2-1-2023

Using artificial intelligence to detect human errors in nuclear power plants: A case in operation and maintenance

Ezgi Gursel
The University of Tennessee, Knoxville

Bhavya Reddy
San Jose State University

Anahita Khojandi
The University of Tennessee, Knoxville

Mahboubeh Madadi
San Jose State University, mahboubeh.madadi@sjsu.edu

Jamie Baalis Coble
The University of Tennessee, Knoxville

See next page for additional authors

Recommended Citation

Ezgi Gursel, Bhavya Reddy, Anahita Khojandi, Mahboubeh Madadi, Jamie Baalis Coble, Vivek Agarwal, Vaibhav Yadav, and Ronald L. Boring. "Using artificial intelligence to detect human errors in nuclear power plants: A case in operation and maintenance" *Nuclear Engineering and Technology* (2023): 603-622. https://doi.org/10.1016/j.net.2022.10.032

Authors

Ezgi Gursel, Bhavya Reddy, Anahita Khojandi, Mahboubeh Madadi, Jamie Baalis Coble, Vivek Agarwal, Vaibhav Yadav, and Ronald L. Boring

Nuclear Engineering and Technology 55 (2023) 603–622

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Using artificial intelligence to detect human errors in nuclear power plants: A case in operation and maintenance

Ezgi Gursel [a, 1], Bhavya Reddy [b, 1], Anahita Khojandi [a, *], Mahboubeh Madadi [c], Jamie Baalis Coble [d], Vivek Agarwal [e], Vaibhav Yadav [e], Ronald L. Boring [e]

[a] Department of Industrial and Systems Engineering, University of Tennessee, Knoxville, TN, 37996, USA
[b] Department of Computer Science, San Jose State University, San Jose, CA, 95192, USA
[c] Department of Marketing and Business Analytics, San Jose State University, San Jose, CA, 5192, USA
[d] Department of Nuclear Engineering, University of Tennessee, Knoxville, TN, 37996, USA
[e] Idaho National Laboratory, PO Box 1625, Idaho Falls, ID, 83415, USA

ARTICLE INFO

Article history:
Received 17 June 2022
Received in revised form
23 September 2022
Accepted 23 October 2022
Available online 29 October 2022

Keywords:
human error detection
anomaly detection
nuclear power plants
machine learning

ABSTRACT

Human error (HE) is an important concern in safety-critical systems such as nuclear power plants (NPPs). HE has played a role in many accidents and outage incidents in NPPs. Despite the increased automation in NPPs, HE remains unavoidable. Hence, the need for HE detection is as important as HE prevention efforts. In NPPs, HE is rather rare. Hence, anomaly detection, a widely used machine learning technique for detecting rare anomalous instances, can be repurposed to detect potential HE. In this study, we develop an unsupervised anomaly detection technique based on generative adversarial networks (GANs) to detect anomalies in manually collected surveillance data in NPPs. More specifically, our GAN is trained to detect mismatches between automatically recorded sensor data and manually collected surveillance data, and hence, identify anomalous instances that can be attributed to HE. We test our GAN on both a real-world dataset and an external dataset obtained from a testbed, and we benchmark our results against state-of-the-art unsupervised anomaly detection algorithms, including one-class support vector machine and isolation forest. Our results show that the proposed GAN provides improved anomaly detection performance. Our study is promising for the future development of artificial intelligence based HE detection systems.

1. Introduction

Human error (HE) poses risk to any industry that relies on humans to complete tasks. The consequences of HE can be widespread and significant, ranging from financial losses and system unavailability to loss of life [1]. HE is an especially important concern in safety-critical industries and settings, such as aviation, transportation, healthcare, and nuclear power plants (NPPs). As seen in the past accidents at NPPs, failure of safety-critical systems can result in loss of life, economic damage, widespread health risks, and environmental contamination [2].

HE is of particular interest regarding NPPs as they can be linked to many system failures and accidents in history, including the Three Mile Island accident (TMI), Chernobyl, and Fukushima accidents [2,3]. According to [4], approximately 70% of NPP operation-related errors appear to be directly or indirectly result from HE. Furthermore, according to the studies from the Institute of Nuclear Power Operation (INPO), about 48% of all events in NPPs from 2010-2011 were a result of HE [5]. Deadly accidents and HE-induced incidents in safety-critical systems have sparked the interest of researchers across various disciplines in regard to finding ways to manage and mitigate the effects of HE, especially in NPPs.

HE in NPPs can result from a variety of factors (e.g., personal and environmental stressors, task complexity, lack of training or experience for the role, ergonomics, and complex or unfamiliar user interfaces) [6–8]. Human operators in NPPs are faced with a large number of tasks, including monitoring the status of plant instruments, observing specific plant areas, and taking measurements [9]. Furthermore, when system conditions deviate from normal, many alarms may send out warning messages simultaneously, causing cognitive overload for the operator [10].

[*] Corresponding author.
E-mail address: khojandi@utk.edu (A. Khojandi).
[1] Authors contributed equally.

https://doi.org/10.1016/j.net.2022.10.032

Given this stressful taskload and the importance of the stable operation of the NPP, HE can arise in various ways in NPPs (e.g., when an operator performs a task from memory, does not understand the instructions clearly, or omits parts of a procedure) [4,11]. Other common types of HE in NPPs include forgetting to perform tasks or incorrectly diagnosing system state [12]. Given these opportunities for human error, it is critical to find ways to both assist human operators with their tasks and alert them of the potential presence of HE.

In recent years, the use of artificial intelligence (AI) has received significant attention. AI has been applied across many domains to solve complex problems. As data volumes increase, AI systems for digesting these data are becoming increasingly prevalent. The sheer volume of data collected in NPPs makes them valuable locations for incorporating AI-based technologies, though the high-stakes environment of NPPs make implementation a slow and cautious process. Balanced usage of AI technologies in NPPs can improve plant productivity and efficiency [13], lower long-term operating and maintenance costs [14], and reduce opportunities for HE [15]. AI has been applied in NPPs in various ways, including in operator decision-making support systems [16–18] and real-time monitoring and automation systems [19,20].

One branch of AI is anomaly detection, which involves finding data patterns that differ from the expected behavior [21]. The challenge lies in cases where some samples differ, based on a given metric, from the rest of the dataset. Depending on the application domain, these anomalous patterns may be referred to as outliers, anomalies, or faults in the dataset. In recent years, anomaly detection has attracted the attention of the research community due to the relevance of its implementation in real-world applications, such as image processing, intrusion detection, fraud detection, and system health monitoring, among many others [21].

Anomaly detection algorithms are broadly categorized as either supervised, unsupervised, or semi-supervised. The differences lies in the amount of labeled data required for training the algorithm. Specifically, supervised algorithms require that the data be fully labeled, unsupervised algorithms do not require labeled data, and semi-supervised algorithms require only that a small subset of the data to be labeled. In supervised anomaly detection, both non-anomalous (i.e., normal) and anomalous data are labeled. The algorithm can build a predictive model based on the given labels. While supervised anomaly detection techniques are prevalent in the recent literature, they are generally hampered by the need for large datasets of labeled data to train the algorithms. This is further complicated by the fact that anomalies may come in different types. Hence, training supervised models to detect each of these anomaly types would require very large datasets containing adequate representation from each anomaly type.

Given the challenges faced in training supervised algorithms, unsupervised or semi-supervised anomaly detection algorithms may often be more appropriate. In unsupervised anomaly detection, the observations do not have labels. The algorithm trains based on the assumption that non-anomalous data will be more common in the dataset than anomalous instances [22]. If this assumption is untrue, unsupervised methods can suffer from a high false positive rate, with non-anomalous points being classified as anomalous instances. Semi-supervised anomaly detection falls somewhere between unsupervised and supervised models, and takes advantages of whatever labeled data are present [23].

Anomaly detection is especially important in NPPs, since NPP safety is highly dependent on stable operation. An anomaly in a safety-critical system, such as an NPP, can indicate atypical conditions that can impact system performance [24]. NPPs store a large amount of data collected from various sources, including equipment, condition reports, maintenance logs, and process instrumentation and control, to name a few [25]. Anomalies can occur in various areas of the plant, for various reasons, including instrument error and HE [24]. Some of these anomalies may be hard for operators to detect, or as aforementioned, even be caused by the operator. As such, AI allows for correlating all data and fusing the data collected from various sources in the NPP in order to detect potential anomalies in manually collected data.

In the context of NPPs, HEs can be treated as anomalies. In general, the presence of HE through erroneous data collection can result in data points that do not conform to usual patterns [26]. Although the risk associated with HEs can be significant, the overall likelihood of occurrence of a consequential HE is rather low, given the extensive training and expertise of NPP personnel. According to [12], the HE rate in an NPP is not expected to exceed 10%. Hence, anomaly detection methods can serve as promising tools to detect the presence of HE in NPPs.

In this study, we take a data-driven look at the issue of identifying HEs in NPPs via anomaly detection. Specifically, we develop an unsupervised HE anomaly detection model based on a generative adversarial network (GAN) and compare it against state-of-the-art anomaly detection benchmarks, including one-class support vector machine (OCSVM) and isolation forest (iForest). Specifically, by using an unlabeled dataset provided by an operational NPP, we analyze automatically collected sensor data and manually collected vibration data from an NPP balance of plant. Under the assumption that sensor data are non-anomalous, we developed an approach to correlate automatically collected and manually collected data in order to identify mismatches (i.e., anomalies) possibly arising from manual data collection (HE) and evaluate the model performance under various assumed anomaly rates, bounded above at 10%. Furthermore, we validated our approach on an external dataset collected from a testbed, with known labels and intentionally introduced erroneous data during collection process associated with HE. To the best of our knowledge, our study is the first to develop a GAN-based unsupervised anomaly detection technique for HE anomaly detection in NPPs. The framework developed in this study has the potential to foster HE mitigation by immediately alerting human operators of potential errors and prompting operators to further evaluate or retake measurements. Additionally, it can lay the foundation for future studies addressing HE identification and developing a human-in-the-loop (HITL) system for smooth NPP operations.

