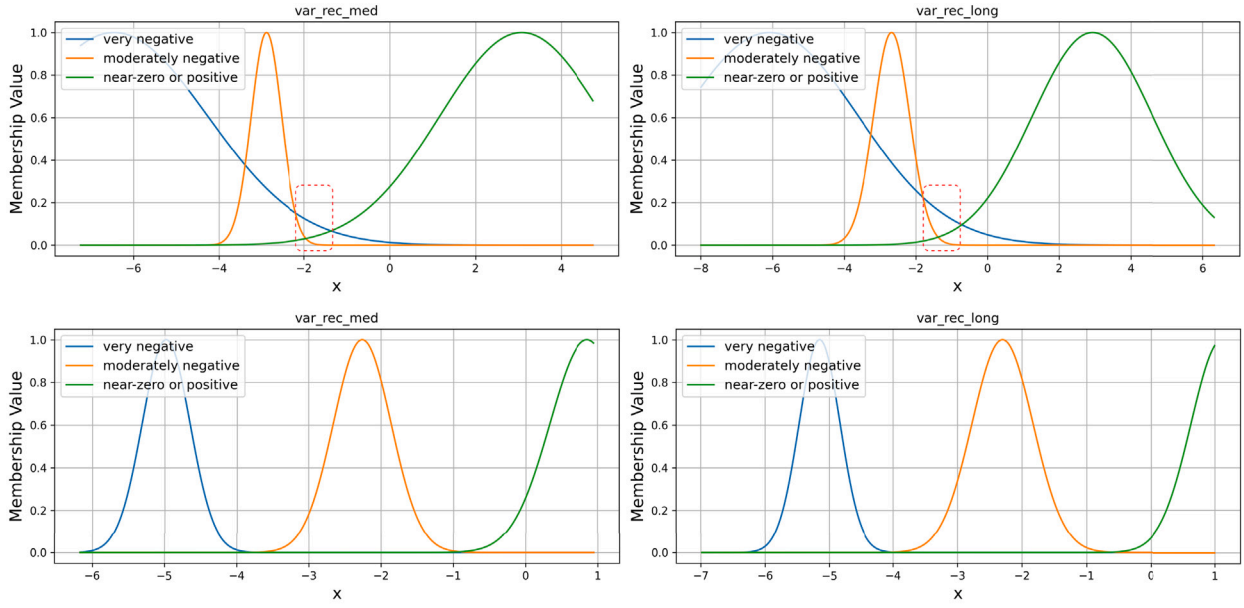
As discussed in Section 3.5.3, interpretability, particularly semantic interpretability, may not always be preserved in an adaptive learning process. In this study, we also focus on enhancing the low-level interpretability of ANFIS by optimising MFs based on semantic criteria to generate more meaningful local explanations. Figs. 6(a) and 6(b) show the MFs of the ANFIS trained with semantic constraints and the unconstrained version, respectively. These figures demonstrate that the ANFIS trained with semantic constraints achieves a more interpretable input space partitioning compared to the conventional ANFIS.

In Fig. 6(a), where ANFIS is trained with semantic constraints, the learnt MFs are well-distributed across the entire universe of discourse, ensuring full coverage. Each MF is sufficiently distinct from the others, ensuring that the corresponding linguistic terms carry clear semantic meanings. Conversely, when trained without constraints, it becomes challenging to assign distinct linguistic labels and meaningful semantics to the resulting fuzzy sets.

In Fig. 6(b), we present cases of unconstrained ANFIS for two different sites. In the top case of Fig. 6(b), for the variable displayed in the



(a) ANFIS trained with semantic constraint.



(b) Unconstrained ANFIS.

Fig. 6. MFs of two inputs for ANFIS.

left subplot, the "very negative" fuzzy set has the highest membership value in the region marked by the red dashed box, whereas the "moderately negative" fuzzy set has the highest value between −3.5 and −2, despite its lower value compared to the red-boxed region. Similar issues can be observed in the variable displayed in the right subplot, particularly within the area highlighted by the red box. This misalignment makes it challenging to assign appropriate linguistic terms that match the actual value ranges, thereby affecting the interpretability of the model. Although post-processing techniques, such as merging overlapping MFs, could be a potential solution, they are impractical in our case. Since one ANFIS is trained for each site to account for varying characteristics while there are many different sites, reassignment of linguistic terms would introduce inconsistency and complexity. In the bottom example of Fig. 6(b), while the MFs are distinguishable, certain input regions are not well-covered. For instance, no fuzzy set has a high membership value for input values between −4 and −3.5 in the left variable. This lack of coverage may hinder the comprehensibility of the system's knowledge (Alonso, Castiello, & Mencar, 2015).

We further compute the average similarity measure (Dubois, 1980; Setnes, Babuska, Kaymak, & van Nauta Lemke, 1998) and the 0.5-completeness score (Mencar & Fanelli, 2008) across all variables and sites to assess the impact of constraints on the semantic interpretability of the system. The similarity measure quantifies the overlap between fuzzy sets, with lower values indicating more distinct fuzzy sets. The 0.5-completeness score is calculated by evaluating the proportion of input values that are adequately covered by fuzzy sets, specifically by counting the proportion of values with their highest membership value exceeding 0.5. A score of 1.0 indicates perfect coverage.

