Analysis of digital twin and its physical object: Exploring the efficiency and accuracy of datasets for real-world application

Henry Chima Ukwuoma, Gilles Dusserre, Gouenou Coatrieux, Johanne Vincent

Data Science and Management 7 (2024) 361–375

Contents lists available at ScienceDirect

Data Science and Management

journal homepage: www.keaipublishing.com/en/journals/data-science-and-management

Research article

Analysis of digital twin and its physical object: Exploring the efficiency and accuracy of datasets for real-world application

Henry Chima Ukwuoma [a,*], Gilles Dusserre [a], Gouenou Coatrieux [b], Johanne Vincent [b]

[a] Laboratory for the Science of Risks (LSR), IMT Mines Ales, 30100, Alès, France
[b] IMT Atlantique, Bretagne-Pays de la Loire, France

ARTICLE INFO

Keywords:
Cyber physical systems
Digital twin
Cyber security
Water distribution system

ABSTRACT

The concept of digital twin (DT) has recently gained popularity due to its ability to create a virtual representation of systems in order to improve the performance of its cyber-physical counterpart. This study compares and analyses datasets of DTs and their corresponding physical objects to determine the effectiveness and dependability of DT technology for practical applications. The research aims to proffer a framework for ascertaining the level of (dis)similarity between a physical object and its DT equivalent. A study of water distribution (WADI) is put into perspective. Findings revealed that the proffered framework presents a method for ascertaining a (dis)similarity level (considering datasets) of a physical object and its DT using adequate statistical tests.

1. Introduction

The concept of digital twin (DT) has most recently been recognized as a disruptive technology that can completely alter the operations, security, and manufacturing tendencies of a cyber-physical system (CPS) (Attaran and Celik, 2023; Perno et al., 2022). DT provides an innovative way to gather insights, optimize performance, and make data-driven decisions in real-time by developing virtual representations of physical objects, systems, and processes (IBM, 2023). The connection with the world has changed because of the daily improvement of remarkable technical breakthroughs brought about by the rise of the digital age. Digitalization has emerged as a key driver of innovation and development across industries, from smart manufacturing to smart cities. DT is one of the many modern technologies that have garnered considerable interest because of its tendency to narrow the gap in terms of predictive intrusion detection and production capabilities between physical objects and their virtual counterparts (Falah et al., 2020).

A DT is essentially a virtual replica of its real-world counterpart, which is referred to as a physical system (Ercetin, 2023). The developed DT simulates its physical counterpart's behavior and other related characteristics in real-time from data gathered from sensors, actuators, IoT devices, and other sources. These capabilities enable businesses to benefit in terms of predictive modeling, sophisticated analytics, and real-time data-driven decision-making abilities (Attaran and Celik, 2023). A DT is fundamentally dependent on the datasets it generates in order to predict the behavior, enhance intrusion detection, and improve the performance of its physical equivalent (Varghese et al., 2022). These synthetically generated datasets play a crucial role in developing and improving virtual representation, ensuring that the DT holistically captures the behavior, functionality, and response to diverse operations of the real object. Equally, physical objects generate datasets via network devices, sensors, measurements, and observations, which are crucial for overseeing and managing their operations and further assist in predictive intrusion detection (Braunegg et al., 2020). The accuracy and efficiency of physical objects are also based on the datasets they generate. Building a complete model of a physical object from data gathered from sensors, historical records, and outside sources enables real-time simulation and predictive analytics. As a result, DTs have become a potent instrument with enormous potential that support these predictions or improved manufacturing processes in industries, including manufacturing, healthcare, transportation, and urban planning, since they can mimic physical objects. Therefore, improved manufacturing processes or intrusion detection capabilities can be tested on the DT before implementing them on physical objects. This approach saves time, energy, and, most importantly, resources (Moi et al., 2020).

This article aims to delve into datasets that underpin both DTs and physical objects, examining their similarities and differences, and the implications for real-time decision-making. For this study, the datasets considered are those gathered from a CPS system and its digital object equivalent (data collected from the operations of CPS and its digital

Peer review under responsibility of Xi'an Jiaotong University.
* Corresponding author.
E-mail address: henry.ukwuoma@mines-ales.fr (H.C. Ukwuoma).

https://doi.org/10.1016/j.dsm.2024.04.002
Received 9 October 2023; Received in revised form 1 April 2024; Accepted 6 April 2024
Available online 12 April 2024

equivalent). Understanding these differences is crucial for comprehending the potential advantages and limitations of employing DT technology in real-world scenarios. This article further investigates the level of effectiveness and accuracy of a DT and its physical object for use in real-world applications by comparing and exploring the datasets obtained from the two entities. In this context, a study of water distribution (WADI) is undertaken and validated using the C-Town distribution system.

The remainder of the paper is structured as follows: Section 2 presents the literature review and background information needed to comprehend the study. Section 3 explains materials and methods, the study of WADI, and the proposed architecture for conducting a similarity analysis. Section 4 provides results and discussion. Finally, Section 5 concludes the paper and discusses the scope for future studies.

The proposed framework will serve as a guide to industry and researchers on the possible tests that should be conducted on datasets generated from both the physical object and its digital object equivalent. The (dis)similarity between the datasets will aid industry practitioners in ascertaining what level of decision-making to take and how to improve the DT or physical asset, as the case may be.

2. Literature review

To establish a framework for ascertaining the similarity or dissimilarity of datasets generated by the physical object and its DT counterpart, the literature domain is perused. However, no framework specifically tailored for performing this comparison regarding a WADI system is found. Nonetheless, this study considers the application of statistical tests and virtual comparison tools for ascertaining the similarity or dissimilarity of these datasets. The following studies are reviewed to give a better understanding of statistical tools adopted in the development of a test framework.

Mallikharjuna et al. (2023) adopted machine learning for data pre-processing. The authors also adopted and applied a skewness test during the data pre-processing phase. The authors noted that a crucial statistical metric of skewness is utilized to evaluate the asymmetry of data distribution, emphasizing that skewness can negatively impact how well machine learning models perform, particularly those that rely on the assumption that the data is normal or symmetrical. Furthermore, the authors adopted and applied skewness tests to determine the presence and degree of skewness in various aspects (variables) of the dataset used by their study. There are tests to evaluate skewness, such as the Shapiro-Wilk test, the D'Agostino-Pearson skewness test, and the Pearson skewness test. In the process of data pre-processing, the proper data transformation technique is applied after identifying skewed features to lessen skewness. Depending on the type and degree of skewness in the data, common transformations include the log transformation, square root transformation, and box-cox transformation. This application aims to build a more evenly distributed and representative dataset, ensuring that the machine learning models can better recognize relationships and patterns in the data, increasing their generalization and performance.

In the study of comparing multiclass classification methods using the "Dry Bean Dataset", Salauddin et al. (2023) used a systematic approach that integrated numerous techniques and statistical tools. This included the use of box plots and measures of dispersion and central tendency, to successfully prepare the data for analysis. In a bid to perform data pre-processing, the authors meticulously conducted data cleaning to handle missing values, anomalies, or data errors and achieve data integrity and accuracy before applying appropriate classification techniques. Box plots were used to understand the data's distribution and identify outliers. Box plots present a visual depiction of the range of the data by showing the quartiles, median, and any potential extreme values. The authors were able to locate characteristics with significant variability, skewness, or the presence of outliers by creating box plots for each feature in the dataset. The datasets were subjected to measures of dispersion, such as variance and standard deviation. Understanding the dispersion aids in determining the unpredictability of the data and any potential classification difficulties.

Additionally, measures of tendency were applied to the datasets to provide insights into the typical or average value of the data. Understanding the central tendency is essential for imputing missing values and handling class imbalances. Box plots and dispersion measures assisted in identifying potential class imbalances in a multiclass dataset, avoiding a biased model performance. The authors further used data balancing techniques such as oversampling or undersampling to create a more balanced and representative dataset. Subsequently, the authors applied Gradient Boosting, Random Forest, k-nearest neighbors (k-NN), and support vector machines (SVM) on the dataset.

Moreover, K-fold cross-validation was used to train each approach on the pre-processed data to ensure accurate model evaluation. Standard performance metrics like accuracy, precision, recall, and F1-score were adopted when evaluating the models' performance on the classification task. The study used statistical tools like box plots, dispersion measures, and central tendency to prepare the "Dry Bean Dataset" for comparison with multiclass classification techniques. These tools helped identify outliers, manage missing values, and address class imbalances.

Hussain et al. (2023) stated that homogeneity tests are crucial for identifying the change points or breakpoints in a dataset where a distribution undergoes a change. The authors suggested the adoption of the pyHomogeneity package in Python to perform the test for time series or climate series data. The homogeneity test includes packages for the Pettitt test, four variants of Buishand's test, and the SNHT test. The authors applied the approach to a synthetic dataset to identify break points. The homogeneity test assesses the equal variances of two or more groups or datasets. It is crucial in statistical analyses like ANOVA, *t*-tests, and linear regression. It ensures similar data dispersion across groups, preventing biased results and ensuring statistical analysis validity.

Yang et al. (2022) researched the need to monitor systems and investigate complex attacks that have long-range sequences using system auditing. The authors proposed ZEBRA, a system that can harmoniously combine the search for attack patterns and causal dependency tracking—giving security personnel the option to choose between search and tracking. The proposed system was evaluated using a series of attacks that further proved the proposed system as very effective and efficient.

Mihai et al. (2022) opined that DT is a technology that replicates the elements, processes, dynamics, and firmware of a physical system into a digital counterpart, allowing for seamless monitoring, analysis, evaluation, and prediction. They further stated that it goes beyond traditional computer-based simulations and analysis, incorporating technologies like Internet of Things (IoT), artificial intelligence (AI), 3D models, 5G/6G mobile communications, augmented reality (AR), virtual reality, distributed computing, transfer learning, and electronic sensors. DT offers a platform for testing and analyzing complex systems, which would be impossible in traditional simulations and modular evaluations. Their study revealed that the development of DT faces challenges that include complexities in communication and data accumulation, data unavailability for machine learning models, lack of processing power for high-fidelity twins, interdisciplinary collaboration, and lack of standardized development methodologies and validation measures. The authors proffered that as DTs are in the initial stages of development, thus there is a need for sufficient documentation to address these challenges.

Aheleroff et al. (2021) examined the need to identify the best out of DT capabilities regarding industrial transformation. The author's aim was to determine a DT reference architecture model in the era of Industry 4.0. The authors also proposed DT as a service alongside the proposed reference model. They also investigated relevant Industry 4.0 technologies necessary for building DT. The findings of their study revealed that DT improves by employing Industry 4.0 technologies, such as Cloud, IoT, and AR, to promote industrial transformation.

Wang et al. (2020) proposed GuardHealth, a decentralized blockchain system for improved data privacy and sharing that is efficient and secure. The proposed system can manage sensitive information by adequately

managing data sharing, data preserving, authentication, and confidentiality by exploiting consortium Blockchain and smart contracts. Graph neural network provided a trust model while guaranteeing malicious node detection. Validation revealed that the proposed system possesses better efficiency compared to other approaches.

Yao et al. (2020) discussed the widespread adoption of the IoT and its impact on the increasing prevalence of edge computing. The use of edge nodes, deployed in proximity to device users, is highlighted for its potential to address concerns related to task delay, network bandwidth, battery life, and data privacy. Emphasizing the importance of enduring battery life in mission-critical systems, the authors proposed an energy-efficient task offloading problem based on the alternating direction method of multipliers (ADMM) in a three-tier mobile edge computing (MEC) network. Privacy concerns in data transmission among IoT devices are addressed through the application of differential privacy, integrating privacy-preserving methods with task-offloading processes. Simulations and experiments validate the proposed algorithm's performance and convergence.