The rest of the manuscript is organized as follows. Section 2 reviews the related literature on HEs and unsupervised anomaly detection. Section 3 presents the data and methods. Section 4 presents the results, and Section 5 discusses the findings. Finally, Section 6 concludes the paper and provides future directions.

2. Literature review

Due to the impact of HEs on plant safety and operations, HE identification and mitigation have been the subject of extensive research [16,27]. For example, studies have investigated the correlation between performance shaping factors and errors in NPPs [28]. Other studies developed various frameworks for evaluating the effects of maintenance-related HEs in NPPs [29] and for identifying accidents that the NPP's operator support system cannot diagnose with certainty [30]. [31] considered the use of SACADA (Scenario Authoring, Characterization, and Debriefing Application) and HuREX (human reliability data extraction) databases for HE probability estimation. Other works such as [32] and [33] also considered the issue of calculating HE probabilities.

Recent literature has explored various approaches for using anomaly detection, a subset of AI, across many domains. Anomaly detection has been covered extensively in the recent literature, and surveyed in various works such as [21] and [34]. Anomaly detection techniques are generally classified into two main categories (i.e., model-based and data-driven [35]) and are further subdivided into distribution, distance, density, clustering, and classification techniques [36] based on how they categorize instances as anomalous anomalies. Numerous efforts have been made to detect anomalous sensor data in safety-critical systems by using different data-driven techniques. Due to the complexity of data dimensions and the scarcity of labeled data, much of the existing anomaly detection research employs a combination of different anomaly detection techniques. In [36], the authors proposed the combination of yet another segmentation algorithm (i.e., YASA), a novel fast and high-quality segmentation algorithm, with an OCSVM approach for efficient anomaly detection in turbomachines in the petroleum industry. OCSVM learns a region containing all training data instances (a boundary) and flags as anomalies any instances that fall outside the boundary. In [37], the authors compared different machine learning (ML) methods to estimate power output performance and detect anomalies in a combined-cycle power plant, based on 5 years of recorded data and using autoencoders, SVMs, random forest, and iForest.

Most current studies on anomaly detection in NPPs relate to fault detection and diagnosis [38], a review of which is provided in [35,39]. In supervised anomaly detection, the authors of [40] apply anomaly detection for condition monitoring in NPPs by using symbolic dynamic filtering (SDF). The results of this study are compared with principal component analysis, a popular data-driven method. The authors found that SDF-based anomaly detection outperformed principal component analysis, proving that SDF can be a useful tool for real-time anomaly detection. Recently, the authors of [41] used artificial neural networks (ANN) - more specifically, recurrent neural networks, which incorporate historical anomalies in the training-as a means of detecting anomalous sensor signals. Semi-supervised and unsupervised anomaly detection have also been explored in the recent literature. In [42], a semi-supervised variational graph autoencoder method was proposed for identifying system-level anomalies in NPP data. In [43], a deep learning approach using a combination of convolutional neural networks (CNN), *k*-means clustering, and denoising autoencoders was used to unfold nuclear power reactor signals. Other approaches to anomaly detection in NPPs include using Kalman filters for instrument failure detection [44].

ML and anomaly detection have also been specifically considered in the context of HE detection/evaluation. In [45], an unsupervised anomaly detection model was developed to evaluate data quality, which could be negatively impacted by manual data entry and/or measurement errors. The authors of [46] predicted HE using anomaly detection based on Shallow CNN through analyzing the human operators' electroencephalography (EEG) signals. In the context of NPPs [47], considered the use of a long short-term memory variational autoencoder (LSTM-VAE) based anomaly detection model to detect system and component anomalies to reduce HE in NPP diagnostic tasks. Another study [48], proposed the use of ANNs to predict the trends of 55 plant parameters and detect the presence of HE. In [49], the authors developed a framework based on deep neural networks and colored Petri nets to determine operator errors with the goal of reducing HE in NPP operations.

In recent years, the use of GANs for anomaly detection has gained traction. The ability of GANs to model high-dimensional data make them suitable candidates for anomaly detection [50]. The use of GANs in anomaly detection has been surveyed in various works such as in [51] and [52]. The first GAN was proposed by [53] and [54] first proposed using GANs for unsupervised anomaly detection (AnoGAN). Other variations of GAN-based anomaly detection techniques have since been proposed. The authors of [55] proposed MAD-GAN for detecting cyber-intrusion-caused anomalies in time series data. In [56], the authors applied GAN-based anomaly detection to multivariable time series data in a power plant [50]. considered the application of BiGAN as a novel anomaly detection model using the GAN architecture.

Despite the popularity of deep learning and anomaly detection in the literature, there is comparatively little literature on using GANs as a form of anomaly detection in NPPs. The authors of [57] developed a GAN-based model that can be used to reconstruct missing signals under emergency situations in NPPs. While this paper also uses a GAN-based approach for the purpose of anomaly detection in NPPs, the focus of the study/methodology differ from ours. Whereas that paper focused on signal reconstruction during emergency situations, our study aims to detect anomalies in the day-to-day plant operations. Additionally, that study only considered sensor data while ours takes in both automatically collected sensor data and manually collected data as input, and identifies potential mismatches in order to detect anomalies. In [58], the authors use VAEs and iForest to detect anomalous operation state in NPP. They compare VAE performance in extracting data from thermal hydraulic transient operation parameters to a traditional GAN architecture, as well as to Deep Boltzmann machine and an autoencoder. The results suggest VAE to be the more appropriate preprocessing method for detecting abnormal operation in NPPs. However, it should be noted that this study considers NPP accident conditions (e.g., loss of coolant and steam generator tube rupture), and not necessarily HE detection, which is the focus of our study.

3. Methods

3.1. Data, challenges, and preprocessing

Two datasets are used in this study. The first dataset, referred to as the NPP dataset, was obtained from an electric utility corporation in the United States. The second dataset, referred to as the testbed dataset, was obtained from a testbed at the Department of Nuclear Engineering at the University of Tennessee, Knoxville.

The NPP dataset has two facets, the automatically collected sensor data and the manually collected surveillance data. All data are collected from a main feed pump in an NPP. This dataset is considered to have rare incidents of HE, attributing to the high standards and reliability of the NPP personnel. The corresponding data types from the main feed pump are as follows:

- **Sensor data:** Continuous, minute-by-minute, datastreams were collected from the pump via mounted sensors. The datastreams captured vibrations and turbine speed. Specifically, ten vibration datastreams (in units of one-thousandth of an inch [mils]) were collected from the inboard, outboard, and thrust bearings, and one speed datastream (in units of RPM) was collected from the turbine. The datastreams were captured from 2016 through 2020.
- **Surveillance data:** Surveillance data were manually collected by field workers and/or operators via vibration monitors. These data are extremely rich and include the vibration waveform and spectrum; however, they are not fully captured in the maintenance records. Rather, the records that are available include the time of data collection, speed (RPM), plus the parameters obtained from spectral analysis. These specifically include vibration parameters across three different axes (i.e., vertical, horizontal and axial, expressed in in./second). During the time window for collecting the sensor data, i.e., 2016–2020, surveillance data were intermittently collected in 844 instances. Hence, at each of these instances, a total of 13 parameters (10 from the sensors and three from the surveillance data) are available to use for HE detection.

Human error was introduced in the testbed dataset, as shown in Table 1. Similar to the NPP dataset, two different sets of data types were collected from the testbed dataset:

- **Sensor data:** Continuous datastreams were collected from the system via mounted sensors. Permanently mounted sensors captured the temperature (in C°) and coolant flow (in milliamps) for the primary and secondary loops. These readings were captured at a frequency of .25Hz. The currents of the variable frequency drive, motor-operated valve, and the heater were also outputted by the controllers. Additionally, the permanently mounted sensors captured vibrations. Specifically, high-frequency tri-axial acceleration readings (in units of millivolts) were collected from the accelerometer. A flow loop diagram of the testbed is provided in Figure A.1 of the Appendix. The datastreams were captured intermittently from June to October 2021.
- **Surveillance data:** Surveillance data were manually collected via Fluke handheld probes used by the research team. During each reading, Fluke probes were used to collect data from two locations, and HE may be intentionally introduced when collecting measurements from location 1, location 2, or both. Table 1 summarizes the HEs introduced during the data collection.

For each surveillance reading, multiple speed datastreams (in units of 'RPM') were estimated for the pump. Additionally, as with the surveillance data in the NPP dataset, the tri-axial acceleration data (mm/sec) collected by the Fluke monitors include the vibration waveform and spectrum making them extremely rich. Thus, this data are not fully captured in the Fluke-generated reports, which simply include graphs obtained via order analysis. For each of the two Fluke probe locations, we extracted eight peaks from the relevant low-range graphs [59] (below order 1, and between orders i and $i + 1$, $i \in \{1, 7\}$) across three axes (i.e., horizontal, tangential, and radial), for a total of 24 readings parameters per location). Although the reports provide both low- and high-range graphs, the low-range graphs were specifically used, as the higher orders did not contain any valuable information.

In total, for each reading, we extracted 48 parameters for each of the 32 normal and 8 anomalous instances collected.

3.1.1. *Data challenges*

Each dataset is subject to certain limitations. The NPP dataset is unlabeled (i.e., no labels are associated with the anomalous and non-anomalous data), making model validation efforts difficult. In the testbed dataset, the Fluke-generated graphs are in PDF format. Hence, data are manually extracted from the graphs using the open-source Engauge Digitizer Software [60], restricting the granularity with which they are recorded in the dataset. The readings are recorded to two decimal points to preserve the level of detail and provide more information to the models for analysis. We address these challenges through data preprocessing, cleaning, and experimental design efforts.