The results show that training with constraints significantly enhances semantic interpretability in terms of coverage, achieving a perfect coverage score of 1.0. In contrast, the unconstrained model obtains a coverage score of 0.7865, suggesting that some input regions are inadequately represented, with maximum membership values below 0.5. The constrained model has a similarity score of 0.1464, which is higher than the score of 0.0404 for the unconstrained model, indicating a greater overlap between the fuzzy sets. Such a trade-off is anticipated. As the two measures are inherently conflicting (Alonso et al., 2015), achieving perfect coverage inevitably increases overlap between fuzzy sets. Nonetheless, this slight reduction in distinguishability is outweighed by the critical improvement in coverage, ensuring that the fuzzy sets effectively cover the entire input range. Supported by the case studies presented in Fig. 6, we can conclude that training with semantic constraints indeed enhances the local interpretability of ANFIS.

4.5. Leak detection performance

In this section, we conduct experiments to demonstrate that EXFLD can maintain good performance in accuracy, by comparing its performance in leak detection to state-of-the-art online CPD baselines, while providing enhanced explanations given in the previous section.

4.5.1. Baseline methods

We compare EXFLD with eight commonly used or state-of-the-art online CPD algorithms. For these baseline models, as they are not designed to handle scenarios with exogenous input series as mentioned in Section 2.2, we use them to analyse the fuel variance sequential data only, which can be considered as CPD on a univariate stream. They are:

- **CUSUM** (Page, 1954): a sequential analysis technique introduced in Page (1954) that can be applied to detect changes in mean.
- **BOCD** (Adams & MacKay, 2007): a model detects changepoints based on the estimated probability distribution of the current run length.
- **M-Statistic** (Li, Xie, Dai, & Song, 2015): an approach that uses computational efficiency kernel M-statistics to measure dissimilarities between blocks of data.
- **ONNC** (Hushchyn, Arzymatov, & Derkach, 2020): an online CPD approach based on neural networks classifier.
- **ONNR** (Hushchyn et al., 2020): an approach based on neural network regressor that follows the idea of RuLSIF (Yamada, Suzuki, Kanamori, Hachiya, & Sugiyama, 2013)
- **LIFEWATCH** (Faber, Corizzo, Sniezynski, Baron, & Japkowicz, 2022): a lifelong learning method that uses Wasserstein distances to compare data distributions and leverages memory to model the distributions.
- **NODE** (Wang, Borsoi, Richard, & Chen, 2023): A strategy based on neural density-ratio estimation, which conducts binary classification across two sliding windows(reference and test), and variational continual learning to facilitating adaptive detection.
- **MOCPD** (Chu et al., 2024): an online CPD framework designed for early fuel leakage detection that stores representative historical data in the memory and adaptively updates the memory and threshold.

The hyperparameters for the baseline models are either set to the default values or optimised through grid search. The window size of baseline methods is set to 72.

4.5.2. Evaluation metrics

We evaluate the performance of EXFLD and the baseline models in detecting fuel leakages using metrics relevant to fuel leakage and online CPD criteria, including recall, precision, F2-score and detection delay. A leak is considered a true positive if detected within 7 days of the ground truth. Recall is measured as the proportion of correctly identified leakages, while precision is calculated by dividing the number of accurately detected CPs by the total number of alarms triggered. F2-score is the weighted harmonic mean of precision and recall, which is calculated as:

$$F_\beta = (1+\beta^2)\frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}, \tag{19}$$

with $\beta = 2$, giving more weight to recall as unrecognised leakages are considered worse than false alarms. Detection delay is computed by taking the average time differences between the detection timestamp and the actual change point timestamp. It is important to note that the F2-score for CPD is not exactly the same as in classification problems. The F2-score in classification evaluates performance on a fixed dataset. Whereas in CPD, detections are continuously performed, and the model must identify change points which are generally rare events, increasing the likelihood and impact of false positives.

4.5.3. Results

In Table 3, we present the experimental results on 0.2 gph fuel leakage data. The results are reported based on the average scores on five runs for those non-deterministic methods. The results show that EXFLD outperforms the baselines in terms of accuracy, achieving the highest F2-score by maintaining both high recall and precision rates compared to other methods. Among the baselines, those incorporating continual learning strategies, which update the models online while retaining past knowledge (e.g. ONNC, ONNR, NODE and MOCPD), show competitive performances. However, some use a fixed threshold, leading to lower precision rates. Given a large number of tank instances with varied dissimilarity score ranges in our dataset, methods employing an adaptive threshold that determines the threshold based on the specific cases show good results. Methods specialised in detecting changes in the mean (e.g. CUSUM) also show good performance due to their alignment with the characteristics of change points in our problem. As the design of EXFLD covers the aforementioned aspects and additionally considers the impact of exogenous variables, it achieves the best accuracy performance. In terms of detection delay, though not having the shortest delay, EXFLD's turnaround time is significantly shorter than the typical time of over 20 days seen with offline SIR-based methods used in the industry.[1]

Table 3
Results on 0.2 gph fuel leakage data.

Method	Recall	Precision	F2	Delay (day)
CUSUM	0.6946	0.2442	0.5074	1.91
BOCD	0.3713	0.2109	0.3222	4.19
M-Statistic	0.4036	0.2767	0.3691	3.56
ONNC	0.6257	0.3050	0.5170	3.83
ONNR	0.7820	0.2703	0.5639	4.25
LIFEWATCH	0.5928	0.3437	0.5177	3.38
NODE	0.4900	0.2727	0.4214	3.27
MOCPD	0.7844	0.4844	0.6979	4.69
EXFLD	0.8204	0.7153	**0.7969**	4.50

We also report the efficiency of EXFLD and baseline models at inference by recording the time taken to make a decision at each step. The average runtimes in milliseconds for each method, listed in ascending order, are as follows: *CUSUM*: 0.0028, *MOCPD*: 1.09, *LIFEWATCH*: 1.19, *BOCD*: 1.67, *EXFLD*: 2.26, *M-Statistic*: 4.07, *ONNC*: 4.28, *ONNR*: 11.25, *NOCD*: 16.88. Although inference time is recorded, it is less critical to our study as these times are significantly shorter compared to the average detection delay.