A study by Farine and Carter (2022) focused on the use of permutation tests in analyzing animal social network data to test null hypotheses. Permutation tests are common but can lead to significant type I and type II errors if they do not accurately simulate the intended null hypothesis. Two main types of permutations exist: pre-network permutations and node permutations. Pre-network permutations account for biases like geographical, temporal, or sampling effects but only suit random social structure hypotheses. Node permutations are ineffective when nuisance effects impact observed networks, but they can handle non-random social structure hypotheses. The authors proposed a solution by adjusting node or edge values before applying node permutation tests through pre-network permutations. They assessed error rates via simulations due to confounding effects in raw data. Their "double permutation" strategy shows lower elevated error rates than using node permutations alone or with simple variables. While all approaches can lead to increased type I errors under specific conditions, the double permutation approach maintains a 5% error rate even when pre-network permutation testing yields over 30% type I errors. Their study explores robust inference methods, including mixed effects models, limited node permutations, testing various null hypotheses, and creating replicated networks from large datasets. The authors emphasize acknowledging and integrating uncertainty in the analysis. In conclusion, they provide a viable approach to tackle high error rates when testing null hypotheses with social network data, along with alternative methods for robust inference and a call to consider uncertainty explicitly.

Boyes and Watson (2022) proposed an analysis framework for all DTs. They subscribed to the definition of DT by Catapult (2021), which states that "a live digital coupling of the state of a physical asset or process to a virtual representation with a functional output" is universal and sector/domain independent. The author also established a clear-cut difference between CPS and DT, stating that the cyber element is a critical and integral component of the CPS while a DT is as described above. Their study further identified 16 functional components and their characteristics, which are included in the proposed framework. The approach suggests that by concentrating only on functionality and not addressing non-functional requirements, the analysis allows the assessment of numerous physical and logical instantiations of DT.

de Gois et al. (2020) examined a 71-year rainfall dataset in Rio de Janeiro, Brazil, adopting normality and homogeneity tests. While the normality test examines whether a specific dataset follows a normal distribution (a bell-shaped curve) or demonstrates significant deviations from it, the homogeneity test assesses whether two or more datasets have similar variances, indicating comparable variability or significant differences in their variability. The authors adopted Shapiro-Wilk and Jarque-Bera tests for the normality test and Bartlett and Fligner-Killeen tests for the variance test. Findings from their study revealed that the 71-year rainfall dataset does not follow a normal distribution, and the Bartlett test outperformed the Fligner-Killeen test for evaluating variance. The authors emphasized that to adequately define historical time series data, it is necessary to perform several statistical tests.

Liu et al. (2019) proffered a novel density-based spatio-temporal clustering approach that arrests the challenges in detecting hidden dynamic patterns in big spatio-temporal data collected from earth observation systems. The sole aim was to unravel clusters of objects with similar attribute values occurring together across both space and time. The approach utilizes density-based clustering, which is known for its effectiveness in finding arbitrarily shaped clusters and requiring less prior knowledge about the cluster number. However, existing density-based methods often suffer from unstable performance due to the need for user-specified parameters. To overcome these limitations, the proposed method incorporates permutation tests into the clustering process. The steps in the proffered approach include calculating the density of each object on the variance and providing a measure of its density. A fast permutation test is also applied to identify high-density objects, i.e., objects with densities significantly higher than expected by chance. A two-stage grouping strategy is employed to group high-density objects and their neighbors. This step helps in forming spatiotemporal clusters by minimizing the increase in homogeneity. Another permutation test is conducted to evaluate the significance of the identified clusters based on the permutation of cluster members. Experiments were conducted on both simulated and meteorological datasets. The results demonstrate that the proposed method outperforms two state-of-the-art spatiotemporal clustering methods, namely ST-DBSCAN and ST-OPTICS. The superiority of the proposed method lies in its ability to identify inherent cluster patterns in spatiotemporal datasets using permutation tests while also alleviating the difficulty of selecting appropriate clustering parameters.

Gal and Rubinfeld (2019) opined that data portability and interoperability are essential for the global economy. The authors further stated that data standardization can improve data use by lowering metadata uncertainties, data transfer obstacles, and missing data. While standards may restrict private economic activity, they may be necessary for optimal data analysis benefits. They also presented the benefits of data standardization, which include interfacing with other datasets. More so, standardization enhances smoother data flows, improves machine learning, and easier policing. Challenges of standardization include the possible creation of negative externalities such as better profiling, privacy risks, and cybersecurity risks.

Lenhard and Lenhard (2017) stated that the statistical significance of a finding reveals whether it is likely to be due to random fluctuations in the data or if it represents a true effect. However, not all statistically significant results suggest a considerable influence, and some may reflect phenomena that are not readily apparent in ordinary life. The level of significance is determined by factors such as sample size, data quality, and statistical power of the investigation. The authors stated that with huge datasets, even minor impacts can acquire statistical significance, which may or may not be practical. Effect sizes are used to quantify the magnitude of an effect. The authors emphasized that Cohen's d is one of the most used effect size measurements, but there are several others, including Glass' Delta, Hedges' g, Odds Ratio, Eta Square, and others. These effect size indicators go beyond statistical significance to represent the strength of a phenomenon. It also includes tools for computing effects from *t*-tests and ANOVAs.

Gabel and Godehardt (2015) created a dataset comprising pairs of data points and their respective similarities to train a similarity measure. Subsequently, the dataset is utilized to train a neural network to represent the similarity metric. The authors suggest that the network must identify the most essential characteristics in terms of similarity for both data points and then combine these features to get a similarity measure for their approach.

Shirkhorshidi et al. (2015) presented a technological framework for analyzing, comparing, and benchmarking the performance of various similarity measures in distance-based clustering algorithms, with a particular emphasis on high-dimensional datasets. The authors averred

that while similarity metrics have been extensively researched in two- and three-dimensional spaces, their behavior in high-dimensional datasets has received less attention. The authors assessed the impact of various similarity measures on the output of clustering algorithms using fifteen publicly accessible datasets classed as low and high-dimensional. Using the datasets, their approach aimed to improve reproducibility and enable future comparisons of new distance metrics to existing ones. Furthermore, the authors proffered the computation of appropriate distance measurements for datasets that allow comparisons between newly developed and classic similarity or distance measurements. Finally, the authors suggested ways to enhance the accuracy and efficacy of distance-based clustering algorithms in real-world applications by offering insights into the behavior of similarity measures in high-dimensional datasets.

Rodríguez del Águila and Benítez-Parejo (2011) adopted the concept of linear regression models to determine relationships between dependent and independent variables (IVs). The authors further discussed Pearson's correlation coefficient (r) and the coefficient of determination (R^2) to measure the linear association and the explained variability between variables. Simple linear regression is explained, with emphasis on linear equations and hypothesis testing. Their study also adopts multiple linear regression with a structure that involves multiple IVs and regression coefficients. The authors emphasized the importance of verifying assumptions, such as linearity, homoscedasticity, normality of errors, independence, and collinearity. They presented methods for model construction and variable selection, including significance levels, Akaike Information Criterion (AIC), and changes in R^2. More so, the process of goodness-of-fit evaluation is highlighted, involving graphical checks for normality, linearity, homoscedasticity, autocorrelation, collinearity, and influential observations. The study provided a comprehensive introduction to linear regression models, a powerful tool in scientific research for establishing relationships between variables and making data-driven predictions.

Furthermore, Dattalo (2013) opined that multivariate multiple regression (MMR) is a technique for modeling the linear relationship between multiple IVs and multiple dependent variables (DVs). MMR is plural since there are many IVs, and MMR is multivariate because there are several DVs.

2.1. Approaches to conducting similarity

A similarity measure is a quantitative assessment that establishes the level to which a physical object and a digital equivalent are similar and is carried out over time and space. The conduct of similarity analysis reveals the quality of the DT and provides common ground for comparison. These similarity measures are applied in areas such as data mining, image clustering, and recognition (Schleich et al., 2017). However, studies have shown that in order to validate the two phenomena, statistical processes should be adopted for easy comparison and to reduce the complexity for the process of similarity computations (Wright and Davidson, 2020). The reliance on these computations beacons on the ability to establish a probability that the inputs, outputs, and validation data all have related uncertainties for both phenomena. According to Wright and Davidson (2020), these uncertainties inform how reliant or dependable these models can be, as revealed in the work of Rasmussen et al. (2015). Yang et al. (2019) deployed a novel dynamic time wrapping (DTW) algorithm for the achievement of a remote motion-abnormality detection (dynamic system). DTW was used as a similarity tool, and comparison confirmed that the novel DTW is excellent in real-world applications for the detection of time-changing or speed-changing abnormality. While the choice of DTW to the case performed very well, it may not be most suitable for high-dimensional data or data with multiple variables.

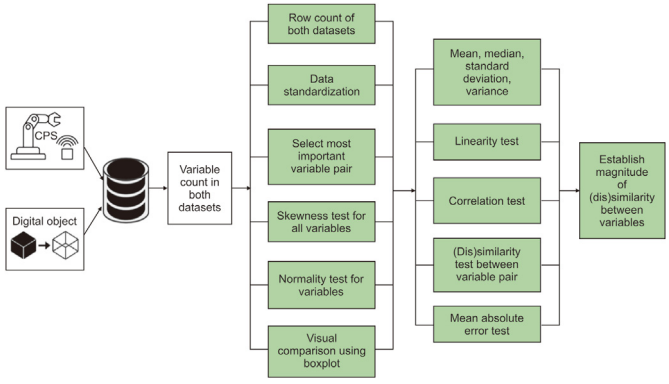

Fig. 1. Proposed framework for testing (dis)similarity between datasets.

Hanoun and Hashim (2019) proffered a novel similarity measure that used the Manhattan distance to quantify the level of similarity of human faces (images). The application of Manhattan distance is limited to grid-like datasets and may not be the most applicable to categorical or continuous data. Huo et al. (2021) established a measure referred to as the quality similarity rate used to ascertain tobacco quality control. The authors applied the similarity regression learning using Mohalanobis distance. The study embarked only on a qualitative approach, which may not be sufficient to scientifically quantify the similarity between a physical and digital object. Khan et al. (2023) explored the standardization of DT technology, introducing a "correspondence measure" approach. The authors explored existing methods, highlighting the importance of interoperability, data privacy, security, and accuracy. The proffered "correspondence measure" can enhance the standardization process by providing a standardized measure of DT's accuracy and reliability. However, there was no practical implementation of the proffered approach.

Thus, this study proposes an approach and demonstrates with WADI systems.

3. Materials and methods

The article proposes a framework for comparison between the physical object and its DT using datasets generated from the two phenomena. More so, a case of WADI for a CPS is adopted for this study. The proposed framework comprises statistical tests that will be executed using Python to ascertain the level of similarity between the CPS WADI dataset and its DT equivalent. The intent is to establish a framework for comparison for other DTs as to how similar the DT is to its physical equivalent. Machine learning is used to apply each statistical test to establish (dis)similarity. The experiment was conducted on a Dell Precision 3571, system with the following specifications: a 12th Gen Intel(R) Core(TM) i7-12800H processor running at 2.4GHz, equipped with 32 GB of RAM, and a 1TB hard disk space. The system operated on the Ubuntu 20.04 LTS operating system. This setup facilitated the installation and execution of the DHALSIM simulator for the experiment.

3.1. The proposed framework

The proposed framework is based on a dataset acquired from the physical testbed (iTrust) such that three major attributes that constitute the major functionality and distribution of water in the distribution network are considered as input variables into the digital hydraulic simulator (DHALSIM). The adopted WADI network depicts the physical connectivity of equipment such as pumps, valves, tanks, pipelines, two raised reservoir tanks, six consumer tanks, two raw water tanks, and a return tank (iTrust, 2018). Chemical dosing systems, pumps, booster pumps, and valves are also included. The WADI network also includes sensors such as flow indicator transmitter (FIT) and level indicator transmitter (LIT) (iTrust, 2018). While sensor readings are continuous, actuator values are discrete.