3.1.2. *Data Preprocessing and Cleaning - NPP dataset*

In the NPP dataset, the two sub-datasets of sensor and surveillance data were merged based on timestamps. Before data merging, the surveillance data were preprocessed as shown in Algorithm 1.

Algorithm 1 NPP Data Preprocessing and Cleaning

```
Require: Surveillance dataset (mils) ($x$: vertical, $y$: horizontal, $z$: axial) + surveillance
         turbine speed
Require: Sensor data (in./sec) (10 vibration data) + sensor turbine speed
procedure Surveillance dataset cleaning($x$, $y$, $z$)
    for matching timestamps in $x$, $y$, $z$ do
        preprocessed surveillance ← merge($x$,$y$,$z$)
    end for
end procedure
procedure NPP dataset cleaning(preprocessed surveillance, sensor)
    for every preprocessed surveillance data do
        for every sensor data do
            for matching timestamps do
                merged ← merge(preprocessed surveillance, sensor)
            end for
        end for
    end for
end procedure
procedure steady state filtering(merged)
    for every merged data do
        if sensor turbine speed $< 4800$ RPM or sensor turbine speed $> 5100$ RPM then
           remove data
        else if surveillance turbine speed $< 4800$ RPM or surveillance turbine speed $>$
              5100 RPM then remove data
        end if
    end for
end procedure
```

Table 1
Testbed Fluke HE descriptions.

HE	Error type	Description of HE
1	Base Measurement	No HE in either location, correct position for both locations
2	Off axis	Axis rotated 90° CW in 1st location,90° CCW in 2nd location
3	Off axis (45°)	Axis rotated 45° CCW in both locations
4	Off position	Axis rotated 180° in first location,
5	First location for both measurements	Both measurements taken from first location
6	Second location for both measurements	Both measurements taken from second location
7	Off position 1st measurement	Measurement from 1st location off position (axis rotated 180°)
8	Off position 2nd measurement	Measurement from 2nd location off position (moved ‘left’ to offset position 45°)
9	Swapped locations	Locations for 1st and 2nd measurements swapped

CW: clockwise, CCW: counterclockwise.

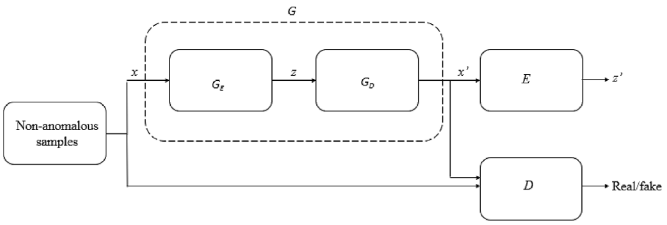

Fig. 1. GAN-based method architecture (adapted from [65]). Note that G_E and G_D denote the autoencoder network that acts as a generator, E denotes the encoder network, and D denotes the discriminator network. Further, x and z represent the non-anomalous data and the features extracted from x, respectively, and x' and z' represent the reconstructed data and the features of the reconstructed data, respectively.

Each surveillance data entry consisted of the timestamp, the speed, and vibration data in one of the three axes (i.e., vertical, horizontal, and axial). By matching the timestamps (with a maximum variation of 1 min), the corresponding vibration data in the three axes were combined into a single entry.

After preprocessing the surveillance data, they were merged with the sensor data by matching the corresponding timestamps. In this data merging, speed values from the sensor and surveillance data were both preserved for further fine-tuning. Our goal was to conduct the analysis for the period in which the system was in steady state (normal operation). As such, only the observations with speed values in the range of 4800–5100 RPM in either the sensor or surveillance data were considered. Selection of this range is guided by our industry partners and was based on the frequency distribution of the sensor-collected speed data, as shown in Fig. 4.

A high-level overview of the data curation and cleaning process for the NPP dataset is presented in Fig. 5. The clean data contain a total of 189 sensor-surveillance observations. As discussed in Section 3.1, each observation has 13 parameters (10 from the sensors and three from surveillance) used in the learning models.

3.1.3. Data Preprocessing and Cleaning - Testbed dataset

Algorithm 2 Testbed Data Preprocessing and Cleaning

```
Require: Surveillance dataset (tri-axial acceleration data (mm/sec) for each of the two
    probe locations)
Require: Sensor data (tri-axial acceleration readings (millivolts) ($x$: vertical, $y$: horizontal,
    $z$: axial)
procedure SENSOR DATASET CLEANING($x$, $y$, $z$)
    for every data entry $x$, $y$, $z$ do
        preprocessed sensor ← stationary wavelet transform on ($x$, $y$, $z$) during the minute
        of interest to extract features
    end for
end procedure
procedure TESTBED CLEANING(surveillance, preprocessed sensor)
    for every surveillance data do
        for every preprocessed sensor data do
            for matching timestamps do
                merge(surveillance, preprocessed sensor)
            end for
        end for
    end for
end procedure
```

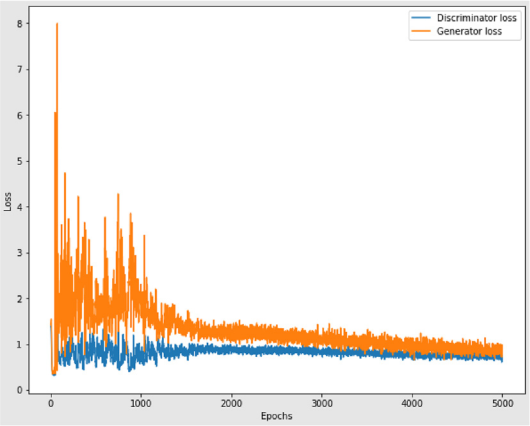

Fig. 2. Convergence of the discriminator and generator losses for GAN training optimization.

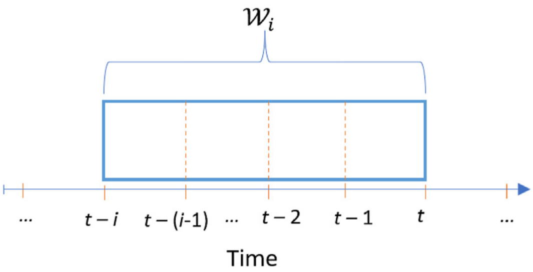

Fig. 3. Representation of temporal window $\mathcal{W}_i$, which includes all the data between time points $t-i$ and t. In this study, window lengths up to $\mathcal{W}_3$ are considered.

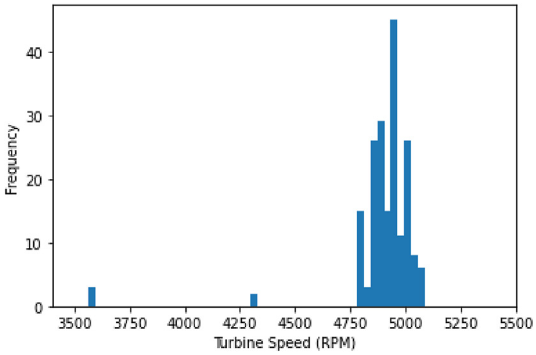

Fig. 4. Frequency distribution of speed (RPM), as recorded by mounted sensors. The 4800–5100 RPM range is considered as steady state (normal operation).

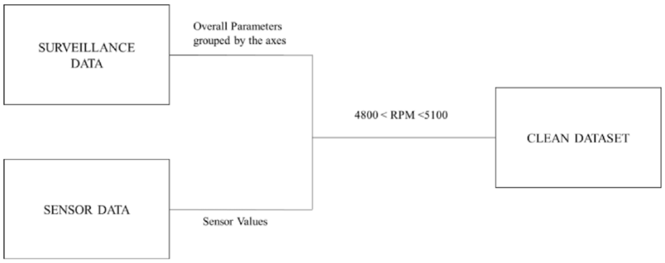

Fig. 5. High-level overview of the NPP dataset preprocessing and cleaning scheme used to curate a clean dataset of sensor and surveillance data.

A brief overview of the algorithm used for testbed data cleaning is provided in Algorithm 2. As with the NPP dataset, in the testbed dataset, the two sub-datasets of sensor and surveillance data were merged based on timestamps. Since the testbed data were collected at high frequency, prior to merging the sub-datasets, we first performed stationary wavelet transform (SWT) [61] on the sensor datastreams collected during the minute of interest in order to extract features. In particular, we opted to use SWT for feature extraction, as the datastream from the sensors is non-stationary [62,63]. Furthermore, by using the SWT, the coefficients are provided in a “non-decimated” manner in which the length of the coefficients are the same as the length of the original signal, allowing us to merge the SWT-transformed sensor data with the surveillance data, without conducting any additional preprocessing on the datasets. As in the literature [62], we used the Daubechies fourth-order (db4) wavelet and recorded the mean of the first-level approximation coefficients corresponding to the 1-min interval. While SWT provides both approximation and detail coefficients, we opted to solely use the approximation coefficients, as they represent the low frequencies and contain the signal’s key features for examining long-term trends [64]. To further enable testing of the method’s robustness to our choice of transform, we also preprocessed the sensor datastreams using fast Fourier transform (FFT), itself a well-established approach to feature extraction from signals [63]. The clean dataset contained a total of 40 sensor-surveillance observations: 32 normal and 8 anomalous.

3.2. Models and benchmarks

This section outlines the GAN-based model used for anomaly detection in this study. It also discusses the traditional anomaly detection algorithms (i.e., OCSVM and iForest) we used for initial labeling of the data, as well as benchmarking.