4.5.4. Sensitivity analysis

In this section, we examine the sensitivity of EXFLD to critical parameters that influence its performance, including the prediction length (τ) and key configurations of the ANFIS model. The analysis helps evaluate how these parameters impact accuracy, detection delay and explainability, providing insights into their trade-offs and guiding the selection of the optimal configuration.

Effect of prediction length τ:

Achieving high accuracy and low detection delay simultaneously in fuel leak detection poses challenges due to the noisiness of real-world data. Single anomalies or short-term trend changes may occur unexpectedly. We conduct experiments to analyse EXFLD's sensitivity to the prediction length τ, which affects detection accuracy and detection delay. τ serves as a decision delay allowance, enabling the method to detect leaks based on a proportion of recent data to alleviate the impact of outliers. Fig. 7 depicts the F2-scores versus detection delays for τ values between $\{24, 48, 60, 72, 96\}$. The graph reveals that as τ increases, the model achieves higher accuracy by leveraging a broader temporal context, enabling it to better differentiate between genuine leaks and local trend changes. Meanwhile, this improvement in accuracy comes at the cost of longer detection delays, as the system requires more data for decision-making. The point closest to the upper-left corner of the graph indicates the optimal prediction length that provides the best trade-off between accuracy and detection delay.

[1] http://www.nwglde.org/methods/sir_quantitative.html.

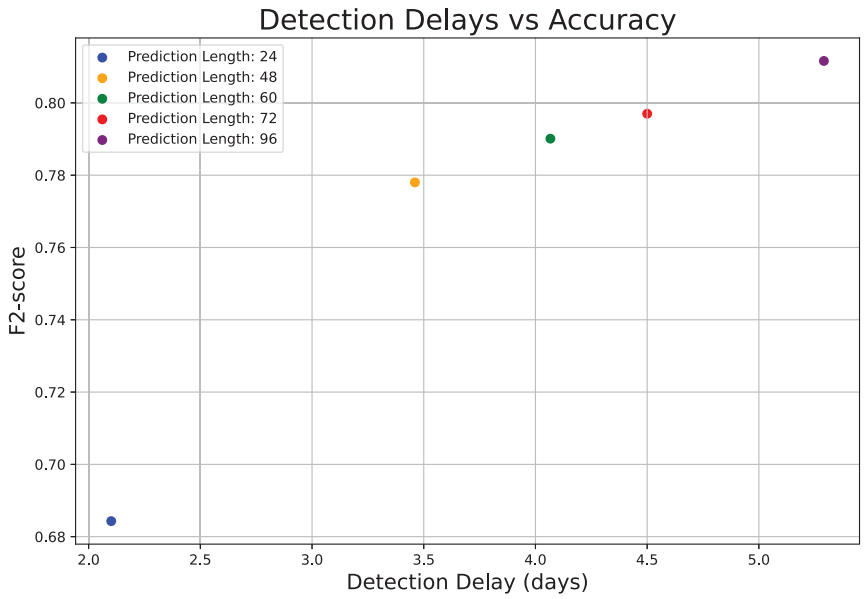


Fig. 7. Trade-off between accuracy and detection delay.

Table 4
Comparative analysis of the seven top-performing ANFIS setups. NFS indicates the number of fuzzy sets, NR denotes the number of rules and NC denotes the number of conditions.

Setups			Accuracy	Interpretability	
Feature set	MF	NFS	F2-score	NR	NC
$\{\Delta(y_{recent}, y_{medium}), \Delta(y_{recent}, y_{long})\}$	Gaussian	5	0.8579	25	2
$\{\Delta(y_{recent}, y_{medium}), \Delta(y_{recent}, y_{long})\}$	Gaussian	3	0.8569	9	2
$\{\Delta(y_{recent}, y_{medium}), \Delta(y_{recent}, y_{long})\}$	Trapezoidal	3	0.8565	9	2
$\{\Delta(y_{recent}, y_{medium}), \Delta(y_{recent}, y_{long})\}$	Gaussian	2	0.8557	4	2
$\{\Delta(y_{recent}, y_{medium}), \Delta(y_{recent}, y_{long}), \text{Height}_{recent}\}$	Gaussian	5	0.8556	125	3
$\{\Delta(y_{recent}, y_{medium}), \Delta(y_{recent}, y_{long})\}$	Trapezoidal	5	0.8536	25	2
$\{\Delta(y_{recent}, y_{medium}), \Delta(y_{recent}, y_{long}), \text{Height}_{recent}\}$	Gaussian	3	0.8526	27	3

Effect of feature set, membership function and number of fuzzy sets:

For ANFIS, we explore various configurations, with λ_1 and λ_2 (introduced in Section 3.5.3, which controls the contribution of semantic constraints) set to 0.1. The configurations include:

- **Membership functions**: Gaussian, Triangular and Trapezoidal. These are all the most common types of MFs (Ali, Ali, & Sumait, 2015).
- **Feature set**: In addition to the two key features $\{\Delta(y_{recent}, y_{medium}), \Delta(y_{recent}, y_{long})\}$ introduced in Section 3.5, we also consider other potentially related features, including $\text{Temperature}_{recent}$ and Height_{recent}, based on the expert knowledge. Temperature is associated with evaporation losses which contributes to fuel variances, while inventory height may reflect tank calibration errors which also affect fuel variance.
- **Number of Fuzzy Sets (NFS)**: 2, 3 and 5. NFS determines the granularity of the fuzzy partition. These values are chosen to maintain a manageable number of categories based on the 7 ± 2 theory (Miller, 1956), ensuring the system remains interpretable while accurate (Alonso, Magdalena, & Guillaume, 2008). Two fuzzy sets provide binary decisions, leading to the simplest structure. Three or five fuzzy sets allow for more granularity without introducing excessive complexity. As odd numbers, they enable a middle term, which was also explored in Alonso, Ducange, Pecori, and Vilas (2020).

To study and determine the optimal ANFIS setup, we consider both accuracy and interpretability. While ANFIS is primarily used to generate textual explanations, we check its accuracy to gauge its performance and determine the best setup. For accuracy, we report the F2-score of ANFIS on the validation set (i.e. 25% of the train set for ANFIS) in performing binary classification. For interpretability, we report the Number of Rules (NR) and the Number of Conditions (NC), where NC corresponds to the number of features. As highlighted in reviews of interpretability measures for fuzzy rule-based systems (Alonso et al., 2015; Gacto, Alcalá, & Herrera, 2011), these metrics are widely recognised in the field as key indicators of model interpretability. NR reflects the compactness of the rule system, while NC indicates the total rule length (Alonso et al., 2015). Lower values suggest simpler structures and thus lead to a better understanding of the model.

Table 4 presents the top seven ANFIS setups that achieve the highest accuracy scores alongside their respective interpretability results. The accuracy scores across setups are relatively close, with Gaussian and Trapezoidal MFs achieving higher results. Another critical observation is that adding more features or increasing the NFS does not always improve accuracy. In fact, excessive complexity can lead to overfitting, which limits the model's ability to generalise on unseen data. When accuracy scores are close, lower complexity is preferred as it leads to a simpler and more intuitive system, which is critical for end-users who need to interpret the model's decision-making process. While the top-performing setup with five fuzzy sets slightly improves accuracy, it generates a much higher NR, which can be overwhelming for users

to manage. Based on the results in Table 4, we select the second top-performing setup with Gaussian MFs and three fuzzy sets as it achieves a great balance between performance and interpretability.

4.6. Discussion

In the previous sections, we have evaluated the explainability and detection performance of EXFLD. For explainability, we demonstrate that the graph of prediction versus actual fuel variances and textual explanations are more understandable and useful in scenarios where a change point needs to be explained. These explanations highlight the severity and persistence of the abnormal trend that relates to the potential of leakage, assisting practitioners in understanding the current situation. Textual explanations complement visualised outputs by describing differences in human-understandable linguistic terms, which is suitable for scenarios involving instances with various characteristics. By leveraging semantic constraints, EXFLD achieves robust semantic-based interpretability, effectively partitioning the input space with the MFs and ensuring the meaningfulness of the generated local explanations.

In terms of detection performance, EXFLD outperforms baseline online CPD methods by integrating an adaptive threshold system and accommodating exogenous variables. In operational settings, the consequences of false negatives where leakage goes undetected are significant, including financial losses and environmental harm. Meanwhile, false positives may lead to unnecessary interventions, such as operational halts. In the context of fuel leakage, the impact of a false negative is more severe than that of a false positive. This priority informed our decision to use the F2-score for performance evaluation, emphasising more on recall rate. As reflected in Table 3, EXFLD strikes a strong balance between recall and precision, offering improved reliability over baseline methods. EXFLD's comprehensive design, especially with the explainability module, helps mitigate the risks of false positives and false negatives. Specifically, practitioners can adjust the threshold setup to prioritise recall, reducing the likelihood of false negatives. Although this may inevitably lead to increased false positives, the explanation module provides tools such as prediction-versus-actual variance graphs, confidence scores, and degrees of difference to aid practitioners in validating false alarms.

It is important to note that the noisiness of real-world data remains an issue, potentially leading to incorrect judgements by the system. The explanation module may provide explanations that are consistent with the decision of a false alarm. Nevertheless, practitioners can utilise the prediction versus actual fuel variance graph, the confidence score and the provided degree of difference to make a judgement. Ultimately, the system's role is to assist practitioners, who retain the responsibility for making final judgements on whether to act upon the model's recommendations. This collaborative approach ensures EXFLD remains a practical and trustworthy tool for fuel leakage detection.

5. Conclusion and future work

In this paper, we present EXFLD, a novel explainable fuel leakage detection method that provides intuitive explanations for detection validation while ensuring accurate early detection of fuel leakage. EXFLD integrates the high-performance TFT model for online leak detection with the inherently interpretable model, ANFIS, to generate textual explanations. This combination enables EXFLD to address the critical challenge of explainability in fuel leakage detection, making it the first method in this domain to prioritise both interpretability and early detection.

Through several case studies, we demonstrate that EXFLD can provide explanations that sufficiently elucidate its decision, especially by using linguistic terms to describe the degree of deviation of fuel variances between different periods, which is intuitive for humans. We also show that incorporating semantic constraints during the training of ANFIS improves the distribution of MFs, thereby enhancing semantic interpretability and ensuring the meaningfulness of the textual explanations. Finally, the experimental evaluation underscores the effectiveness of EXFLD, achieving an F2-score of 0.7969, which outperforms other online CPD baselines in terms of accuracy.