The application of machine learning methods for a WADI dataset-CPS arises from the desire to enhance predictive capabilities, optimize system

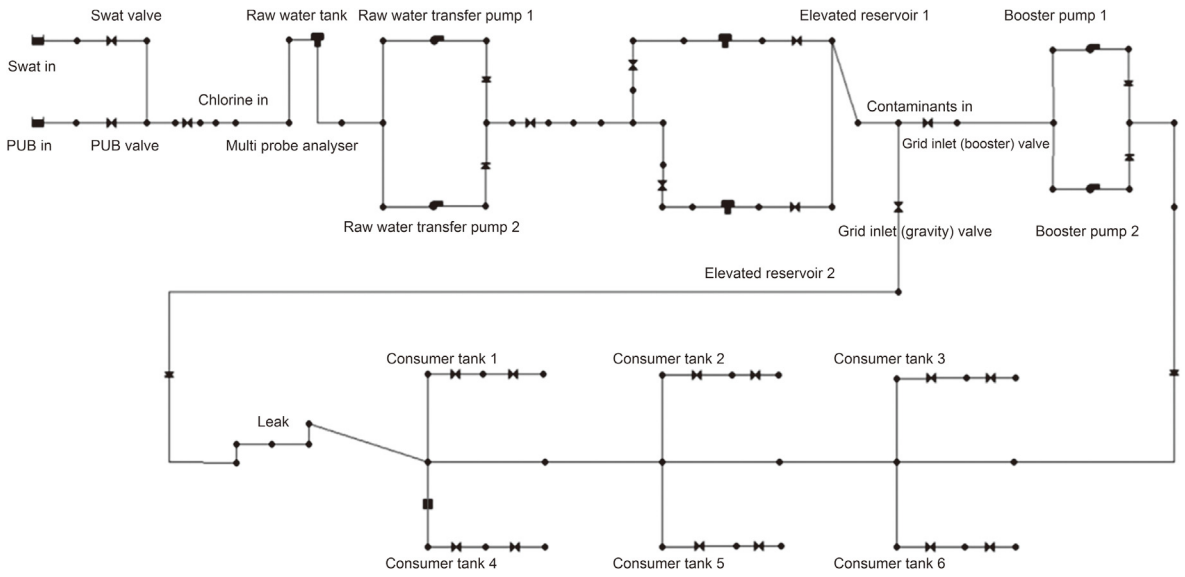

Fig. 2. Digital twin (DT) water distribution (WADI) cyber-physical system (CPS) network topology (Murillo et al., 2021).

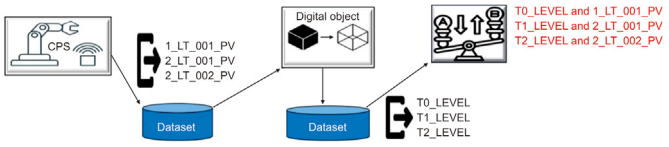

Fig. 3. Features of interest.

operations, and extract meaningful insights from complex and large-scale data. These methods empower water system operators, engineers, and decision-makers to make informed choices for the effective and sustainable management of WADI networks.

The attack dataset created in two days has 172,803 iterations or entries and 131 data columns with float64 (128), int64 (1), and object (2) datatypes with 15 attack scenarios. Furthermore, for normal operations, which were recorded in 14 days, there are 1,048,571 instances or iterations and 130 features or variables with data types of float64 (128) and object (2). For this study, the dataset without attacks is considered, which also served as a baseline dataset for generating a similar dataset from DHALSIM, which is adopted from Murillo et al. (2021), based on the values of the tanks ("1_LT_001_PV", "2_LT_001_PV", "2_LT_002_PV").

In the proposed framework (Fig. 1), a series of tests is conducted to ascertain how similar the two datasets are to improve prediction or production capabilities.

Fig. 1 shows tests that could be carried out in an attempt to establish similarity between the datasets obtained for the physical testbed and the synthetically generated dataset from its equivalent (from the DT). These tests include ascertaining the number of features in both datasets and the row count of each dataset. More so, standardization of both datasets to ensure that both datasets are on the same scale for easy comparison. Based on the nature of the network topology, three variables are identified for this case, and subsequently, skewness and normality tests are carried out on these three selected most valuable features, and the outcome is compared between the feature pairs from both datasets. Visual comparison using a boxplot is used to assess the similarity in shape and values of these selected feature pairs. Additionally, the mean, median, standard deviation, and variance of the features are assessed. Linearity, correlation, and (dis)similarity tests are conducted to establish how the variable pairs behave and are related, and finally, mean absolute error (MAE) is conducted to determine the error margin between the three variables (tank) from the physical testbed dataset and the DT test.

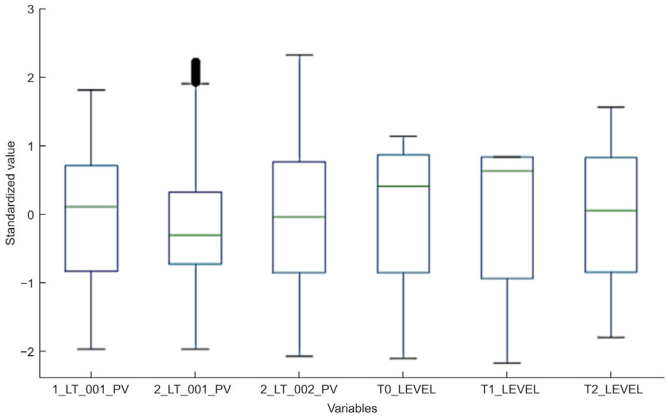

Fig. 4. Boxplot for variables from both datasets (after outlier removal).

The lesser the error, the closer and more similar the variable pairs are, while the larger error proves dissimilarity.

3.2. A study of WADI CPS

To implement the proposed framework, datasets are obtained from iTrust and synthetically generated using DHALSIM (Murillo et al., 2021). Both datasets are expected to behave alike. Moreover, the proposed framework will test both datasets to ascertain their (dis)similarities. Fig. 2 depicts the WADI network topology for the DHALSIM DT.

Fig. 2 shows the flow of water from the source (swat in) to the destination (consumers), showing the arrangement of tanks, pumps, and valves.

Fig. 3 shows the process of how the features of interest are extracted from the two objects for subsequent comparison.

4. Results and discussion

4.1. Implementation of the framework using the datasets

The study applied data standardization to the two datasets. Data standardization is a phase in data analysis and machine learning that is performed before the data is analyzed. It entails changing a dataset's characteristics (features) to have a consistent scale or distribution (Doug,

Table 1
Features of the two datasets.

Count	Physical testbed dataset	Digital twin (DT) dataset
Variable count	130	131
Row count (selected)	30,002	30,002

2023). The goal of data standardization is to bring data to a comparable level, which is especially crucial when dealing with features measured in different units or with varying value ranges (Dickie et al., 2018). Standardization is not always required, although it can provide various advantages such as equalizing variable scales, enhancing model performance, speeding up convergence, and making it easy to interpret for developed models. The concept for applying standardization in the two datasets is simply to ensure the features that are subjected to the test are on the same scale. These standardization procedures create compatibility, measurement, similarity, and symbol standards (Gal and Rubinfeld, 2019).

The three variables from each of the two datasets were extracted to a new dataset, and data pre-processing and tests were carried out thereafter. For the purpose of this study, the first 30,002 iterations were extracted from the physical testbed dataset since the simulator was able to generate 30,002 iterations for the synthetic dataset.

The code below shows how the data was standardized.

```
# Standardize the variables and create a new DataFrame
scaler = StandardScaler()
standardized_df = pd.DataFrame(scaler.fit_transform(df1), columns = df1.columns)
```

The provided code standardizes the variables in the DataFrame df1 using the StandardScaler from scikit-learn and creates a new DataFrame named standardized_df. Standardization (or Z-score normalization) involves transforming the data in such a way that it has a mean of 0 and a standard deviation of 1.

Table 1 displays the characteristics of the two datasets. Kindly note that PT stands for physical testbed, and DT stands for DT. For the purpose of similarity, kindly note that neither dataset contains any form of attack.

Table 1 clearly indicates an extra feature that is contained in the DT dataset. This implies that the two datasets do not contain the same number of variables, which necessarily does not affect their similarity since the DT could capture additional feature(s) to enhance the performance of the physical equivalent.

For this study, the most critical components are selected for comparison, which are the three major tanks that supply water in the WADI network. The primary water supply tank and two elevator tanks supply water to the other segments of the network and to consumers. Based on the knowledge domain of the WADI topology, the study proposes to use only these three features from each of the datasets to demonstrate the (dis)similarity of the datasets as obtained in literature (Bochare et al., 2014; Groves, 2015; Zhao et al., 2019) in the use of selected features adoption and usage for constructive analysis.

4.2. Standardization and visualization of selected variables

For the rest of the study, three variables, i.e., tank levels (primary and 2 elevator tanks), are considered from both datasets. The variable pair includes "1_LT_001_PV" and "T0_LEVEL", "2_LT_001_PV" and "T1_LEVEL", and "2_LT_002_PV" and "T2_LEVEL", where the preceding feature in each pair are from the physical testbed and the latter feature are from the DT synthetically generated dataset.

Both datasets were standardized for ease of comparison. StandardScaler from scikit-learn was adopted and applied. The StandardScaler is a pre-processing tool that performs standardization in the dataset, transforming it to have a mean of 0 and a standard deviation of 1 for each feature. Subsequently, visualization is carried out using a boxplot to explain the possible relationships between the variable pairs.

The code below shows how the process of mean, median, standard deviation, and percentiles were computed.

```
    # Detect outliers using the IQR method
    def remove_outliers_iqr(df, column, lower_bound = 0.25, upper_bound = 0.75):
    Q1 = df[column].quantile(lower_bound)
    Q3 = df[column]. quantile(upper_bound)
    IQR = Q3 - Q1
    lower_bound = Q1-1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] ≥ lower_bound) & (df[column] ≤ upper_bound)]
# Remove outliers from the standardized data
outlier_removed_df = standardized_df.copy()
for column in standardized_df.columns:
    outlier_removed_df = remove_outliers_iqr(outlier_removed_df, column)
# Calculate statistics
statistics = outlier_removed_df.describe().loc[['mean', 'std', '25%', '50%', '75%']]
# Plot the box plot for each variable after outlier removal and standardization
plt.figure(figsize=(10, 6))
outlier_removed_df.boxplot(rot = 45, grid = False)
plt.title('Box Plot of Standardized Variables (After Outlier Removal)')
plt.ylabel('Standardized Value')
plt.xlabel('Variables')
plt.show()
```

The code above defines an IQR-based (interquartile range) outlier removal function, applies it to a standardized DataFrame (outlier_removed_df), calculates statistics (mean, standard deviation, quartiles), and plots box plots for standardized variables after outlier removal. Fig. 4 shows a visual comparison of the feature pairs upon outlier removal.

Comparing the variable pairs in Fig. 4 shows slight pictorial variations in the shapes of the boxplot for the feature pairs. To further analyze, Table 2 shows the mean, median, standard deviation, 25th and 75th percentile of the variable pairs.