3.2.1. GAN-based method

GAN is an unsupervised learning technique originally proposed in [53] for the purpose of generating photorealistic images. GANs utilize two subnetworks: a generator and a discriminator. The generator works to generate new instances, and the discriminator attempts to classify them as real or fake (i.e., generated by the generator). The generator and the discriminator train adversarially, with the goal of having the generator mislead the discriminator into categorizing the generated instances as real inputs. The objective of GAN is expressed with a two-player minimax loss function, defined as:

$$\min_{G}\max_{D} V(D,G) = \mathop{\mathbb{E}}_{x\sim p_x}[\log D(x)] + \mathop{\mathbb{E}}_{z\sim p_z}[\log(1-D(G(z)))], \qquad (1)$$

where $V(D, G)$ is the cost function, G denotes the generator, D denotes the discriminator, x denotes an instance which is randomly drawn from the data p_x, and z denotes the latent variable which is randomly drawn from prior p_z. In essence, equation (1) learns model parameters such that it maximizes the likelihood of the discriminator D correctly distinguishing between real-world data and generated data. At the same time, it seeks to minimize the probability of the generator G generating new instances that are categorized correctly by the discriminator D.

The GAN-based method adapted in this study for the purpose of anomaly detection was introduced in [65] as a novel anomaly detection model. It improves upon the original GAN architecture by employing an autoencoder network that acts as a generator (G_E and G_D), the encoder network E, and the discriminator network D. Further, to cater to the requirements of our dataset, the input sizes and convolutional layer dimensions were changed accordingly. This is required since our dataset has textual data unlike the image dataset that was used in [65]. The objective function for the generator takes into consideration three loss functions: adversarial loss, contextual loss, and encoder loss. The autoencoder networks helps the model in better understanding the features of the dataset. Fig. 1 depicts the architecture of the GAN-based implementation where x denotes non-anomalous data, z denotes the features extracted from x, x' denotes the reconstructed data, and z' denotes the features of the reconstructed data.

The adversarial loss measures how successfully the generator G can distinguish between real and generated instances. Given the function $f(\cdot)$ which represents the output of the discriminator D's interior layer, the formula for adversarial loss is given as follows:

$$\mathcal{L}_{adv} = \mathop{\mathbb{E}}_{x \sim p_x} \| f(x) - \mathop{\mathbb{E}}_{x \sim p_x} f(G(x)) \|_2. \tag{2}$$

where $\| \cdot \|_2$ denotes the $\mathcal{L}_2$ norm of the interior features of the real instance, $x \sim p_x$, and the generated instance, $G(x)$.

The contextual loss optimizes the generator to learn contextual information from the input instances, i.e.,

$$\mathcal{L}_{con} = \mathop{\mathbb{E}}_{x \sim p_x} \| x - G(x) \|_1, \tag{3}$$

where $\| \cdot \|_1$ denotes the $\mathcal{L}_1$ norm.

Lastly, the encoder loss measures the degree to which the generator successfully encodes the features of the generated instance. The encoder loss calculates the difference between the encoded features of the generated instance $G_E(x)$ and the features of the real instance $E(G(x))$ [65]. The formula for encoder loss is given by:

$$\mathcal{L}_{enc} = \mathop{\mathbb{E}}_{x \sim p_x} \| G_E(x) - E(G(x)) \|_2. \tag{4}$$

Given these losses, the objective function of the GAN-based method is given by:

$$\mathcal{L} = w_{adv}\mathcal{L}_{adv} + w_{con}\mathcal{L}_{con} + w_{enc}\mathcal{L}_{enc}, \tag{5}$$

where w_{adv}, w_{con}, and w_{enc} represent the weights of the adversarial, contextual, and encoder losses in the overall objective function, respectively.

Once the GAN-based model is trained, the encoder loss $\mathcal{L}_{enc}$, defined in equation (4), is used to determine the anomaly score $\mathcal{A}(\widehat{x})$ in a test instance $\widehat{x}$. The formula for the anomaly score is given by:

$$\mathcal{A}(\widehat{x}) = \| G_E(\widehat{x}) - E(G(\widehat{x})) \|_1, \tag{6}$$

where $\| \cdot \|_1$ denotes the $\mathcal{L}_1$ norm. The anomaly score is computed for the instances in the test set $\widehat{\mathcal{D}}$, which gives a set of anomaly scores $S = \{s_i : \mathcal{A}(\widehat{x}_i),\ \widehat{x}_i \in \widehat{\mathcal{D}}\}$. Finally, the anomaly scores are scaled from [0, 1] as follows:

$$s_i' = \frac{s_i - \min(S)}{\max(S) - \min(S)}. \tag{7}$$

Anomalies are determined based on an optimal threshold. If the output exceeds the pre-determined threshold, the instance is considered to be anomalous; otherwise, it is considered to be normal.

A 1-D GAN architecture was used in this study, with the sensor and the surveillance data being fed as a 1-D array. As is consistent with literature [50,66,67], the feature layer was used. The architecture used for the GAN-based method in this study is presented in Table A.1 of the Appendix. To ensure that the model was sufficiently trained and optimized, the losses of the discriminator and generator were plotted. The convergence of the respective losses was considered as a measure for the algorithm optimization. Fig. 2 depicts the convergence of the discriminator and generator losses. For our preliminary analysis, a 2-D GAN architecture was also implemented, with the sensor and the surveillance data being fed as a 2-D “image.” However, because the preliminary results did not show model improvement, we ultimately opted for a 1-D GAN.

3.2.2. OCSVM

OCSVM is an extension of the popular supervised learning method, SVM, and establishes a hyperplane to separate a dataset into two or more subsets. OCSVM computes a non-linear decision distribution or boundary around the training data, using a given kernel. It consequently categorizes them as “suspicious” if the training data fall outside this boundary with respect to the kernel chosen. The dataset is mapped to a high-dimensional feature space. The origin is labeled as -1, and the training instances are labeled as $+1$ [68]. The goal in OCSVM is to build a hyperplane that features the maximum distance between the training instances and the origin [69].

In OCSVM, the optimal hyperplane is found by solving the following optimization problem:

$$\min_{w,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{n \cdot v} \sum\nolimits_{i=1}^{n} \max(0, \rho - \langle w, \phi(y) \rangle) - \rho, \tag{8}$$

where n represents the number of data points, $v \in (0, 1)$ is a parameter that controls the tradeoff between maximizing the distance to the origin and false positives, ρ and w are the hyperplane parameters, and ϕ is a mapping function [70]. Consequently, the hyperplane is constructed as follows:

$$f(x) = w \cdot \phi(x) - \rho. \tag{9}$$

As is consistent with the literature [71], [72], we used the Gaussian kernel in this study, since it is a widely adopted kernel for SVM with regard to high-dimensional data.

3.2.3. iForest

iForest is an unsupervised ML algorithm based on decision trees. Specifically, iForests detect anomalies by using binary trees [73]. iForests work on the basis of two main assumptions: (1) anomalous instances tend to occur at a lower frequency than do normal instances, and (2) the parameter values of these anomalous instances differ from those of normal instances. These assumptions mean that anomalies are more susceptible to becoming isolated from the rest of dataset. The goal of iForest is to create a collection of isolation trees (iTrees), a tree structure in which the anomalous instances are grouped closer to the root and the normal instances form the leaves of the tree [73].

Partition selection in the algorithm is random. That is, iForest chooses a random feature and then selects a value between the maximum and the minimum values of that feature [73]. Partitioning occurs until all instances have been isolated. Anomalies are then determined based on the path length. Given the above assumptions about anomalies, they require fewer partitions to be isolated from the rest of the dataset (i.e., they have a shorter path length to isolation).

Given n instances in the dataset where $n > 2$, the path length is given by

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}, \tag{10}$$

where $H(n)$ is approximated by the harmonic number $\ln(n) + \gamma$

(Euler's constant). Consequently, the anomaly score of an instance x is calculated using the following formula:

$$s(x,n) = 2^{-\frac{\mathbb{E}(h(x))}{c(n)}}, \tag{11}$$

where $h(x)$ represents the path length or the number of iterations to isolate instance x. If the sample x returns an s that is higher than the pre-defined threshold (typically ≥ 0.5), x is marked as anomalous; otherwise it is considered non-anomalous [73].

3.3. Experimental design

As discussed in Section 3.1.1, the NPP dataset is unlabeled. Although this does not prohibit us from using unsupervised learning algorithms, it makes model validation efforts difficult. As such, in this section we design experiments to enable model validation. More specifically, in these experiments, we first use a given learning algorithm for labeling the observations in the clean dataset as being either normal or anomalous. Then we test the performance of the GAN-based method in detecting the anomalous instances against state-of-the-art benchmarks. For the benchmarks, we perform grid search and hyperparameter tuning to ensure training the best models possible. To further validate the models, we apply the models to testbed data, with known labels.

To evaluate model performance, we randomly split the data into training/validation/testing sets. We use the training/validation sets for fine-tuning the models and the test sets for objective model evaluation. We repeat the process 10 times to report the mean performance for the test set along with the corresponding confidence intervals. The experiments are repeated for various potential anomaly rates ξ, $\xi \in \Xi = \{1\%, 2.5\%, 5\%, 10\%\}$ in the NPP dataset, for which the true anomaly rate is unknown. Note that we do not consider anomaly rates above 10%, as NPP personnel are generally highly trained and the HE rate is not expected to exceed 10% [12]. The experiments are repeated for various anomaly rates ξ, $\xi \in \Xi = \{10\%, 20\%\}$ in the testbed dataset. The higher anomaly rates of 10 and 20% are especially considered, as smaller rates would require too many normal instances. The different anomaly rates mentioned above were incorporated during the training, validation, and testing splits of the normal and anomalous datasets. Table A.2 and A.3 show these splits for the NPP dataset and testbed dataset, respectively.

3.3.1. Experiment NPP-I: OCSVM labeled anomaly detection

In this experiment, we use OCSVM for the initial labeling of the dataset. The labeled data are then split to create training/validation and testing sets. The size of the training/validation set is determined based on the testing set. To get equal proportions of anomalous and non-anomalous datastreams in the testing set, the training and testing split for the datasets must be carefully considered. The iForest and GAN-based methods are trained on the training/validation data and then tested on the testing data to compare their performance.