We demonstrate EXFLD's superior accuracy through experiments using real-world data with induced leakages, where the simulation design adheres closely to industry-standard test procedures to ensure a reasonable approximation of real-world leakage behaviours. However, it is important to acknowledge that these simulations may not fully capture the complexities of actual leaks, such as irregularities in duration and leak rate. Additionally, real leaks occur with much lower frequency than the simulated ones in our experiment. While estimating the relative frequency of fuel leakage, i.e. the proportion of true fuel leakage, remains a challenge. This disparity could make real leaks more difficult to distinguish in operational settings, potentially leading to decreased precision due to a higher rate of false positives. Given the rarity of real-world leakages, future work could focus on closer collaboration with industry stakeholders to conduct controlled physical experiments that simulate actual leak conditions. These experiments would provide valuable data sources to evaluate EXFLD's performance more accurately under true operational settings.

As mentioned in the Dataset Section, an ANFIS model is trained separately for each site, given the variation in individual site data characteristics. Due to its adaptability, ANFIS can adjust the MFs' parameters tailored to site-specific conditions. This ensures that the learned fuzzy sets and generated textual explanations align with the unique characteristics of each environment. On the contrary, the reliance on site-specific training data and configurations indicates EXPLS's limited scalability to diverse geographies with significantly different conditions. Extending EXFLD's application to new regions requires access to data that reflects these varying conditions. Collaborative efforts to share and gather data from different regions could facilitate the method's adaptation to broader contexts.

Finally, the presented EXFLD is an innovative solution to address the challenge of achieving explainability and performance simultaneously, which are desirable properties for many real-world problems. Thus, for future work, we would like to explore the application of EXFLD to other real-world problems with similar settings, e.g. anomaly detection in manufacturing processes and environmental monitoring application tasks where problems can be formulated as CPD or anomaly detection. However, applying EXFLD to these new domains presents certain challenges. It requires a thorough understanding of the specific problem context to ensure that the model's structure is aligned with the unique characteristics of the problem. This adaptation includes determining an optimal number of MFs in the ANFIS and designing meaningful features that effectively clarify reasons for detected change points or anomalies, as well as the degree of the differences involved. Incorporating expert knowledge specific to each domain will be essential to tune EXFLD's setup and eventually help produce meaningful explanations to human operators. Although this customisation process requires some expert input, it is limited to essential structural parameters and, therefore, remains manageable.

CRediT authorship contribution statement

Ruimin Chu: Writing – original draft, Data curation, Methodology, Software, Investigation. **Li Chik:** Data curation, Writing – review & editing. **Yiliao Song:** Writing – review & editing. **Jeffrey Chan:** Supervision, Writing – review & editing. **Xiaodong Li:** Supervision, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Li Chik reports a relationship with Titan Cloud Software that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is funded by the Australian Research Council (ARC) Linkage Grant (LP190100991).

Data availability

Data will be made available on request.

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access*, *6*, 52138–52160. http://dx.doi.org/10.1109/ACCESS.2018.2870052.

Adams, R. P., & MacKay, D. J. (2007). Bayesian online changepoint detection. http://dx.doi.org/10.48550/arXiv.0710.3742, arXiv preprint arXiv:0710.3742.

Alanqary, A., Alomar, A., & Shah, D. (2021). Change point detection via multivariate singular spectrum analysis. *Advances in Neural Information Processing Systems*, *34*, 23218–23230. http://dx.doi.org/10.5555/3540261.3542039.

Alayón, S., Sigut, M., Arnay, R., & Toledo, P. (2020). Time windows: The key to improving the early detection of fuel leaks in petrol stations. *Safety Science*, *130*, Article 104874. http://dx.doi.org/10.1016/j.ssci.2020.104874.

Ali, O. A. M., Ali, A. Y., & Sumait, B. S. (2015). Comparison between the effects of different types of membership functions on fuzzy logic controller performance. *International Journal*, *76*, 76–83.

Alonso, J. M., Castiello, C., & Mencar, C. (2015). Interpretability of fuzzy systems: Current research trends and prospects. *Springer Handbook of Computational Intelligence*, 219–237. http://dx.doi.org/10.1007/978-3-662-43505-2_14.

Alonso, J. M., Ducange, P., Pecori, R., & Vilas, R. (2020). Building explanations for fuzzy decision trees with the expliclas software. In *2020 IEEE international conference on fuzzy systems (FUZZ-iEEE)* (pp. 1–8). IEEE, http://dx.doi.org/10.1109/FUZZ48607.2020.9177725.

Alonso, J. M., Magdalena, L., & Guillaume, S. (2008). HILK: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism. *International Journal of Intelligent Systems*, *23*(7), 761–794. http://dx.doi.org/10.1002/int.20288.

Aminian, E., Ribeiro, R. P., & Gama, J. (2021). Chebyshev approaches for imbalanced data streams regression models. *Data Mining and Knowledge Discovery*, *35*, 2389–2466. http://dx.doi.org/10.1007/s10618-021-00793-1.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115. http://dx.doi.org/10.1016/j.inffus.2019.12.012.

Beitner, J. (2020). PyTorch Forecasting: Time series forecasting with PyTorch.

Bhatia, A., & Hagras, H. (2022). A time series based explainable interval type-2 fuzzy logic system. In *2022 IEEE international conference on fuzzy systems (FUZZ-iEEE)* (pp. 1–7). IEEE, http://dx.doi.org/10.1109/FUZZ-IEEE55066.2022.9882556.

van den Burg, G. J. J., & Williams, C. K. I. (2022). An evaluation of change point detection algorithms. http://dx.doi.org/10.48550/arXiv.2003.06222, arXiv:2003.06222.