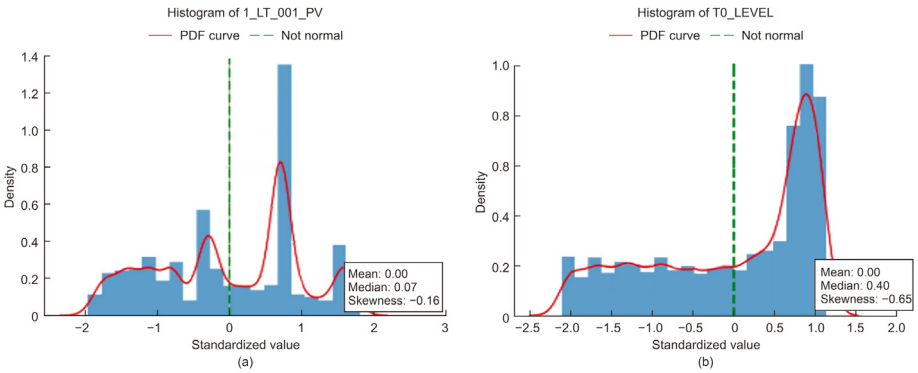

Fig. 5. Normalization and skewness test of variables (a) 1_LT_001_PV and (b) T0_LEVEL.

Table 2
Indices of variable pairs on boxplots.

Variable	Mean	Median	Standard deviation	25th percentile	75th percentile
1_LT_001_PV	0.02	0.11	0.99	−0.82	0.71
T0_LEVEL	0.01	0.41	1.00	−0.85	0.87
2_LT_001_PV	−0.10	−0.30	0.87	−0.73	0.33
T1_LEVEL	0.01	0.64	1.00	−0.93	0.85
2_LT_002_PV	0.00	−0.03	1.00	−0.85	0.76
T2_LEVEL	−0.01	0.06	0.97	−0.84	0.83

Table 2 depicts a high level of similarity in direction (positive or negative) and quantity in values of the mean, median, standard deviation, 25th and 75th percentiles for the 1_LT_001_PV and T0_LEVEL feature pair. This implies a narrow difference in the above-stated indices, which clearly shows a high level of similarity. Similarly, for the 2_LT_001_PV and T1_LEVEL pair, only the standard deviation, 25th, and 75th percentiles represent a considerable similarity in terms of their value. Thus, it can be inferred that the pair has a moderate level of similarity. Table 2 also reveals a high similarity between 2_LT_002_PV and T2_LEVEL variables for the mean, standard deviation, 25th, and 75th percentiles values. It can also be inferred that the variable pair has a high level of similarity for mean, median, standard deviation, 25th and 75th percentile.

4.3. Normalization and skewness test for selected features

Subsequently, the Anderson-Darling test is implemented on the feature pairs to ascertain the normality status of the features. This test is adopted because the dataset has more than 5,000 observations and is a more sensitive test. Additionally, the PDF is a statistical term that is used to characterize the probability distribution of a continuous random variable (Feng et al., 2023). PDF is used to display the nature of distribution as regards normality to enable easy evaluation.

Skewness test is conducted on the three feature pairs. Skewness is a statistical term that describes the asymmetry of a probability distribution. According to Sisodia and Sisodia (2022), it determines whether the data is skewed to the left (negatively skewed), slanted to the right (positively skewed), or essentially symmetric (zero skewness). Skewness is a key tool in analyzing the shape of a dataset's distribution. The Shapiro-Wilk test is used to ascertain skewness in the dataset. It works well with bigger datasets and is not affected by the number of observations. The Shapiro-Wilk test is well-known for its precision and dependability, and it can produce reliable results even when there are a high number of observations. It is frequently recommended as a viable choice for verifying the skewness of continuous data, particularly with big datasets.

The code below shows how the skewness and normality tests are conducted for the features.

```
# Perform the Anderson-Darling test and classify variables as normalized or not
def check_normality(data, significance_level = 0.05):
    normal_vars = []
    non_normal_vars = []
    for column in data.columns:
        statistic, critical_values, significance_levels = anderson(data[column])
        if statistic ≤ critical_values[2]: # Compare the test statistic to the 95% critical value
            normal_vars.append(column)
        else:
            non_normal_vars.append(column)
    return normal_vars, non_normal_vars.
# Check normality of each variable
normal_vars, non_normal_vars = check_normality(standardized_df)
# Plot separate charts for each variable indicating if it's normal or not
for column in standardized_df.columns:
    plt.figure(figsize=(6, 4))
    plt.hist(standardized_df[column], bins = 20, alpha = 0.7, density = True)
    sns.kdeplot(standardized_df[column], color = 'red', label = 'PDF Curve')
    plt.axvline(standardized_df[column].mean(), color = 'green', linestyle = 'dashed', linewidth = 2,
                label = 'Normal' if column in normal_vars else 'Not Normal')
# Add mean, median, and skewness annotations at the right bottom of the chart
    skewness = standardized_df[column].skew()
    plt.text(0.85, 0.05, f'Mean: {standardized_df[column].mean():.2f}\nMedian: {standardized_df[column].median():.2f}\nSkewness: {skewness:.2f}',
             transform = plt.gca().transAxes, fontsize = 10, bbox = dict(facecolor = 'white', alpha = 0.8))
    plt.title(f'Histogram of {column}')
    plt.xlabel('Standardized Value')
    plt.ylabel('Density')
    plt.legend()
    plt.grid(True)
    plt.show()
```

The code conducts the Anderson-Darling test for normality on each variable in a standardized DataFrame (standardized_df). Variables are classified as either normal or non-normal based on the test results. It then generates separate histograms for each variable, overlaying a kernel density estimate (PDF Curve, PDF), mean line, and a label indicating whether the variable is normal or not. Additional annotations include mean, median, and skewness information.

Figs. 5–7 depict the normality and skewness tests for the selected standardized features.

Fig. 5(a) reveals that 1_LT_001_PV is a slightly left-skewed distribution (−0.160630) and its not normally distributed. Fig. 5(b) shows that the variable T0_LEVEL is also a slightly left-skewed distribution (−0.646385) and not normally distributed.

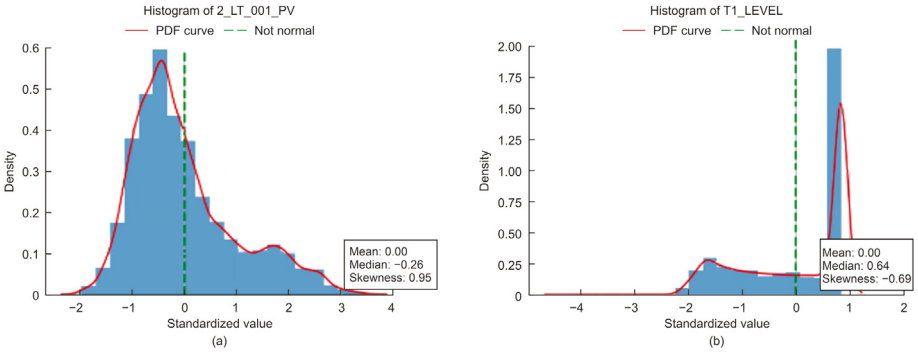

Fig. 6. Normalization and skewness test of variables (a) 2_LT_001_PV and (b) T1_LEVEL.

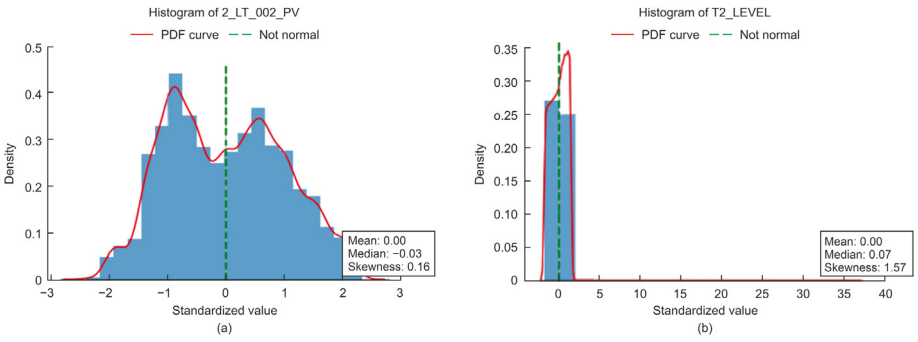

Fig. 7. Normalization and skewness test of variables (a) 2_LT_001_PV and (b) T2_LEVEL.

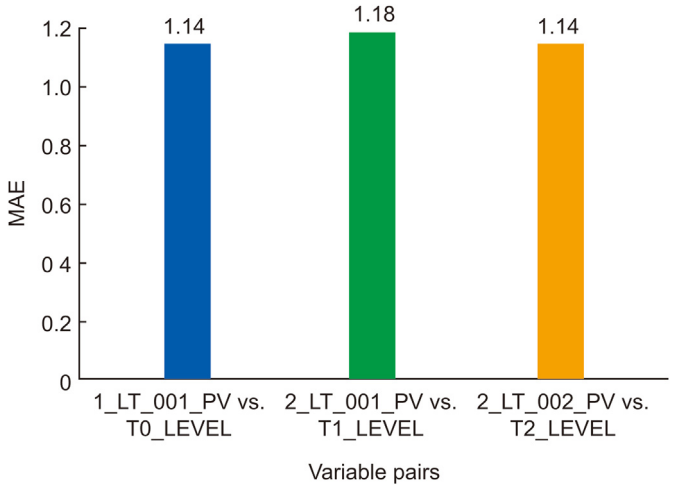

Fig. 8. Mean absolute error (MAE) for each feature pair.

Fig. 6(a) depicts the PDF curve for variable "2_LT_001_PV", and with a significant right-skewed distribution (0.95330) and is not normally distributed. Fig. 6(b) depicts the PDF curve for variable "T1_LEVEL", with a slightly left-skewed distribution (−0.691317) and its not normally distributed.

Fig. 7(a) depicts the PDF curve for variable "2_LT_002_PV", with a relatively weak right-skewed distribution (0.158863) and is not normally distributed. Fig. 7(b) depicts the PDF curve for variable "T2_LEVEL", with a relatively strong right-skewed distribution (1.567730) and not normally distributed. Thus, it can be deduced from Figs. 5(a) and 5(b), Figs. 6(a) and 6(b), and Figs. 7(a) and 7(b) that each variable pair has low similarity with its varying directions as regards skewness (left or right).

4.4. Linearity test for selected features

The study adopts the correlation coefficient test to establish the degree to which the featuree pairs are associated in a linear space. A statistical measure known as the correlation coefficient evaluates the strength and direction of a linear relationship that exists between two continuous variables. It is represented by the symbol "r" and has values ranging from −1 to 1 (Jawabreh et al., 2020). More so, a correlation coefficient of +1 shows a perfect positive linear relationship, which means that the two variables increase concurrently. A correlation coefficient of −1 depicts a complete negative linear relationship, which implies that as one variable increases, the other decreases proportionally. A correlation coefficient close to 0 implies that the variables have a weak or no linear relationship.

The code below shows how the correlation coefficient tests were conducted to ascertain the level of linearity in the feature pairs.

Table 3
Correlation coefficient of variable pairs.

Variable pair	Correlation coefficient
1_LT_001_PV and T0_LEVEL	−0.009
2_LT_001_PV and T1_LEVEL	−0.015
2_LT_002_PV and T2_LEVEL	−0.013

```
# Select the two variables for which you want to calculate the correlation
variable1 = '1_LT_001_PV'
variable2 = 'T0_LEVEL'
# Standardize the selected variables and create a new DataFrame
scaler = StandardScaler()
scaled_df = pd.DataFrame(scaler.fit_transform(df1[[variable1, variable2]]), columns = [f'{variable1}_scaled', f'{variable2}_scaled'])
# Calculate the correlation coefficient between the two standardized variables
correlation_coefficient = scaled_df[f'{variable1}_scaled'].corr(scaled_df[f'{variable2}_scaled'])
```

This code selects two variables, "1_LT_001_PV" and "T0_LEVEL", standardize them, and calculates the correlation coefficient between the standardized versions. The result indicates the strength and direction of the linear relationship between the variables. The process is repeated for the other feature pairs. Furthermore, the three feature pairs are subjected to the correlation coefficient test. Table 3 shows the values obtained.