In the iForest anomaly detection, the model is trained on the training set that includes both normal and anomalous data. The proportion of anomalies in the training dataset (i.e., the contamination rate) is given to the algorithm as an input. This allows iForest to create branches that can isolate a predetermined proportion of the data as anomalies.

In contrast, in the GAN-based method, the GAN architecture is trained only on the non-anomalous data. This allows the GAN to learn the behavior of normal data without the need for any *a priori* known contamination rate. Once this behavior is learned, the validation set that includes both normal and anomalous data is used to optimize the threshold for anomaly scores, as discussed in Section 3.2.1. Figs. 6 and 7 provide an overview of the pipelines used for implementing OCSVM Labeled iForest and GAN-based anomaly detection methods, respectively.

3.3.2. Experiment NPP-II: iForest labeled anomaly detection

Unlike Experiment NPP-I, this experiment employs iForest for the initial labeling of the dataset. Once the data are labeled, they are split to create training/validation and testing sets, just as in Experiment I. The OCSVM and GAN-based methods are trained on the training/validation data and then tested on the testing data to compare their performance. In the OCSVM anomaly detection, the model is trained on the training set that includes both normal and anomalous data, with the contamination rate fed to the algorithm as an input. In the GAN-based method, the GAN architecture is trained only on the non-anomalous data, and the validation set is then used to optimize the threshold for anomaly scores. Figs. 8 and 9 provide an overview of the pipelines used for implementing iForest Labeled OCSVM and GAN-based anomaly detection methods, respectively.

3.3.3. Experiment Testbed-III: Anomaly detection in the testbed data

In this experiment, we implement anomaly detection algorithms on the testbed data. Recall that, as mentioned in Section 3.1, the labels are already known in the corresponding dataset. For this experiment, the dataset is divided into normal and anomalous data, based on the location of where the anomaly was introduced, as summarized in Table 1. We consider two cases: one in which the dataset is considered holistically (with HEs potentially being present throughout the entire data collection process), and one that is location-specific (with HEs being present at the specific locations where the data are collected). As with the NPP experiments, the data are then split to create training/validation and testing sets, given the anomaly rate considered. The iForest, OCSVM, and the GAN-based methods are subsequently trained on the training/validation sets and then tested on the test set.

3.4. Evaluation metrics

The metrics that are used to evaluate the performance of the anomaly detection approaches are accuracy, sensitivity, specificity, and geometric mean (G-Mean). Each of these metrics is based on true positives (*TP*), true negatives (*TN*), false positives (*FP*) and false negatives (*FN*). The detailed formulas and descriptions are as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Specificity} = \frac{TN}{TN+FP}\text{G} - \text{Mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}}$$

Accuracy is the proportion of correctly identified normal and anomalous instances. Due to unbalanced data (i.e., over-representation of normal data vs. anomalous data), accuracy does not holistically capture the performance of anomaly detection algorithms [74]. However, because it is a commonly reported metric in the literature, we chose to include it among our model evaluation metrics.

Sensitivity, or the true positive rate, is the proportion of correctly identified anomalous instances. Sensitivity provides information on how well a given model performs in detecting anomalous instances; hence, a model with high sensitivity would be ideal.

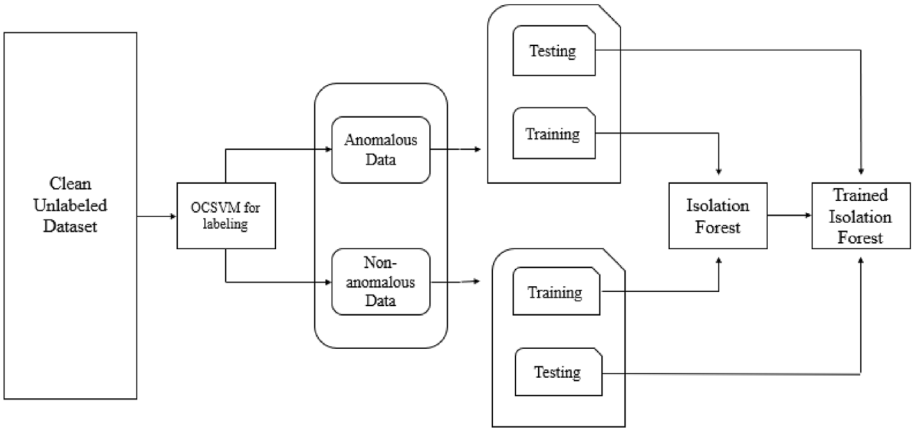

Fig. 6. OCSVM labeled iForest anomaly detection pipeline.

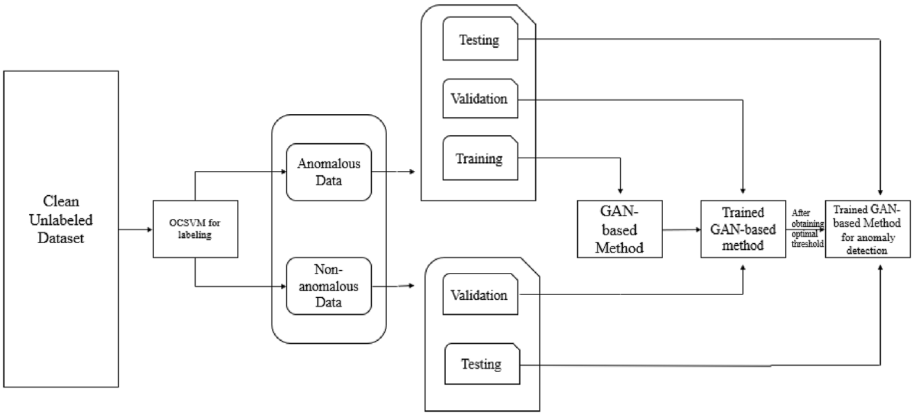

Fig. 7. OCSVM labeled GAN-based anomaly detection pipeline.

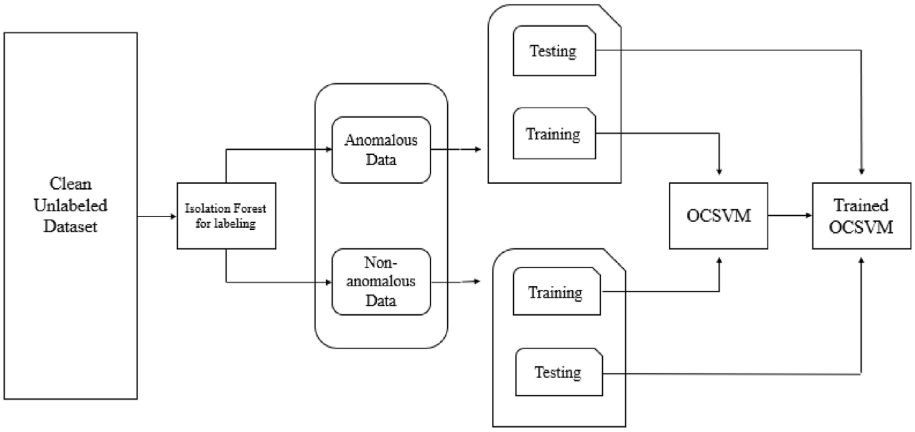

Fig. 8. iForest labeled OCSVM anomaly detection pipeline.

Specificity, or the true negative rate, is the proportion of correctly identified normal instances. This metric provides information on how well a given model performs in detecting normal instances; hence, a model with a high specificity would be ideal.

G-Mean is the product of sensitivity and specificity. Thus, G-Mean takes into consideration both the true positive and true negative rates. This makes it a holistic evaluation metric for anomaly detection algorithms. Hence, we use G-Mean as our primary evaluation metric in this study.

4. Results

In this section, we present the anomaly detection performance in regard to the NPP and testbed datasets. We first present the descriptive statistics of the two datasets (Section 4.1). Next, we examine the differences between OCSVM and iForest labeling in the NPP dataset (Section 4.2). We then reveal the anomaly detection performance for the NPP (Section 4.3) and testbed datasets (Section 4.4).

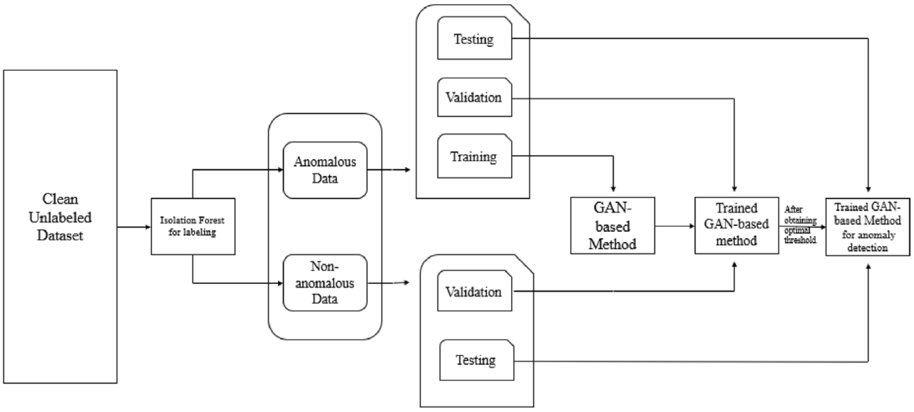

Fig. 9. iForest Labeled GAN-based anomaly detection pipeline.

4.1. Descriptive statistics

Tables 2 and 3 present the descriptive statistics of the NPP sensor and surveillance data, respectively. As seen in the two tables, although the sensor and surveillance data both measure vibration, they do so in fundamentally different ways. Sensors are mounted inside the system and measure internal vibrations, whereas surveillance data are collected through vibration monitors that measure data at a much higher frequency and provide spectral analysis results for the data. Hence, although sensor and surveillance data generally correlate with each other, they do not exactly match.

Tables 4 and 5 present the descriptive statistics of the testbed sensor and surveillance data, respectively. Note that in the former, the statistics were compiled after implementing SWT on the raw sensor values.