Cho, H. (2016). Change-point detection in panel data via double CUSUM statistic. *Electronic Journal of Statistics*, *10*(2), 2000–2038. http://dx.doi.org/10.1214/16-EJS1155.

Cho, H., & Fryzlewicz, P. (2015). Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *77*(2), 475–507. http://dx.doi.org/10.1111/rssb.12079.

Chu, R., Chik, L., Song, Y., Chan, J., & Li, X. (2024). Real-time fuel leakage detection via online change point detection. http://dx.doi.org/10.48550/arXiv.2410.09741, arXiv preprint arXiv:2410.09741.

Cristello, J., Dang, Z., Hugo, R., & Park, S. S. (2024). Artificial intelligence based leak detection in blended hydrogen and natural gas pipelines. *International Journal of Hydrogen Energy*, *91*, 744–764. http://dx.doi.org/10.1016/j.ijhydene.2024.10.146.

de Oliveira, J. V. (1999). Semantic constraints for membership function optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *29*(1), 128–138. http://dx.doi.org/10.1109/3468.736369.

Deng, A., & Hooi, B. (2021). Graph neural network-based anomaly detection in multivariate time series. *Vol. 35*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 4027–4035). http://dx.doi.org/10.1609/aaai.v35i5.16523.

Duarte, J., Gama, J., & Bifet, A. (2016). Adaptive model rules from high-speed data streams. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *10*(3), 1–22. http://dx.doi.org/10.1145/2829955.

Dubois, D. (1980). *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, http://dx.doi.org/10.2307/2273604.

Environmental Protection Agency Victoria (2018). Preventing liquid leaks and spills from entering the environment (fact sheet). URL https://www.epa.vic.gov.au/-/media/epa/files/publications/1700.pdf.

Faber, K., Corizzo, R., Sniezynski, B., Baron, M., & Japkowicz, N. (2022). LIFEWATCH: Lifelong wasserstein change point detection. In *2022 international joint conference on neural networks* IJCNN, (pp. 1–8). IEEE, http://dx.doi.org/10.1109/IJCNN55064.2022.9892891.

Fernandez, A., Herrera, F., Cordon, O., del Jesus, M. J., & Marcelloni, F. (2019). Evolutionary fuzzy systems for explainable artificial intelligence: Why, when, what for, and where to? *IEEE Computational Intelligence Magazine*, *14*(1), 69–81. http://dx.doi.org/10.1109/MCI.2018.2881645.

Gacto, M. J., Alcalá, R., & Herrera, F. (2011). Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, *181*(20), 4340–4360. http://dx.doi.org/10.1016/j.ins.2011.02.021.

Gao, Y., & Er, M. J. (2005). NARMAX time series model prediction: feedforward and recurrent fuzzy neural network approaches. *Fuzzy Sets and Systems*, *150*(2), 331–350. http://dx.doi.org/10.1016/j.fss.2004.09.015.

Gemeinhardt, H., & Sharma, J. (2023). Machine learning-assisted leak detection using distributed temperature and acoustic sensors. *IEEE Sensors Journal*, http://dx.doi.org/10.1109/JSEN.2023.3337284.

Gill, R. S., Keating, J. P., & Baron, M. I. (2006). Detecting abrupt leaks in blended underground storage tanks. *Communications in Statistics. Theory and Methods*, *35*(4), 727–742. http://dx.doi.org/10.1080/03610920500498865.

Gorawski, M., Gorawska, A., & Pasterak, K. (2017). The TUBE algorithm: Discovering trends in time series for the early detection of fuel leaks from underground storage tanks. *Expert Systems with Applications*, *90*, 356–373. http://dx.doi.org/10.1016/j.eswa.2017.08.016.

Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., & Yang, G.-Z. (2019). XAI—Explainable artificial intelligence. *Science Robotics*, *4*(37), eaay7120. http://dx.doi.org/10.1126/scirobotics.aay7120.

Guo, Y., Gao, M., & Lu, X. (2022). Multivariate change point detection for heterogeneous series. *Neurocomputing*, *510*, 122–134. http://dx.doi.org/10.1016/j.neucom.2022.09.021.

Gupta, M., Wadhvani, R., & Rasool, A. (2022). Real-time Change-Point Detection: A deep neural network-based adaptive approach for detecting changes in multivariate time series data. *Expert Systems with Applications*, *209*, Article 118260. http://dx.doi.org/10.1016/j.eswa.2022.118260.

Han, J., Kamber, M., & Pei, J. (2006). Data mining: Concepts and techniques. *Morgan Kaufinann*, *10*(559–569), 4.

Hikmet Esen, M. E., & Ozsolak, O. (2017). Modelling and experimental performance analysis of solar-assisted ground source heat pump system. *Journal of Experimental & Theoretical Artificial Intelligence*, *29*(1), 1–17. http://dx.doi.org/10.1080/0952813X.2015.1056242.

Hushchyn, M., Arzymatov, K., & Derkach, D. (2020). Online neural networks for change-point detection. http://dx.doi.org/10.48550/arXiv.2010.01388, arXiv preprint arXiv:2010.01388.

Jagirdar, H., Talwadker, R., Pareek, A., Agrawal, P., & Mukherjee, T. (2024). Explainable and interpretable forecasts on non-smooth multivariate time series for responsible gameplay. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 5126–5137). http://dx.doi.org/10.1145/3637528.3671657.