Table 3 indicates a correlation coefficient of −0.01 for "1_LT_001_PV and T0_LEVEL" and " 2_LT_002_PV and T2_LEVEL" pairs respectively while "2_LT_001_PV and T1_LEVEL" has −0.02 implying an extremely weak and almost negligible negative linear relationship between the feature pairs. This suggests that there is almost no linear association between the features and that the variable pairs have similar values, which could be why the pairs do not increase or decrease simultaneously.

4.5. Similarity distance test

The study adopted the Euclidean distance measure to ascertain the similarity between the feature pairs. The code below shows the process of computing the Euclidean distance between the variable pairs.

```
# Get the two variables of interest
variable1 = '1_LT_001_PV'
variable2 = 'T0_LEVEL'
# Compute the Euclidean distance between the two variables
euclidean_distance = np.linalg.norm(scaled_df[variable1]-scaled_df[variable2])
# Compute the pairwise squared Euclidean distances
pairwise_squared_distances = np.sum((scaled_df[variable1]-scaled_df[variable2]) ** 2)
# Compute the variance between the two variables
variance = np.var(scaled_df[variable1]-scaled_df[variable2])
# Compute the standard deviation between the two variables
```

Table 4
Euclidean distance of variable pairs.

Variable pair	Mean Euclidean distance	Sum of pairwise squared Euclidean distances	Variance	Standard deviation
2_LT_002_PV and T2_LEVEL	246.5455	60,784.7018	2.0260	1.4234
2_LT_001_PV and T1_LEVEL	246.7830	60,901.8729	2.0299	1.4248
1_LT_001_PV and T0_LEVEL	246.0609	60,545.9839	2.0181	1.4206

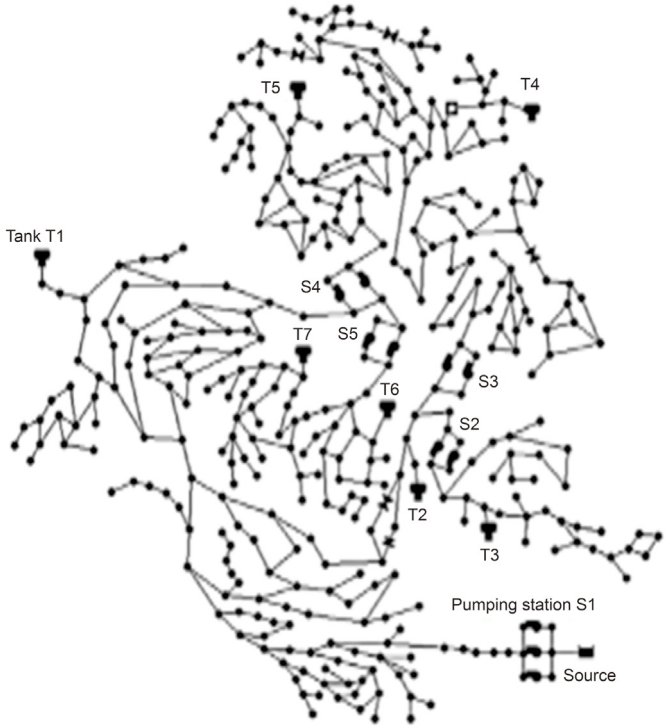

Fig. 9. C-town topology (Taormina et al., 2018).

std_deviation = np.std(scaled_df[variable1]-scaled_df[variable2])

This code calculates various distance metrics and statistical measures between two variables, "1_LT_001_PV" and "T0_LEVEL", that have been standardized. It includes the Euclidean distance, pairwise squared Euclidean distances, variance, and standard deviation between the two variables. The outcome of the similarity is shown in Table 4.

Euclidean distance is a standard measure used to quantify the distance or dissimilarity between two points in a multidimensional space (Roisenzvit, 2023). The mean Euclidean distance between the pairs in Table 4 indicates the average distance between data points of these pairs. Lower values of the mean Euclidean distance imply that the data points are mapped closely together in the multidimensional space, suggesting higher similarity between the variables. Similarly, the sum of pairwise squared Euclidean distances depicted in Table 4 provides an aggregate measure of the overall separation between the data points of the variable pair. A lower sum indicates that the points are more tightly clustered, indicating higher similarity. The variance and standard deviation of the variable pairs also reflect the spread or variability of the data points around the mean Euclidean distance. Lower variance and standard deviation indicate that the data points are relatively close to each other, suggesting higher similarity. The test shows that the pairs are highly similar.

4.6. MAE test

MAE is a statistical measure of errors between two observations reflecting the same phenomena. It is also a metric used to measure the accuracy of a model's predictions. That is, it calculates the average absolute difference between the predicted values and the actual values. Fig. 8 shows values from the computations of the MAE for the variable pairs. The code below shows the process of computing the MAE between the variable pairs.

 # Calculate the MAE for each pair of variables

mae_1 = (df_scaled[predicted_variable_1]-df_scaled[observed_variable_1]).abs().mean()

mae_2 = (df_scaled[predicted_variable_2]-df_scaled[observed_variable_2]).abs().mean()

mae_3 = (df_scaled[predicted_variable_3]-df_scaled[observed_variable_3]).abs().mean()

This code computes the MAE between predicted and observed values for a specific variable pair in a scaled DataFrame.

These MAE values represent the magnitude of the difference between the standardized expected and observed values. Lower MAE values show that the standardized anticipated values are closer to the standardized observed values, whilst higher values indicate that there are more disparities between the standardized predictions and observations. In this case, we have lower values, which suggests that the three variable pairs are similar with minimal error.

4.7. Validation using the C-town topology (physical testbed and DT datasets)

The study validates the approach by applying the framework to another CPS WADI system to establish how effective the proposed framework is and also to ascertain if a level of similarity can be established. Note that the same codes used for the WADI topology test were used for this use case. A C-Town dataset was obtained from Cambrun et al. (2019) for the physical testbed, and its equivalent was generated by DHALSIM (Murillo et al., 2021). The topology of the WADI is depicted in Fig. 9. C-Town comprises 388 junctions, 429 pipes, 7 storage tanks, 5 valves, 11 pumps, and a reservoir.

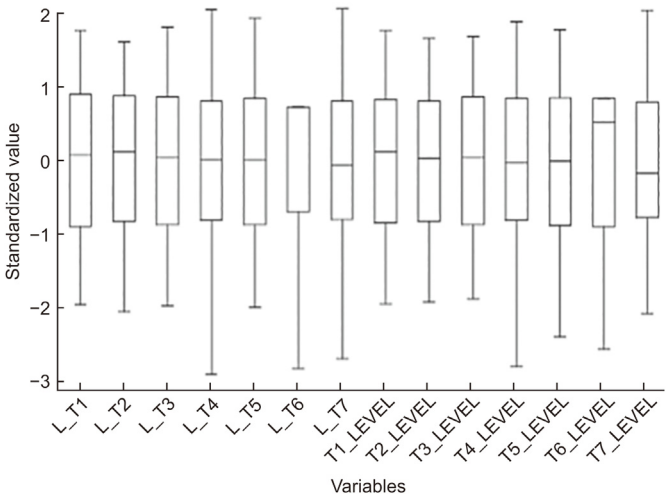

Fig. 10. Boxplot for tank levels (features) from both datasets (C-town) (after outlier removal).

Table 5
Features of the two datasets (C-town).

Count	Physical testbed dataset	Digital twin (DT) dataset
Variable count	45	428
Row count (selected)	5,000	5,000

4.7.1. Dataset features

For validation, the most critical components are selected for comparison, which are the seven major tanks that supply water in the C-Town network. Thus, the study proposes to use only 7 features (tank levels) from each of the datasets to establish the (dis)similarity of the datasets as obtained in the literature (Bochare et al., 2014; Groves, 2015; Zhao et al., 2019), which is also based on the knowledge of the topology. For the purpose of this study, the first 5,000 observations are selected.

From Table 5, it can be observed that the features in the two datasets vary and could be a result of the DTs' simulation of additional functions compared to the physical testbed. This enhances the digital representation by incorporating extra information that might not be easily measurable in the physical system (Flumerfelt et al., 2019).

4.7.2. Standardization and visualization of seven tank variables

As applied in the WADI CPS case, for the C-Town CPS, features of interest are selected, and seven variables (tank levels) of each dataset are standardized to transform the data into a common format, making it consistent and comparable, placing them on a common scale to remove variations that might arise due to different measurement units or scales within the dataset.

Seven feature pairs are considered from both datasets which include "L_T1" and "T1_LEVEL", "L_T2" and "T2_LEVEL", "L_T3" and "T3_LEVEL", "L_T4" and "T4_LEVEL", "L_T5" and "T5_LEVEL", L_T6" and "T6_LEVEL" and "L_T7" and "T7_LEVEL" The preceding features in each pair are from the physical testbed, and the latter variables are from the synthetically generated DT dataset. The datasets were standardized using StandardScaler from scikit-learn, upon pre-processing. This transformed the dataset to possess a mean of 0 and a standard deviation of 1, which were then visualized using a boxplot to explain relationships between pairs. Fig. 10 shows a visual comparison of the pairs upon outlier removal.

By contrasting the pairs in Fig. 10, it can be inferred that very slight pictorial variations exist in the shapes of the boxplot for the feature pairs, with each pair showing a very similar shape. To further analyze, Table 6 shows the interquartile ranges, mean, median and standard deviation of the variable pairs.

Table 6 depicts a high level of similarity in direction (positive or negative) and quantity in values of the mean, median, standard deviation, 25th, and 75th percentiles.

Juxtaposing the statistical measures (mean, standard deviation, and quartiles) for each feature pair. Findings revealed that for "L_T1" and "T1_LEVEL," "L_T2" and "T2_LEVEL," and "L_T3" and "T3_LEVEL" pairs, the quartiles (25th, 50th, 75th) and standard deviation are highly similar in values, but the mean showed some variation in similarity. More so, for "L_T4" and "T4_LEVEL", the quartiles (25th and 75th) and standard deviation are highly similar in values, but the 50th percentile showed some variation in similarity. The "L_T5" and "T5_LEVEL" variable pair showed a very high similarity in values for the mean, standard deviation, and quartiles. Lastly, for the "L_T6" and "T6_LEVEL" and "L_T7" and