4.2. Anomalous vs. normal labeling in NPP dataset – OCSVM vs. iForest

As discussed in Section 3.3, the unlabeled dataset was initially labeled using OCSVM and iForest in the two experiments. In this section, we investigate the differences in the corresponding anomalous-/normal-labeled data resulting from these two algorithms.

Figs. 10 and 11 present the boxplots of the normal and anomalous surveillance data under anomaly rates $\xi \in \Xi = \{1\%, 2.5\%, 5\%, 10\%\}$, as labeled by OCSVM and iForest, respectively. As seen in Fig. 10, the median, fourth spread, and the range of normal data are somewhat consistent under the anomaly rates $\xi \in \Xi = \{1\%, 2.5\%, 5\%, 10\%\}$. However, the characteristics of the anomalous data are generally more sensitive to the choice of the anomaly rate and present more variability in terms of the anomaly rate. These observations are consistent with those made in Fig. 11, in which the characteristics of the normal data are somewhat consistent under the anomaly rates considered, and the anomalous data are generally more sensitive with respect to this choice.

Interestingly, even though both OCSVM and iForest are state-of-the-art approaches, as seen in Figs. 10 and 11, they do not provide consistent anomalous data under the anomaly rates $\xi \in \Xi = \{1\%, 2.5\%, 5\%, 10\%\}$. This is evident from comparing the characteristics of the anomalous data under the different anomaly rates considered across the two figures. As discussed in Section 3, OCSVM and iForest use very different approaches to detect anomalies in the high-dimensional data and hence, differences between the resulting anomalous- and normal-labeled data are expected. Also as expected, sensor data generally present behavior similar to that of surveillance data. That is, the characteristics of normal data are somewhat consistent under the anomaly rates considered, whereas

Table 2
Descriptive statistics of the automatically collected NPP sensor data (mils).

	Count	Mean	SD	Min	1^{st} Quartile	Median	3^{rd} Quartile	Max
Sensor 1	189	2.583	2.875	0.0	0	1.6	1.7	7.5
Sensor 2	189	2.671	2.846	0.0	0	1.8	1.9	7.5
Sensor 3	189	2.555	2.813	0.1	1	1	1	7.5
Sensor 4	189	2.376	2.954	0.0	0	1.2	1.2	7.5
Sensor 5	189	2.543	2.82	0.2	0.9	1	1.1	7.5
Sensor 6	189	2.195	3.034	0	0	0.8	0.9	7.5
Sensor 7	189	4.11	1.946	0.2	2.9	3.2	3.4	7.5
Sensor 8	189	4.086	1.954	0.1	3	3	3.1	7.5
Sensor 9	189	5.502	3.052	0.2	5.8	7.1	7.4	8.1
Sensor 10	189	5.748	3.199	0.2	6	7.4	7.8	8.6

Table 3
Descriptive statistics of the manually collected NPP surveillance data (in./sec).

	Count	Mean	SD	Min	1^{st} Quartile	Median	3^{rd} Quartile	Max
Overall - Vertical	189	0.141	0.083	0.031	0.087	0.114	0.174	0.432
Overall - Horizontal	189	0.190	0.111	0.053	0.116	0.145	0.233	0.516
Overall - Axial	189	0.139	0.05	0.041	1.095	0.133	0.175	0.288

Table 4
Descriptive statistics of the testbed sensor data (SWT).

	Count	Mean	SD	Min	1^{st} Quartile	Median	3^{rd} Quartile	Max
Overall - Acceleration *x*	40	0.022	0.021	0.002	0.007	0.013	0.026	0.079
Overall - Acceleration *y*	40	0.015	0.016	-0.006	0.005	0.008	0.020	0.064
Overall - Acceleration *z*	40	0.031	0.028	0.004	0.011	0.021	0.035	0.102

Table 5
Descriptive statistics of the manually collected testbed surveillance data (mm/sec).

	Count	Mean	SD	Min	1^{st} Quartile	Median	3^{rd} Quartile	Max
Overall - Axial	40	0.812	0.858	0.000	0.055	0.200	0.855	19.911
Overall - Tangential	40	1.274	1.998	0.000	0.059	0.243	1.051	13.050
Overall - Radial	40	0.918	1.784	0.000	0.070	0.251	1.053	9.000

those of the anomalous data change in accordance with the anomaly rate. In addition, again, anomalous data do not seem to follow the same characteristics when labeled by OCSVM vs. iForest. For the detailed boxplots of the normal and anomalous sensor data under anomaly rates $\xi \in \Xi = \{1\%, 2.5\%, 5\%, 10\%\}$, as labeled by OCSVM and iForest, see Figures A.2 and A.3 in the Appendix, respectively.

4.3. Anomaly detection performance in the NPP dataset

Here, we present the results of Experiments NPP-I and NPP-II described in Section 3.3. We first present the models in which only the data collected within a span of a minute are used for anomaly detection. We next examine whether including the temporal history of sensor data can improve model performance. Through these experiments, we examine whether or not the GAN-based approach can consistently retrieve anomalies, regardless of how the initial labeling is done.

4.3.1. Baseline models

Tables 6 and 7 present the mean and standard deviation of the performance metrics for Experiment I: OCSVM Labeled Anomaly Detection and Experiment II: iForest Labeled Anomaly Detection, respectively. The training, validation, and testing splits are presented in Table A.2 of the Appendix. Paired *t*-tests are performed to compare the results between iForest/OCSVM and GAN-based approaches. The results are bolded if they are statistically better at the 0.05 significance level (α).

As seen in Table 8, the GAN-based approaches outperform iForest regardless of the anomaly rate with respect to the primary metric of G-Mean. Additionally, iForest is generally more accurate and sensitive; however, this comes at the cost of reduced specificity, which ultimately results in a lower G-Mean. Overall, the GAN-based approach is more successful than iForest at retrieving the data correctly when the data are labeled by OCSVM. As seen in Table 9, the GAN-based approach outperforms OCSVM under the $\xi = 1$, 2.5, and 5% anomaly rates and presents comparable performance to OCSVM under $\xi = 10\%$ anomaly rate with respect to the primary metric of G-Mean. Hence, in summary, in both Experiments NPP-I and NPP-II, the GAN-based method proved superior with respect to specificity and G-Mean, except in the case of the $\xi = 10\%$ anomaly rate in Experiment NPP-II, for which the difference in the results between OCSVM and the GAN-based method are not statistically significant.

4.3.2. Inclusion of temporal history

In this section, we examine whether including temporal sensor data can help model performance. That is, we include a temporal window $\mathcal{W}$ of sensor data when merging sensor data with surveillance data at time *t*. We specifically use three time windows, which we denote by $\mathcal{W}_i$, corresponding to the temporal sensor data in the interval $(t - i, t)$, i.e., $\mathcal{W}_i \in \{(t - i, t) : i = 1, \ldots, 3\}$ (See Fig. 3). As such, the surveillance data at time *t* is merged with the sensor data in window $\mathcal{W}_i$ associated with time *t*. Contrast this with the baseline model in Section 4.3.1, in which only sensor data from time *t* are included. The data and instances used in this section are otherwise consistent with those in Section 4.3.1.

Tables 8 and 9 present the results under window $\mathcal{W}_1$ for Experiments NPP-I and NPP-II, respectively. As seen in the tables, compared with the results obtained under the baseline model in Section 4.3.1, the results of the OCSVM labeled models remain statistically similar. At 1% anomaly, the iForest-labeled GAN slightly improves with regards to accuracy, G-Mean, and sensitivity (paired *t*-test *p*-value < 0.05). Additionally, the specificity improves for iForest-labeled GAN at $\xi = 10\%$ anomaly rate.

However, increasing the window length to include more temporal sensor data seemingly leads to diminishing returns or even decreased model performance. That is, in some instances, the results under window $\mathcal{W}_1$ are better than those under windows $\mathcal{W}_2$ and $\mathcal{W}_3$ (Tables A.5 - A.8 from the Appendix for the detailed results). Hence, only window lengths up to $\mathcal{W}_3$ are considered in this analysis. Regardless, across all models and windows, our GAN-based method outperforms iForest and OCSVM with respect to G-Mean and specificity at the significance level of 0.05 (paired *t*-test *p*-value < 0.05). The exception is seen in Table A.6, with iForest-labeled anomaly detection under window $\mathcal{W}_2$, where OCSVM and GAN-based method perform comparably across all evaluation metrics at the $\xi = 10\%$ anomaly rate.