Jang, J.-S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, *23*(3), 665–685. http://dx.doi.org/10.1109/21.256541.

Kacprzyk, K., Liu, T., & van der Schaar, M. (2024). Towards transparent time series forecasting. In *The twelfth international conference on learning representations*.

Kawahara, Y., & Sugiyama, M. (2012). Sequential change-point detection based on direct density-ratio estimation. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *5*(2), 114–127. http://dx.doi.org/10.1002/sam.10124.

Keating, J. P., & Mason, R. (2000). Using statistical models to detect leaks in underground storage tanks. *Environmetrics: The Official Journal of the International Environmetrics Society*, *11*(4), 395–412. http://dx.doi.org/10.1002/1099-095X(200007/08)11:4<395::AID-ENV420>3.0.CO;2-K.

Keneni, B. M., Kaur, D., Al Bataineh, A., Devabhaktuni, V. K., Javaid, A. Y., Zaientz, J. D., et al. (2019). Evolving rule-based explainable artificial intelligence for unmanned aerial vehicles. *IEEE Access*, *7*, 17001–17016. http://dx.doi.org/10.1109/ACCESS.2019.2893141.

Knoblauch, J., & Damoulas, T. (2018). Spatio-temporal Bayesian on-line changepoint detection with model selection. http://dx.doi.org/10.48550/arXiv.1805.05383, arXiv.Org.

Li, Z., Shui, A., Luo, K., Chen, J., & Li, M. (2011). SIR-based oil tanks leak detection method. In *2011 Chinese control and decision conference* CCDC, (pp. 1946–1950). IEEE, http://dx.doi.org/10.1109/CCDC.2011.5968519.

Li, D., Tan, Y., Zhang, Y., Miao, S., & He, S. (2023). Probabilistic forecasting method for mid-term hourly load time series based on an improved temporal fusion transformer model. *International Journal of Electrical Power & Energy Systems*, *146*, Article 108743. http://dx.doi.org/10.1016/j.ijepes.2022.108743.

Li, S., Xie, Y., Dai, H., & Song, L. (2015). M-statistic for kernel change-point detection. *Advances in Neural Information Processing Systems*, *28*, http://dx.doi.org/10.48550/arXiv.1507.01279.

Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*(4), 1748–1764. http://dx.doi.org/10.1016/j.ijforecast.2021.03.012.

Lin, T., Horne, B. G., Tino, P., & Giles, C. L. (1996). Learning long-term dependencies in NARX recurrent neural networks. *IEEE Transactions on Neural Networks*, *7*(6), 1329–1338. http://dx.doi.org/10.1109/72.548162.

Liu, Y., Gong, C., Yang, L., & Chen, Y. (2020). DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. *Expert Systems with Applications*, *143*, Article 113082. http://dx.doi.org/10.1016/j.eswa.2019.113082.

Liu, J., Hou, L., Zhang, R., Sun, X., Yu, Q., Yang, K., et al. (2023). Explainable fault diagnosis of oil-gas treatment station based on transfer learning. *Energy*, *262*, Article 125258. http://dx.doi.org/10.1016/j.energy.2022.125258.

Lu, J., Han, X., Sun, Y., & Yang, S. (2024). CATS: Enhancing multivariate time series forecasting by constructing auxiliary time series as exogenous variables. http://dx.doi.org/10.48550/arXiv.2403.01673, arXiv preprint arXiv:2403.01673.

Lu, Y., Kumar, J., Collier, N., Krishna, B., & Langston, M. A. (2018). Detecting outliers in streaming time series data from ARM distributed sensors. In *2018 IEEE international conference on data mining workshops* ICDMW, (pp. 779–786). IEEE, http://dx.doi.org/10.1109/ICDMW.2018.00117.

Matteson, D. S., & James, N. A. (2014). A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, *109*(505), 334–345. http://dx.doi.org/10.1080/01621459.2013.849605.

Mencar, C., & Fanelli, A. M. (2008). Interpretability constraints for fuzzy information granulation. *Information Sciences*, *178*(24), 4585–4618. http://dx.doi.org/10.1016/j.ins.2008.08.015.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, *63*(2), 81. http://dx.doi.org/10.1037/h0043158.

Mounce, S., Boxall, J., & Machell, J. (2010). Development and verification of an online artificial intelligence system for detection of bursts and other abnormal flows. *Journal of Water Resources Planning and Management*, *136*(3), 309–318. http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000030.

Nguyen, T.-L., Kavuri, S., Park, S.-Y., & Lee, M. (2022). Attentive Hierarchical ANFIS with interpretability for cancer diagnostic. *Expert Systems with Applications*, *201*, Article 117099. http://dx.doi.org/10.1016/j.eswa.2022.117099.

Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, *41*(1/2), 100–115. http://dx.doi.org/10.1093/biomet/41.1-2.100.

Pan, Q., Hu, W., & Chen, N. (2021). Two birds with one stone: Series saliency for accurate and interpretable multivariate time series forecasting. In *IJCAI* (pp. 2884–2891). http://dx.doi.org/10.24963/ijcai.2021/397.

Pasquadibisceglie, V., Castellano, G., Appice, A., & Malerba, D. (2021). FOX: a neuro-fuzzy model for process outcome prediction and explanation. In *2021 3rd international conference on process mining* ICPM, (pp. 112–119). http://dx.doi.org/10.1109/ICPM53251.2021.9576678.