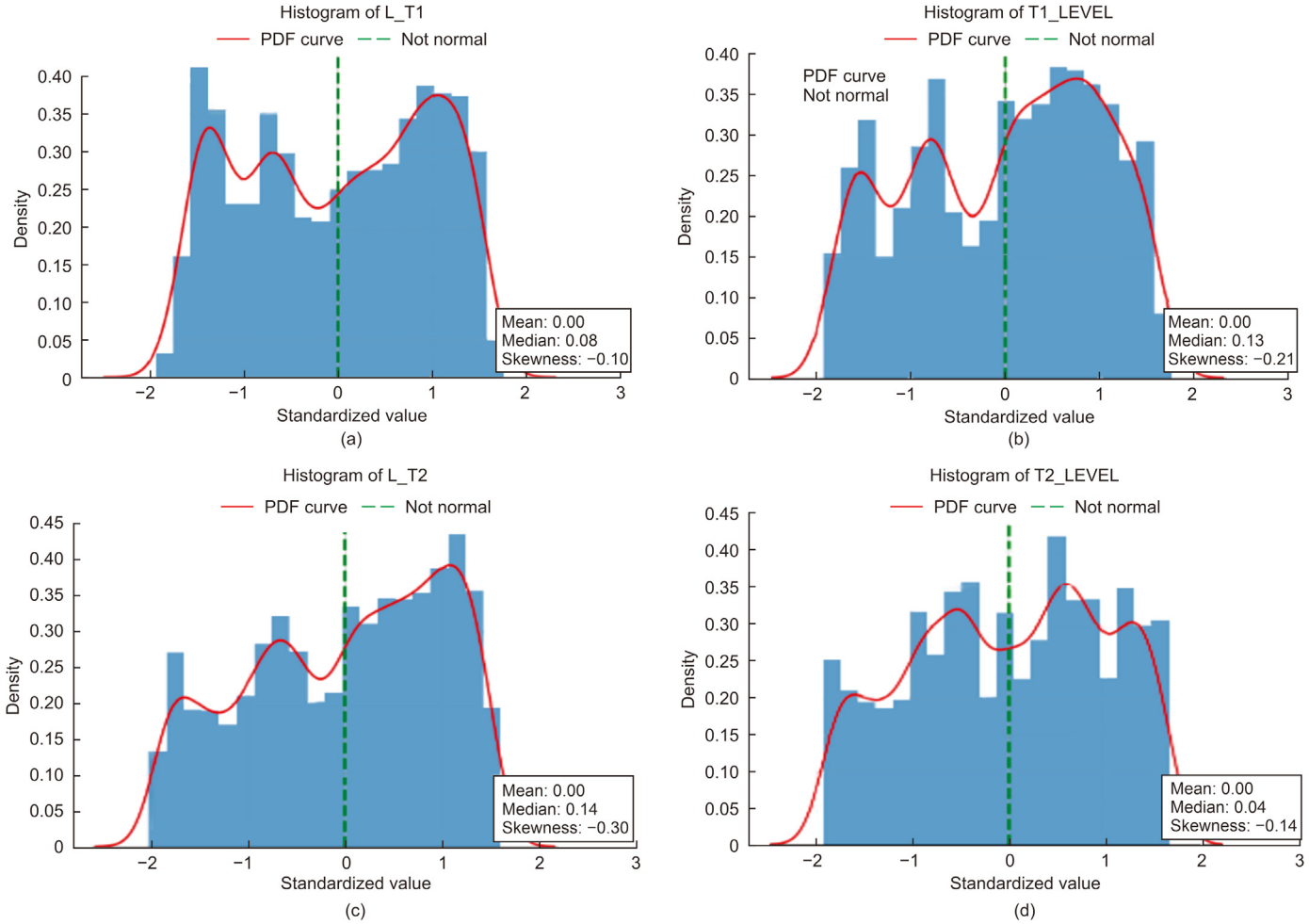

Fig. 11. Normalization and skewness test of variables (a) L_T1", (b) T1_LEVEL", (c) L_T2 and (d) T2_LEVEL.

Table 6
Features of the two datasets (C-town).

Variable	Mean	Standard deviation	25th percentile	50th percentile	75th percentile
L_T1	0.000575	1.000384	−0.87537	0.079703	0.906413
T1_LEVEL	−0.00046	1.000788	−0.83292	0.131777	0.837254
L_T2	−0.00125	1.000361	−0.80343	0.136142	0.878828
T2_LEVEL	0.00004	1.000129	−0.81256	0.034233	0.811887
L_T3	−0.0001	1.00034	−0.85829	0.051974	0.871116
T3_LEVEL	0.000042	1.000298	−0.85233	0.054737	0.871025
L_T4	−0.00031	1.000552	−0.79166	0.025706	0.812818
T4_LEVEL	−0.00023	0.999625	−0.7948	−0.02571	0.848618
L_T5	0.000711	1.000529	−0.85008	0.020058	0.850139
T5_LEVEL	0.000369	0.99985	−0.86316	0.00292	0.858283
L_T6	0.007048	0.990906	−0.67941	0.736366	0.736366
T6_LEVEL	−0.00162	1.000379	−0.87527	0.535144	0.84824
L_T7	0.003172	0.998867	−0.78722	−0.05722	0.818109
T7_LEVEL	−0.00047	1.000201	−0.75807	−0.16198	0.797444

"T7_LEVEL" pairs, the quartiles (25th, 50th, and 75th) and standard deviation are highly similar in values, but the mean showed some variation in similarity.

4.7.3. Normalization and skewness test for selected seven feature pairs

Anderson-Darling test is implemented on the feature pairs to ascertain the normality status of the variables. This test is adopted because the dataset has 5,000 instances and is a more sensitive test compared to other kinds of normality tests. Additionally, the PDF is a statistical term that is used to characterize the probability distribution of a continuous random variable, displaying the nature of distribution relative to normality to enable easy evaluation for the C-town datasets. A skewness test is also conducted on the 7 feature pairs to describe the asymmetry of a probability distribution.

Fig. 11(a) reveals the PDF curve for LT_1. This feature is not normally distributed and is slightly a left-skewed distribution (−0.10). Fig. 11(b) reveals PDF curves for the variable T1_LEVEL, this feature is not normally distributed and is slightly a left-skewed distribution (−0.21). Fig. 11(c) reveals PDF curves for the L_T2. This feature is not normally distributed and is slightly a left-skewed distribution (−0.30). Fig. 11(d) reveals PDF curves for the T2_LEVEL. This feature is not normally distributed and is slightly a left-skewed distribution (−0.14). Figs. 11(a)–12(d) reveal that the PDF curves for the pairs look similarand the feature pairs are not normally distributed, though they have close values and same directions.

Fig. 12(a) reveals that though the PDF curves for the LT_3 is not normally distributed, it has a skewness value of −0.11, which implies that it is slightly a left-skewed distribution. Fig. 12(b) reveals that though the PDF curves for the T3_LEVEL is not normally distributed, it has a skewness value of −0.12, which implies that it is slightly a left-skewed distribution. Fig. 12(c) reveals that though the PDF curves for the L_T4 is not normally distributed, it has a skewness value of −0.17, which implies that it is slightly a left-skewed distribution. Fig. 12(d) reveals that though the PDF curves for the T4_LEVEL is not normally distributed, it has a skewness value of −0.06, which implies that it is slightly a left-skewed distribution. Figs. 12(a)–13(d) reveal that the PDF curves for the pairs look similar and the pairs are not normally distributed having close values and same directions.

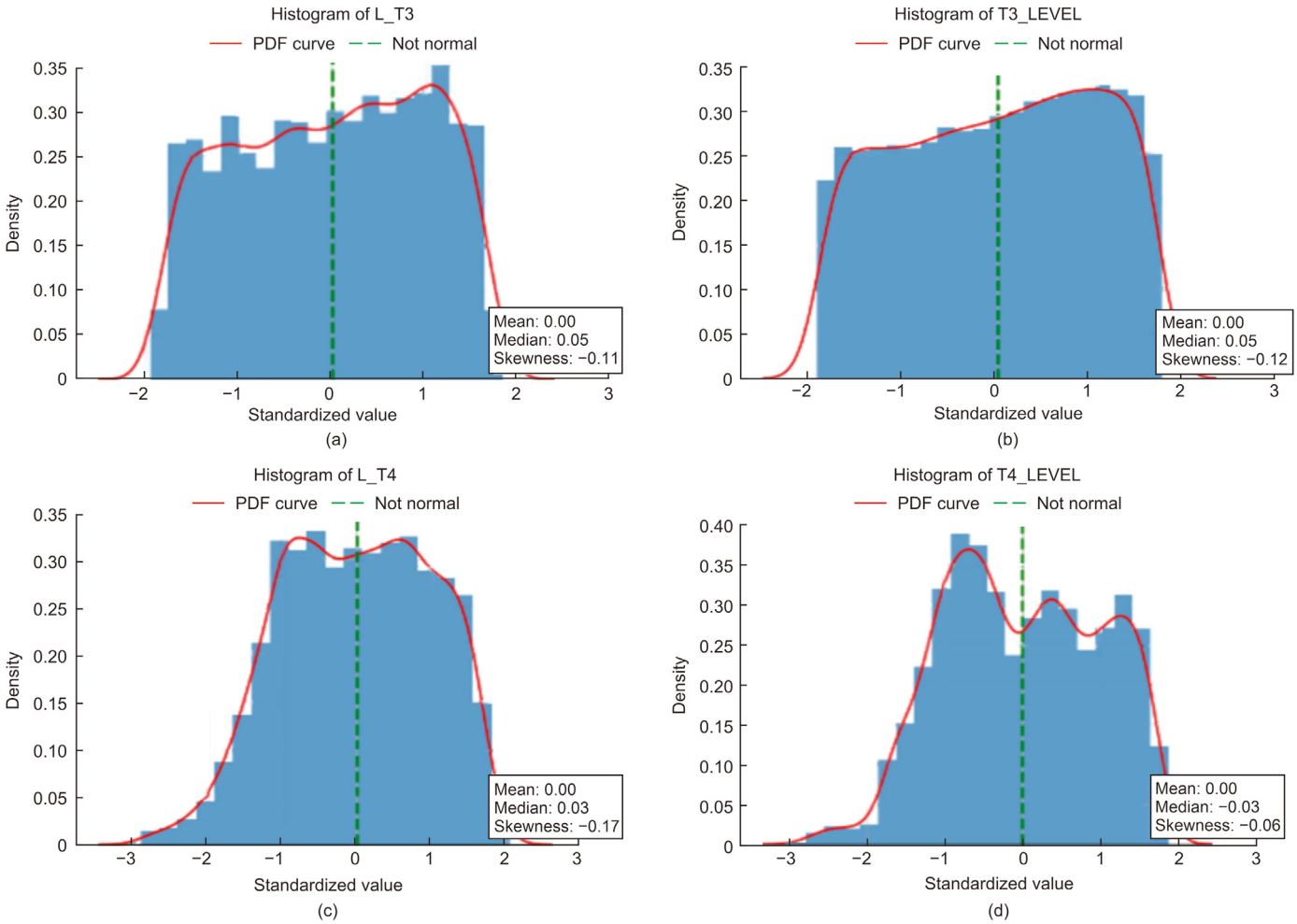

Fig. 12. Normalisation and skewness test of features (a) LT_3, (b) T3_LEVEL, (c) L_T4 and (d) T4_LEVEL.

Fig. 13(a) reveals that though the PDF curves for the L_T5 is not normally distributed, it has a skewness value of −0.01 which implies that it is slightly a left-skewed distribution. Fig. 13(b) reveals that though the PDF curves for the T5_LEVEL is not normally distributed, it has a skewness value of −0.01 which implies that it is slightly a left-skewed distribution. Fig. 13(c) reveals that though the PDF curves for the LT_6 is not normally distributed, it has a skewness value of −1.07 with a significant left-skewed distribution. Fig. 13(d) reveals that though the PDF curves for the T6_LEVEL is not normally distributed, it has a skewness value of −0.76 with a significant left-skewed distribution.

A comparison of Figs. 13 (a)−13 (d) reveals that the PDF curves for all four variables are not normally distributed. Fig. 13 further reveals that the four variables are all left skewed.

Fig. 14(a) reveals that though the PDF curves for the L_T7 is not normally distributed, it has a skewness value of 0.05 with a right-skewed distribution. Fig. 14(b) reveals that though the PDF curves for the T7_LEVEL is not normally distributed, it has a skewness value of 0.28 with a right-skewed distribution. Finally, for the LT_7/T7_LEVEL pair, the PDF curves for the LT_7/T7_LEVEL pairs look similar and are all not normally distributed, though a right-skewed distribution (0.05/0.28) and have close values. This summary provides the skewness values for each variable pair and indicates the direction and strength of skewness in their respective distributions. Thus, it can be deduced that each variable pair has high similarity in its varying directions (left or right).