4.4. Anomaly detection performance in the testbed dataset

Here, we present the results of Experiment Testbed-III as described in Section 3.3. Recall that, as discussed in Section 3.1, surveillance data from Fluke probes are collected from two locations, with HEs potentially being intentionally introduced when collecting measurements from location 1, location 2, or both. We

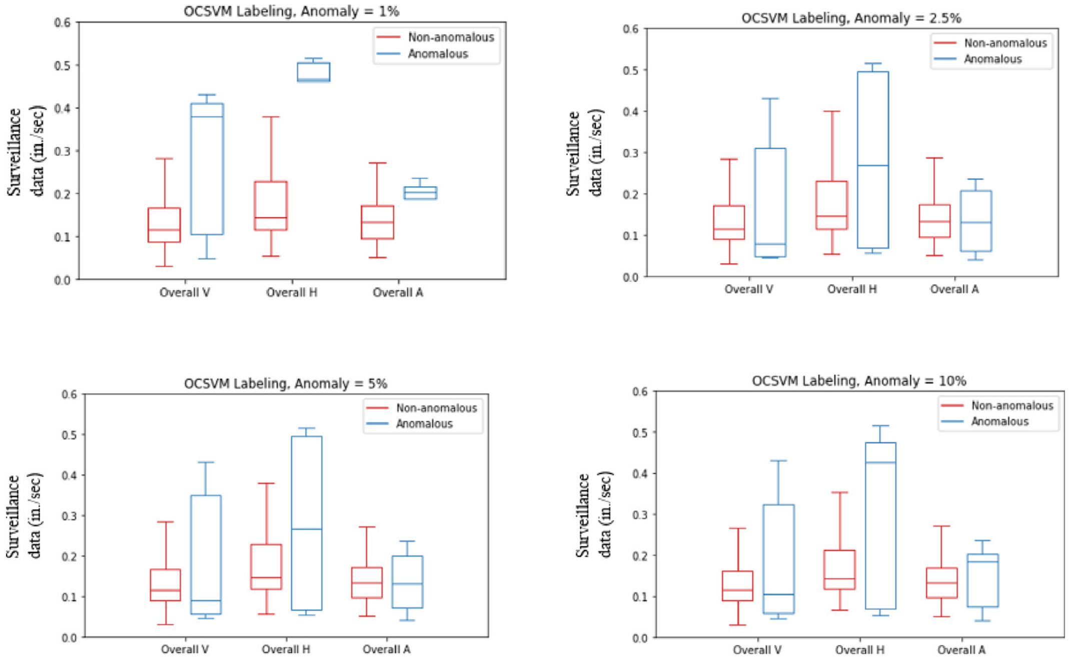

Fig. 10. Boxplots of normal and anomalous surveillance data (in./sec) under anomaly rates $\xi \in \{1\%, 2.5\%, 5\%, 10\%\}$, as labeled by OCSVM.

first present the results for anomalous instances that can be present in either location. We next examine the performance under location-specific anomaly detection. The former allows us to pull the instances together to create a larger dataset; however, the task of anomaly detection may be more difficult in this case as the profiles of the HEs may be slightly different across the two locations. The latter task may be a bit easier for the algorithms; however, the dataset is smaller. Indeed, the dataset in the latter case is too small to execute models for location 2 data in a meaningful manner; hence, this exercise is only performed with location 1 data. The training, validation, and testing splits of the testbed are presented in Tables A.3 and A.4 in the Appendix.

Table 10 presents the results when data from either location are used. At $\xi = 10\%$ anomaly rate, the GAN-based method outperforms iForest in terms of both specificity and G-Mean. At $\xi = 20\%$ anomaly rate, the GAN-based method outperforms OCSVM with regards to accuracy and G-Mean. Hence, as is consistent with the model performance observed for NPP data in Section 4.3, the GAN-based method outperforms other state-of-the-art models. Interestingly, however, overall the performance of all methods are generally higher when applied to the testbed data. This may be partly attributable to the fact that in this dataset, the labels are known *a priori*.

Additionally, we present the results of location-specific anomaly detection in the testbed dataset. Table 11 presents the results when data are collected from location 1 only. As seen in the table, at $\xi = 10\%$ anomaly rate, the GAN-based method outperforms OCSVM in terms of G-Mean, and outperforms both OCSVM and iForest in terms of specificity. All three models perform comparably under the accuracy and sensitivity metrics. Additionally, at $\xi = 20\%$ anomaly rate, the GAN-based method outperforms both models in terms of the G-Mean and specificity metrics. These results are largely consistent with the ones presented in Section 4.3.1, where the GAN-based method outperforms the other methods in terms of G-Mean and specificity.

Finally, to further test the robustness of our analysis, we regenerated the model performance results in this section when we use FFT instead of SWT to process the sensor data. As expected, the results remain largely consistent, with the GAN-based method outperforming other state-of-the-art methods. The detailed results are found in Table A.9 and A.10 of the Appendix.

5. Discussion

In this study, we established a GAN-based approach to detect HE in a manual data collection process in NPPs. Given the assumption that the automatically collected data were non-anomalous (collected from well-calibrated sensors, etc.) and that they were correlated with manually collected surveillance data, any anomalies observed in the data were hence assumed to be attributed to HE. As such, the GAN-based approach was developed to detect HE. We used both real-world NPP data and a testbed that contained both manually collected surveillance data and automatically collected sensor data. The performance of the GAN-based model was benchmarked against iForest and OCSVM. We labeled the observations in the unlabeled NPP dataset by using iForest and OCSVM. As shown by the box plots and explained in Section 4.2, even though both OCSVM and iForest are state-of-the-art unsupervised ML algorithms, there are noticeable discrepancies in how they label the data as either anomalous and non-anomalous across the different anomaly rates. In particular, the differences become more noticeable for the anomalous data. Due to the lack of consensus between the two models, and in the absence of a ground truth (i.e., the lack of existing labels in the NPP dataset), it is at first difficult to ascertain the extent to which our labeling effort produces meaningful results. Clearly, although labeled data are not required for unsupervised learning methods, they could aid in evaluating the performance of the models, which we address by introducing the labeled testbed data.

Despite the lack of labels in the NPP dataset, we devised experiments to evaluate the performance of the models through pairwise comparison of the GAN-based method with OCSVM and iForest. Specifically, we examined whether the GAN-based method does a reasonably good job of retrieving the designated labels, regardless of how the initial labeling was done. This is based on the hypothesis that the GAN-based method can learn the distribution

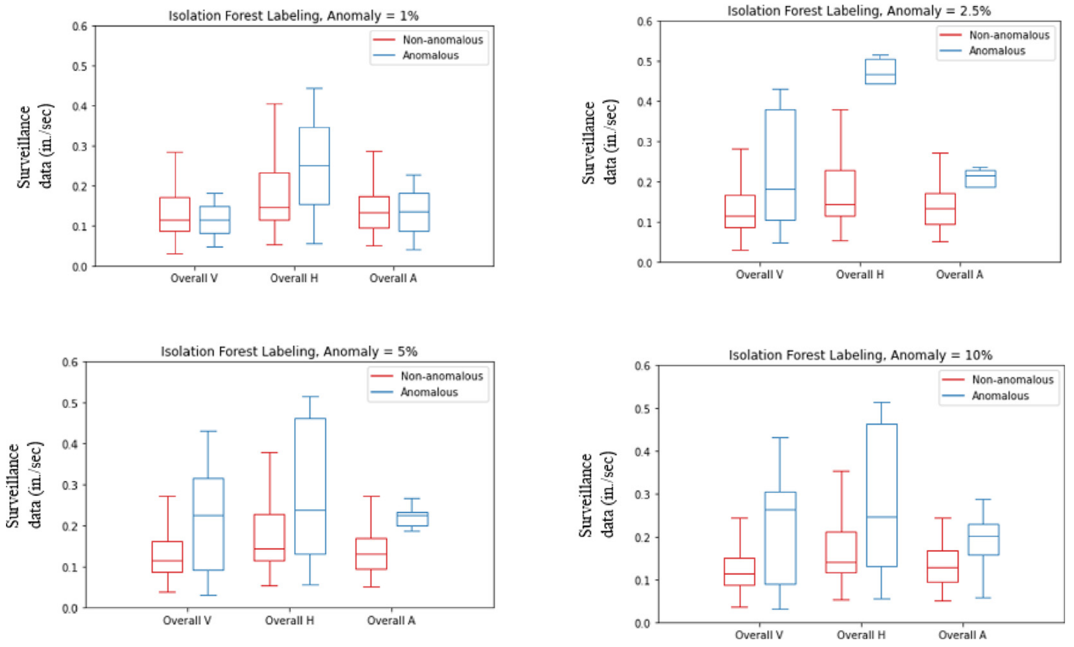

Fig. 11. Boxplots of normal and anomalous surveillance data (in./sec) under anomaly rates $\xi \in \{1\%, 2.5\%, 5\%, 10\%\}$, as labeled by iForest

of any normal data and consequently mark as anomalous any instance that does not match that distribution. This is extremely powerful, as, if the GAN-based method does reasonably well at retrieving the designated labels in the two experiments, it could potentially be used as the basis for a HE detection tool (when labels are known) to efficiently learn the distribution of the normal data. We further validate this using the testbed data in which the labels are known *a priori*.

The results presented in Section 4.3 show the performance of the GAN-based method, compared with OCSVM and iForest, for the NPP dataset. As is consistent with our hypothesis, the results indicate that the GAN-based method can indeed retrieve the designated labels reasonably well. More specifically, as seen in Tables 8 and 9, it outperforms iForest across all anomaly rates, and outperforms OCSVM under all but one anomaly rate, with respect to our primary evaluation metric, G-Mean.

We next validated our findings using the labeled testbed data in Section 4.4. The results show that, as is consistent with the results for the NPP data, the GAN-based method generally outperforms iForest and OCSVM in terms of the G-Mean for the testbed data. Interestingly, this is regardless of how the anomalous and normal data are defined (i.e., whether they come from readings from two

Table 6
Mean and standard deviation of the performance metrics for NPP-I: OCSVM Labeled Anomaly Detection (Results bolded if statistically better at $\alpha = 0.05$).

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
1%	iForest	**0.969** ± **0.011**	**0.993** ± **0.015**	0.300 ± 0.483	0.296 ± 0.477
	GAN-based	0.697 ± 0.100	0.686 ± 0.104	**1.000** ± **0.000**	**0.826** ± **0.063**
2.5%	iForest	**0.943** ± **0.048**	**0.970** ± **0.046**	0.200 ± 0.422	0.198 ± 0.418
	GAN-based	0.664 ± 0.143	0.652 ± 0.148	**1.000** ± **0.000**	**0.800** ± **0.096**
5%	iForest	**0.939** ± **0.041**	**0.959** ± **0.041**	0.400 ± 0.516	0.391 ± 0.504
	GAN-based	0.786 ± 0.303	0.685 ± 0.139	**1.000** ± **0.000**	**0.824** ± **0.081**
10%	iForest	**0.886** ± **0.041**	**0.972** ± **0.033**	0.167 ± 0.176	0.286 ± 0.302
	GAN-based	0.804 ± 0.097	0.824 ± 0.104	**0.633** ± **0.189**	**0.713** ± **0.129**

Table 7
Mean and standard deviation of the performance metrics for NPP-II: iForest Labeled Anomaly Detection (Results bolded if statistically better at $\alpha = 0.05$).