Pramod, C., & Pillai, G. N. (2021). K-means clustering based extreme learning ANFIS with improved interpretability for regression problems. *Knowledge-Based Systems*, *215*, Article 106750. http://dx.doi.org/10.1016/j.knosys.2021.106750.

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., & Cottrell, G. (2017). A dual-stage attention-based recurrent neural network for time series prediction. http://dx.doi.org/10.24963/ijcai.2017/366, arXiv preprint arXiv:1704.02971.

Rojat, T., Puget, R., Filliat, D., Del Ser, J., Gelin, R., & Díaz-Rodríguez, N. (2021). Explainable artificial intelligence (xai) on timeseries data: A survey. http://dx.doi.org/10.48550/arXiv.2104.00950, arXiv preprint arXiv:2104.00950.

Setnes, M., Babuska, R., Kaymak, U., & van Nauta Lemke, H. R. (1998). Similarity measures in fuzzy rule base simplification. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *28*(3), 376–386. http://dx.doi.org/10.1109/3477.678632.

Siddiqui, S. A., Mercier, D., Munir, M., Dengel, A., & Ahmed, S. (2019). Tsviz: Demystification of deep learning models for time-series analysis. *IEEE Access*, *7*, 67027–67040. http://dx.doi.org/10.1109/ACCESS.2019.2912823.

Sigut, M., Alayón, S., & Hernández, E. (2014). Applying pattern classification techniques to the early detection of fuel leaks in petrol stations. *Journal of Cleaner Production*, *80*, 262–270. http://dx.doi.org/10.1016/j.jclepro.2014.05.070.

Silva, M. E., Veloso, B., & Gama, J. (2023). Predictive maintenance, adversarial autoencoders and explainability. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 260–275). Springer, http://dx.doi.org/10.1007/978-3-031-43430-3_16.

Theissler, A., Spinnato, F., Schlegel, U., & Guidotti, R. (2022). Explainable AI for time series classification: a review, taxonomy and research directions. *IEEE Access*, *10*, 100700–100724. http://dx.doi.org/10.1109/ACCESS.2022.3207765.

Toledo, P., Arnay, R., Hernández, J., Sigut, M., & Alayón, S. (2024). Towards a more realistic approach to the problem of detecting fuel leaks in filling stations: Mixed time windows. *Journal of Cleaner Production*, *468*, Article 143094. http://dx.doi.org/10.1016/j.jclepro.2024.143094.

Truong, C., Oudre, L., & Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing*, *167*, Article 107299. http://dx.doi.org/10.1016/j.sigpro.2019.107299.

United States Environmental Protection Agency (2019a). General guidance for using EPA's standard test procedures for evaluating release detection methods. URL https://www.epa.gov/sites/default/files/2019-05/documents/ust-stp-generalguidance_4.pdf.

United States Environmental Protection Agency (2019b). Standard test procedures for evaluating release detection methods: Statistical inventory reconciliation. URL https://www.epa.gov/sites/default/files/2019-05/documents/ust-stp-sir.pdf.

United States Environmental Protection Agency (2023). Semiannual report of UST performance measures end of fiscal year 2023 (october 01, 2022 – september 30, 2023). URL https://www.epa.gov/system/files/documents/2023-11/fy-23-eoy-final-report-11-21-2023.pdf.

United States Environmental Protection Agency (2024). Release Detection for Underground Storage Tanks (USTs): An Introduction. URL https://www.epa.gov/ust/release-detection-underground-storage-tanks-usts-introduction#tankmethods. (Accessed 2024).

Wang, X., Borsoi, R. A., Richard, C., & Chen, J. (2023). Change point detection with neural online density-ratio estimator. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing* ICASSP, (pp. 1–5). IEEE, http://dx.doi.org/10.1109/ICASSP49357.2023.10095321.

Wang, T., & Samworth, R. J. (2018). High dimensional change point estimation via sparse projection. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *80*(1), 57–83. http://dx.doi.org/10.1111/rssb.12243.

Wang, D., Yu, Y., & Rinaldo, A. (2020). Univariate mean change point detection: Penalization, cusum and optimality. *Electronic Journal of Statistics*, [ISSN: 1935-7524] *14*(1), http://dx.doi.org/10.1214/20-EJS1710.

Wu, B., Wang, L., & Zeng, Y.-R. (2022). Interpretable wind speed prediction with multivariate time series and temporal fusion transformers. *Energy*, *252*, Article 123990. http://dx.doi.org/10.1016/j.energy.2022.123990.

Xu, W., Fan, S., Wang, C., Wu, J., Yao, Y., & Wu, J. (2022). Leakage identification in water pipes using explainable ensemble tree model of vibration signals. *Measurement*, *194*, Article 110996. http://dx.doi.org/10.1016/j.measurement.2022.110996.

Yamada, M., Suzuki, T., Kanamori, T., Hachiya, H., & Sugiyama, M. (2013). Relative density-ratio estimation for robust distribution comparison. *Neural Computation*, *25*(5), 1324–1370. http://dx.doi.org/10.1162/NECO_a_00442.

Zhou, S.-M., & Gan, J. Q. (2008). Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. *Fuzzy Sets and Systems*, *159*(23), 3091–3131. http://dx.doi.org/10.1016/j.fss.2008.05.016.

Zhou, B., Liu, S., Hooi, B., Cheng, X., & Ye, J. (2019). Beatgan: Anomalous rhythm detection using adversarially generated time series.. *vol. 2019*, In *IJCAI* (pp. 4433–4439). http://dx.doi.org/10.24963/ijcai.2019/616.