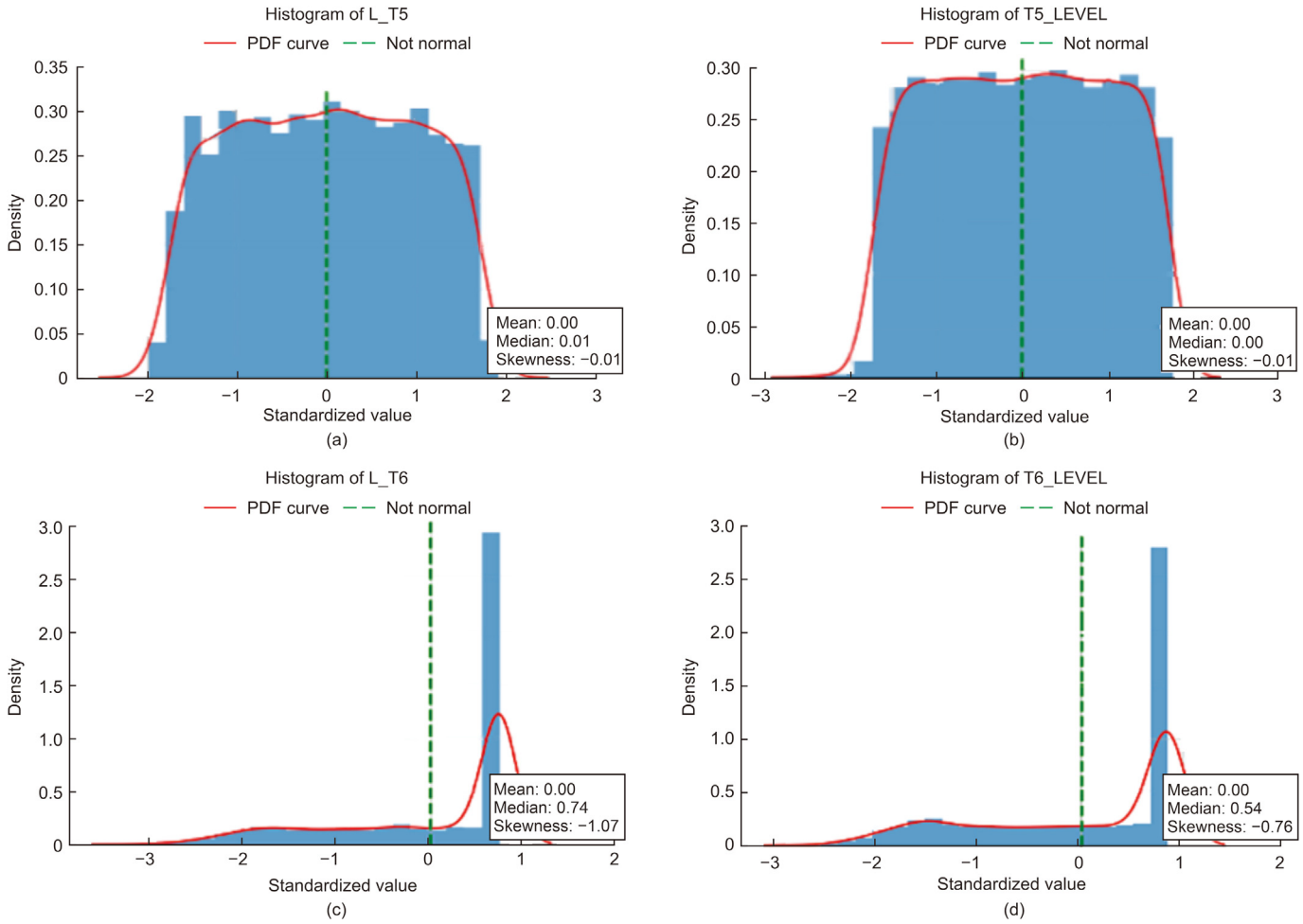

Fig. 13. Normalisation and skewness test of features (a) LT_5, (b) T5_LEVEL, (c) LT_6 and (d) T6_LEVEL.

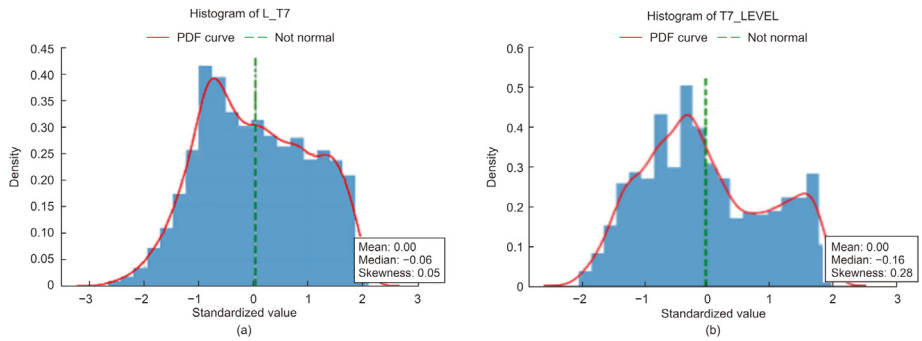

Fig. 14. Normalization and skewness test of features (a) L_T7 and (b) T7_LEVEL.

4.7.4. *Linearity test for selected variables*

The study adopts the correlation coefficient test to establish the degree to which the pairs are associated in a linear space. For validation purposes, seven variable pairs were subjected to the correlation coefficient test. Table 7 shows the values obtained.

From Table 7, it can be inferred that the correlation coefficient between "L_T1" and "T1_LEVEL" is −0.02. This suggests a very weak negative correlation, indicating that as one feature increases, the other tends to decrease slightly, and vice versa. The correlation coefficient between "L_T2" and "T2_LEVEL" is 0.01, indicating a very weak positive correlation. This suggests a minimal tendency for both features to increase or decrease together, although the relationship is not particularly strong. The correlation coefficient between "L_T3" and "T3_LEVEL" is 0.00, signifying no discernible linear correlation between these two variables. Their values do not systematically change in relation to each other. The correlation coefficient between "L_T4" and "T4_LEVEL" is −0.00, suggesting no meaningful linear relationship between these variables. Changes in one variable do not coincide with systematic changes in the other. The correlation coefficient between "L_T5" and "T5_LEVEL" is −0.01, indicating a very weak negative correlation. While there is a slight tendency for one feature to decrease as the other increases, the relationship is minimal. The correlation coefficient between "L_T6" and "T6_LEVEL" is −0.02, revealing a very weak negative correlation. As one feature changes, the other shows a slight tendency to change in the opposite direction. Finally, the correlation coefficient between "L_T7" and "T7_LEVEL" is 0.01, suggesting a very weak positive correlation. While there is a minor tendency for both variables to move together, the relationship is not substantial.

In summary, these correlation coefficients indicate mostly weak and negligible linear relationships between the respective variable pairs. Revelation from Table 7 shows that the seven feature pairs have a very weak and almost negligible negative/positive linear relationship between pairs. This suggests that there is almost no linear association between the features and that the pairs have similar values, which could be why the pairs do not increase or decrease simultaneously.

4.7.5. *Similarity distance test for the C-town datasets*

The study adopted the Euclidean distance measure to ascertain the similarity between the seven variable pairs. The outcome of the similarity is shown in Table 8.

Euclidean distance is a measure used to measure the distance between two points in a multidimensional space. It indicates the average distance between data points, with lower values indicating closer proximity. The sum of pairwise squared Euclidean distances also indicates overall separation, with lower sums indicating tighter clustering. The variance and standard deviation of the variable pairs also reflect the spread or variability around the mean Euclidean distance, with lower values indicating closer proximity. The test indicates high similarity between the variable pairs.

4.7.6. *MAE test*

MAE measures the errors between two observations reflecting the same phenomena and the accuracy of a model's predictions. It computes the average absolute difference between the predicted values and the actual values. Fig. 15 depicts values from the computations of the MAE for the variable pairs.

Table 7
Correlation coefficient of variable pairs.

Variable pair	Correlation coefficient
L_T1 and T1_LEVEL	−0.02
L_T2 and T2_LEVEL	0.01
L_T3 and T3_LEVEL	0.00
L_T4 and T4_LEVEL	−0.00
L_T5 and T5_LEVEL	−0.01
L_T6 and T6_LEVEL	−0.02
L_T7 and T7_LEVEL	0.01

Table 8
Euclidean distance of variable pairs.

Variable pair	Mean Euclidean distance	Sum of pairwise squared Euclidean distances	Variance	Standard deviation
L_T1 and T1_LEVEL	101.0245	10,205.9585	2.0408	1.4286
L_T2 and T2_LEVEL	99.3684	9874.0789	1.9744	1.4051
L_T2 and T2_LEVEL	99.9427	9988.5490	1.9973	1.4133
L_T4 and T4_LEVEL	100.0372	10,007.4315	2.0011	1.4146
L_T5 and T5_LEVEL	100.3457	10,069.2586	2.0134	1.4190
L_T6 and T6_LEVEL	101.1055	10,222.3322	2.0441	1.4297
L_T7 and T7_LEVEL	99.4222	9884.7671	1.9766	1.4059

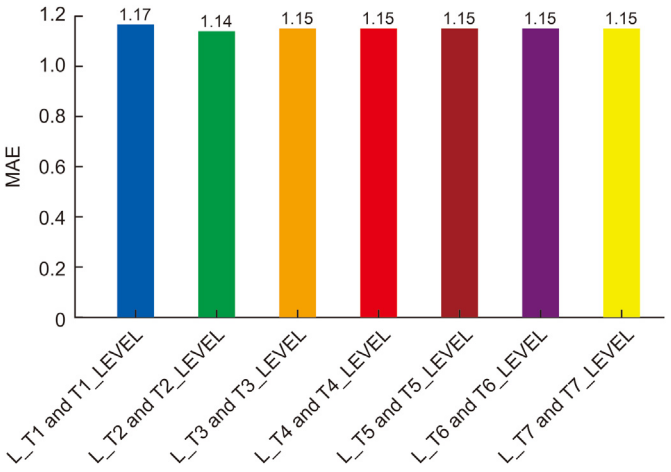

Fig. 15. Mean absolute error (MAE) for each variable pair.

Fig. 15 shows the values obtained from the experiment. MAE values represent the magnitude of the difference between the standardized expected and observed values. The low MAE values show that the standardized anticipated values are closer to the standardized observed values, whilst the higher values indicate that there are more disparities between the standardized predictions and observations. In this case, we have low values, which suggests that the seven variable pairs are similar but with minimal error.

5. Practical implication of findings on CPSs

This research explored the physical testbed and its equivalent using datasets obtained from the two scenarios (WADI and C-Town). Findings revealed a high similarity for the tests conducted. When the mean, quartiles, and standard deviation have similar values for two datasets, it implies that the central tendency, spread, and distributional characteristics of the datasets are comparable. For the tanks, the test conducted implies identifying if the distribution of water from the tanks to their destinations is (un)stable for each tank, thus ensuring normal WADI between the tanks and their final destination. The lower the standard deviation, the more efficient the WADI, and a higher standard deviation implies significant variability or inefficiency, potential issues such as a cyber-attack.

Decision-makers can expect similar typical values for the physical testbed and its digital equivalent, making it easier to compare and analyze individual features (sensors/actuators/tanks, etc) from the datasets. More so, quartile comparisons of water levels can help assess the consistency and variability in the system's performance and may

also indicate (un)stable and (un)reliable or variability of WADI. The standard deviation is a measure of the amount of variation or dispersion of a set of values. Thus, standard deviation has a multi-faceted impact on WADI systems, influencing water quality, system reliability, and design uncertainties within the network. Understanding and managing the standard deviation of relevant parameters is crucial for ensuring the efficient and effective operation of WADI systems. For demand fluctuations, analyzing standard deviation in WADI over specific periods can help utilities plan for peak demand periods and allocate resources efficiently.

For skewness, readings can indicate whether the distribution is symmetric or if there is a tendency for water levels to be more extreme on one side of the distribution. Positive skewness indicates a longer right tail, while negative skewness indicates a longer left tail. This simply indicates where there is more pressure or concentration (low, average, or high) for WADI or the behavior of WADI. Euclidean distance also suggests that the behavior of tanks of both objects is remarkably similar, implying that for any anticipated improved performance, the digital equivalent could be used to anticipate the behavior and project results. This eases planning and scalability. The MAE gives an insight into the amount of error to be expected between operations of the two objects. The lesser the error, the more efficient the system operations.

6. Conclusion

The study established a comparison framework for WADI cyber-physical testbed and its DT counterpart to ascertain how similar they are. Datasets were sourced from iTrust for the physical asset and synthetically generated datasets from DHALSIM, and a series of statistical tests were conducted using machine leaning. Three critical elements (one primary and two elevator reservoir tanks for WADI) were considered for this similarity test because the primary tank provides water to the entire network distribution system, and the two elevated reservoir tanks store water for onward transmission to consumers. Tests conducted revealed that the two datasets had some level of similarity. As regards the row count, 30,000 rows were used for each test, but the DT dataset had one extra variable when compared to its physical counterpart dataset. Standardization was applied to both datasets to scale them to the same level. Box plots also revealed a proficient level of similarity between each variable pair. The three variable pairs also displayed skewness (left and right) and non-normality for each variable pair. The correlation coefficient test was used to ascertain the linearity of the variable pairs, which showed to be very weak or non-existent because the value obtained from the test was close to zero. Euclidean distance was also adopted and applied to the feature pairs to discover a significant similarity in the pairs with minimal variations. Finally, the proposed approach was implemented/validated on C-Town datasets (synthetically generated and generated from the testbed), which further proved to be efficient since similarity was established for most of the tests conducted on the feature pairs. The study established that in the development, adoption, and integration of a DT for application in the real world, there needs to be a moderate to a high level of similarity between the data generated from the physical object and its DT counterpart as this will improve data quality that reflects the physical object and help in decision-making processes on the physical object for intrusion detection or improved manufacturing processes. The research can be extended to accommodate more variables, and a quantifiable similarity index could be used to compare physical objects and DTs.

CRediT authorship contribution statement

Henry Chima Ukwuoma: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Data curation. **Gilles Dusserre:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Conceptualization. **Gouenou Coatrieux:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Data curation, Conceptualization. **Johanne Vincent:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The WADI datasets were provided by iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design. The C-town dataset was obtained from BATADAL (https://www.batadal.net/data.html).

References

Aheleroff, S., Xu, X., Zhong, R.Y., et al., 2021. Digital twin as a service (DTaaS) in industry 4.0: an architecture reference model. Adv. Eng. Inf. 47 (Jan.), 101225.
Attaran, M., Celik, B.G., 2023. Digital Twin: benefits, use cases, challenges, and opportunities. Decis. Anal. J. 6 (Jan.), 100165.
Bochare, A., Gangopadhyay, A., Yesha, et al., 2014. Integrating domain knowledge in supervised machine learning to assess the risk of breast cancer. Int. J. Med. Eng. Inf. 6 (2), 87.
Boyes, H., Watson, T., 2022. Digital twins: an analysis framework and open issues. Comput. Ind. 143 (Aug.), 103763.
Braunegg, A., Chakraborty, A., Krumdick, M., et al., 2020. APRICOT: a dataset of physical adversarial attacks on object detection. In: Computer Vision–ECCV 2020: 16th European Conference. Springer International Publishing, Glasgow, UK, pp. 35–50.
Cambrun, M.B., Hankin, C., 2019. Assessing cyber-physical security in industrial control systems. In: 6th International Symposium for ICS & SCADA Cyber Security Research 2019. ICS-CSR).
Catapult, H.V., 2021. Untangling the Requirements of a Digital Twin. Available at: https://www.amrc.co.uk/files/document/406/1605271035_1604658922_AMRC_Digital_Twin_AW.pdf.
Dattalo, P., 2013. Multivariate multiple regression. In: Dattalo, P. (Ed.), Analysis of Multiple Dependent Variables. Oxford University Press, Oxford.
de Gois, G., de Oliveira-Júnior, J.F., da Silva Junior, C.A., et al., 2020. Statistical normality and homogeneity of a 71-year rainfall dataset for the state of Rio de Janeiro—Brazil. Theor. Appl. Climatol. 141, 1573–1591.
Dickie, I.A., Boyer, S., Buckley, H.L., et al., 2018. Towards robust and repeatable sampling methods in eDNA-based studies. Mol. Ecol. Resour 18 (5), 940–952.
Doug, B., 2021. The principle of least privilege in federal agencies: implementing RBAC. Technol. Solut. Drive Gov. Available at: https://fedtechmagazine.com/article/2021/10/principle-least-privilege-federal-agencies-implementing-rbac-perfcon.
Ercetin, O., 2023. Computational and communication aspects of digital twin: an information theoretical perspective. IEEE commun. Letture 27 (2), 492–496.
Falah, M.F., Sukaridhoto, S., Al Rasyid, M.U.H., et al., 2020. Design of virtual engineering and digital twin platform as implementation of cyber-physical systems. Procedia Manuf. 52, 331–336.
Farine, D.R., Carter, G.G., 2022. Permutation tests for hypothesis testing with animal social network data: problems and potential solutions. Methods Ecol. Evol. 13 (1), 144–156.
Flumerfelt, S., Schwartz, K.G., Mavris, D., et al. (Eds.), 2019. Complex Systems Engineering: Theory and Practice. American Institute of Aeronautics and Astronautics, Inc., Reston.
Gabel, T., Godehardt, E., 2015. Top-down induction of similarity measures using similarity clouds. In: Hüllermeier, E., Minor, M. (Eds.), Case-Based Reasoning Research and Development, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 149–164.
Gal, M.S., Rubinfeld, D.L., 2019. Data standardization. NYUL Rev. 94 (4), 737–770.
Groves, W.C., 2015. Toward Automating and Systematizing the Use of Domain Knowledge in Feature Selection (PhD Thesis). University of Minnesota.
Hanoun, T.M., Hashim, K.M., 2019. Modify manhattan distance for image similarity: new measurement for image similarity. Open J. Sci. Technol. 2 (4), 12–16.
Huo, J., Ma, Y., Lu, C., et al., 2021. Mahalanobis distance based similarity regression learning of NIRS for quality assurance of tobacco product with different variable selection methods. Spectrochim. Acta. A. Mol. Biomol. Spectrosc. 251 (Apr.), 119364.
Hussain, MdM., Mahmud, I., Bari, S.H., 2023. pyHomogeneity: a Python package for homogeneity test of time series data. J. Open Res. Software 11 (1), 1–5.
IBM, 2023. What Is a Digital Twin? IBM. Available at: https://www.ibm.com/topics/what-is-a-digital-twin.
iTrust, 2018. iTrust Labs_WADI. iTrust. Available at: https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_wadi/.
Jawabreh, O., Mahmoud, R., Hamasha, S.A., 2020. Factors influencing the employees service performances in hospitality industry case study AQBA five stars hote. Geoj. Tour. Geosites 29 (2), 649–661.
Khan, T.H., Noh, C., Han, S., 2023. Correspondence measure: a review for the digital twin standardization. Int. J. Adv. Manuf. Technol. 128 (Aug.), 1907–1927.

Lenhard, W., Lenhard, A., 2017. Computation of Effect Sizes. https://doi.org/10.13140/RG.2.2.17823.92329.
Liu, Q., Liu, W., Tang, J., et al., 2019. Permutation-test-based clustering method for detection of dynamic patterns in Spatio-temporal datasets. Comput. Environ. Urban Syst. 75 (Feb.), 204–216.
Mallikharjuna, R.K., Saikrishna, G., Supriya, K., 2023. Data preprocessing techniques: emergence and selection towards machine learning models-a practical review using HPA dataset. Multimed. Tool. Appl. 82 (Mar.), 37177–37196.
Mihai, S., Yaqoob, M., Hung, D.V., et al., 2022. Digital twins: a survey on enabling technologies, challenges, trends and future prospects. IEEE Commun. Surv. Tutor. 24 (4), 2255–2291.
Moi, T., Cibicik, A., Rølvåg, T., 2020. Digital twin based condition monitoring of a knuckle boom crane: an experimental study. Eng. Fail. Anal. 112 (Mar.), 1–10.
Murillo, A., Taormina, R., Tippenhauer, N., et al., 2021. Co-simulating physical processes and network data for high-fidelity cyber-security experiments. In: Sixth Annual Industrial Control System Security (ICSS) Workshop, ICSS 2020. Association for Computing Machinery, New York, USA, pp. 13–20.
Perno, M., Hvam, L., Haug, A., 2022. Implementation of digital twins in the process industry: a systematic literature review of and barriers. Comput. Ind. 134 (Nov.), 1–16.
Rasmussen, K., Kondrup, J.B., Allard, A., et al., 2015. Novel mathematical and statistical approaches to uncertainty evaluation: best practice guide to uncertainty evaluation for computationally expensive models. Brunsw. Ger. Euramet.
Rodríguez del Águila, M.M., Benítez-Parejo, N., 2011. Simple linear and multivariate regression models. Allergol. Immunopathol. 39 (3), 159–173.
Roisenzvit, A.B., 2023. From euclidean enablers distance to spatial classification: unraveling the technology behind GPT models. Universidad del CEMA.
Salauddin, K.M., Nath, T.D., Hossain, Murad, et al., 2023. Comparison of multiclass classification techniques using dry bean dataset. Int. J. Cogn. Comput. Eng. 4 (Jan.), 6–20.
Schleich, B., Anwer, N., Mathieu, L., et al., 2017. Shaping the digital twin for design and production engineering. CIRP Ann. 66 (1), 141–144.
Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y., 2015. A comparison study on similarity and dissimilarity measures in clustering continuous data. PLoS One 10 (12), 1–20.
Sisodia, D., Sisodia, D.S., 2022. Quad division prototype selection-based k-nearest neighbor classifier for click fraud detection from highly skewed user click dataset. Eng. Sci. Technol. Int. J. 28 (Jun.), 1–12.
Taormina, R., Galelli, S., Tippenhauer, N.O., et al., 2018. Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks. J. Water Resour. Plan. Manag. 144 (8), 04018048.
Varghese, S.A., Ghadim, A.D., Balador, A., P, et al., 2022. Digital twin-based intrusion detection for industrial control systems. In: 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops), pp. 611–617.
Wang, Z., Luo, N., Zhou, P., 2020. GuardHealth: blockchain empowered secure data management and Graph Convolutional Network enabled anomaly detection in smart healthcare. J. Parallel Distr. Comput. 142 (Apr.), 1–12.
Wright, L., Davidson, S., 2020. How to tell the difference between a model and a digital twin. Adv. Model. Simul. Eng. Sci. 7 (Dec.), 1–13.
Yang, C.-Y., Chen, P.-Y., Wen, T.-J., et al., 2019. IMU consensus exception detection with dynamic time warping—a comparative approach. Sensors 19 (10), 1–19.
Yang, X., Liu, H., Wang, Z., et al., 2022. Zebra: Deeply Integrating System-Level Provenance Search and Tracking for Efficient Attack Investigation. Available at: https://arxiv.org/abs/2211.05403.
Yao, Y., Wang, Z., Zhou, P., 2020. Privacy-preserving and energy efficient task offloading for collaborative mobile computing in IoT: an ADMM approach. Comput. Secur. 96 (May), 1–10.
Zhao, J., Karimzadeh, M., Masjedi, A., et al., 2019. FeatureExplorer: interactive feature selection and exploration of regression models for hyperspectral images. In: 2019 IEEE Visualization Conference (VIS). Presented at the 2019 IEEE Visualization Conference (VIS), pp. 161–165.