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
1%	OCSVM	**0.886** ± **0.085**	**0.918** ± **0.088**	0.000 ± 0.000	0.000 ± 0.000
	GAN-based	0.503 ± 0.140	0.486 ± 0.145	**1.000** ± **0.000**	**0.700** ± **0.105**
2.5%	OCSVM	**0.948** ± **0.041**	**0.968** ± **0.043**	0.400 ± 0.516	0.391 ± 0.505
	GAN-based	0.638 ± 0.225	0.625 ± 0.233	**1.000** ± **0.000**	**0.777** ± **0.151**
5%	OCSVM	**0.907** ± **0.054**	**0.933** ± **0.051**	0.200 ± 0.422	0.194 ± 0.410
	GAN-based	0.607 ± 0.157	0.593 ± 0.163	**1.000** ± **0.000**	**0.760** ± **0.126**
10%	OCSVM	0.875 ± 0.064	0.936 ± 0.071	0.367 ± 0.246	0.513 ± 0.292
	GAN-based	0.725 ± 0.195	0.748 ± 0.224	0.533 ± 0.172	0.611 ± 0.128

Table 8
Mean and standard deviation of the performance metrics for NPP-I: OCSVM Labeled Anomaly Detection under window $\mathcal{W}_1$ (Results bolded if statistically better at $\alpha = 0.05$).

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
1%	iForest	**0.966 ± 0.000**	**1.000 ± 0.000**	0.000 ± 0.000	0.000 ± 0.000
	GAN-based	0.793 ± 0.149	0.786 ± 0.154	**1.000 ± 0.000**	**0.883 ± 0.086**
2.5%	iForest	**0.954 ± 0.029**	**0.979 ± 0.026**	0.200 ± 0.422	0.200 ± 0.422
	GAN-based	0.793 ± 0.149	0.786 ± 0.154	**1.000 ± 0.000**	**0.877 ± 0.122**
5%	iForest	**0.950 ± 0.042**	**0.970 ± 0.034**	0.500 ± 0.527	0.492 ± 0.519
	GAN-based	0.711 ± 0.118	0.730 ± 0.157	**0.978 ± 0.070**	**0.838 ± 0.077**
10%	iForest	**0.900** ± 0.040	**0.972 ± 0.019**	0.300 ± 0.332	0.407 ± 0.373
	GAN-based	0.718 ± 0.174	0.704 ± 0.210	**0.767 ± 0.225**	**0.706 ± 0.101**

Table 9
Mean and standard deviation of the performance metrics for NPP-II: iForest Labeled Anomaly Detection under window $\mathcal{W}_1$ (Results bolded if statistically better at $\alpha = 0.05$).

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
1%	OCSVM	**0.897 ± 0.104**	**0.911 ± 0.098**	0.600 ± 0.516	0.487 ± 0.514
	GAN-based	0.709 ± 0.164	0.700 ± 0.169	**1.000 ± 0.000**	**0.831 ± 0.103**
2.5%	OCSVM	**0.942 ± 0.040**	**0.954 ± 0.038**	0.600 ± 0.516	0.585 ± 0.504
	GAN-based	0.752 ± 0.142	0.742 ± 0.147	**1.000 ± 0.000**	**0.858 ± 0.086**
5%	OCSVM	**0.902 ± 0.065**	**0.930 ± 0.056**	0.300 ± 0.483	0.290 ± 0.470
	GAN-based	0.768 ± 0.160	0.760 ± 0.166	**0.979 ± 0.07**	**0.866 ± 0.101**
10%	OCSVM	**0.854 ± 0.059**	**0.916 ± 0.076**	0.400 ± 0.306	0.449 ± 0.325
	GAN-based	0.625 ± 0.163	0.620 ± 0.208	**0.666 ± 0.223**	**0.611 ± 0.044**

Table 10
Mean and standard deviation of the performance metrics for Testbed-III when using all data from either location (Results bolded if statistically better at $\alpha = 0.05$).

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
10%	iForest	0.833 ± 0.136	0.940 ± 0.097	0.300 ± 0.483	0.300 ± 0.483
	GAN-based	0.867 ± 0.172	0.840 ± 0.207	**1.000 ± 0.000**‡	**0.909 ± 0.123**‡
	OCSVM	0.767 ± 0.211	0.720 ± 0.253	1.000 ± 0.000	0.833 ± 0.168
20%	iForest	0.833 ± 0.157	0.860 ± 0.135	0.700 ± 0.483	0.658 ± 0.456
	GAN-based	**0.917 ± 0.162***	0.900 ± 0.194	1.000 ± 0.000	**0.942 ± 0.117***
	OCSVM	0.733 ± 0.225	0.720 ± 0.253	0.800 ± 0.422	0.667 ± 0.388

* and ‡ denote significance with respect to OCSVM and iForest, respectively.

different locations, or from a single specific location).

Overall, the findings of this study are encouraging and lay the ground work for providing a novel way to detect HE in manually collected surveillance data in NPPs. The results of this study show promise for a HITL anomaly detection system in NPPs, which leverages the expertise of humans with the machine intelligence of AI.

6. Conclusion and future work

HEs contribute to a considerable percentage of incidents in NPPs, and HE detection is as important as HE mitigation efforts. While NPPs are evolving to be highly automated and AI systems are likely to alleviate part of the burden placed on human operators, there remains little doubt that humans will continue to play an integral part in NPP operations. The current work developed an anomaly detection approach to uncovering anomalous surveillance data. Such an approach has the potential to foster HE mitigation by immediately alerting human operators of potential errors.

This study developed a GAN-based anomaly detection approach for uncovering anomalies in NPP sensor and surveillance data. The approach was benchmarked against two state-of-the-art anomaly detection techniques: OCSVM and iForest. Model evaluation was

Table 11
Mean and standard deviation of the performance metrics for Testbed-III when using the data from location 1 only (Results bolded if statistically better at $\alpha = 0.05$).

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
10%	iForest	0.867 ± 0.131	0.912 ± 0.140	0.600 ± 0.516	0.567 ± 0.493
	GAN-based	0.850 ± 0.146	0.911 ± 0.302	**1.000 ± 0.000***‡	**0.901 ± 0.098***
	OCSVM	0.750 ± 0.142	0.880 ± 0.169	0.100 ± 0.316	0.089 ± 0.283
20%	iForest	0.883 ± 0.157	0.880 ± 0.169	0.600 ± 0.516	0.556 ± 0.484
	GAN-based	0.850 ± 0.123	0.820 ± 0.148	**1.000 ± 0.000***‡	**0.902 ± 0.083***‡
	OCSVM	0.800 ± 0.189	0.840 ± 0.207	0.600 ± 0.516	0.542 ± 0.479

* and ‡ denote significance with respect to OCSVM and iForest, respectively.

made complicated by the fact that the surveillance data were not properly labeled to record true anomalies. To overcome this, OCSVM and iForest were used to detect and initially label the data under different percentages of suspected anomalies. The GAN-based method and the benchmarks were then implemented and consequently evaluated to examine the extent to which they were able to retrieve these labeled data. The results were validated on a testbed dataset with known labels. The results of both the NPP and testbed dataset analyses show that the GAN-based approach generally outperformed the state-of-the-art benchmarks in detecting anomalous data that may be attributed to HE. Hence, GAN-based approach shows promise for further development as part a HE detection AI tool.

This work is subject to some limitations. As discussed in the study, labels as well as true anomaly rates are not known for the real-world data. Hence, future work includes collecting more detailed data that include such information and then using them for model development and validation. Furthermore, our validation using the testbed data was somewhat limited due to the size of our dataset, particularly the number of our normal instances. Given that the HE rate in an NPP is not expected to exceed 10%, ideally the models can be tested on lower anomaly rates than those used in our study. However, this requires collecting a large sample of normal data (e.g., 99 normal samples for every anomalous sample). This is exacerbated by the fact that separate sets must be used for training, validation, and testing to avoid overfitting. Hence, for this proof-of-concept study, we chose to use larger anomaly rates. However, future work includes collecting more samples from the testbed and expanding on the testbed data analysis. Another limitation of this study is our assumption that the sensor data are non-anomalous. While sensor data are not perfectly non-anomalous (e.g., subject to noise), we assume that they are collected from well-calibrated sensors, and hence any minor anomalies present in the sensor data will not take away from the purpose of our study.

Furthermore, future models may be developed to perform data fusion to potentially improve model performance. That is, for example, data of various types (e.g., temperature and flow current) can be fused as part of the greater anomaly detection scheme to provide a more accurate picture of the system state. Future work also includes error identification. That is, instead of simply detecting whether or not an anomaly/a HE has occurred, the models may be set to identify specific error types. This can allow for presenting the human operator with meaningful messages that further enable effective error mitigation. Hence, the resulting approach can serve as the basis for a HITL system that synergizes with humans to promote smooth NPP operations.

CRediT authorship contribution statement

Ezgi Gursel: Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing. **Bhavya Reddy:** Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing. **Anahita Khojandi:** Conceptualization, Methodology, Funding acquisition, Supervision, Writing – review & editing. **Mahboubeh Madadi:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing. **Jamie Baalis Coble:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing. **Vivek Agarwal:** Funding acquisition, Writing – review & editing. **Vaibhav Yadav:** Funding acquisition. **Ronald L. Boring:** Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors 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 of authorship was prepared as an account of work sponsored by the U.S. Department of Energy, an agency of the U.S. Government, under Grant No. DE-NE0008978 to University of Tennessee-Knoxville. The authors would like to thank the Tennessee Valley Authority (TVA) for providing the research team with real-world NPP data for use in this study. The authors would also like to thank Erica Christine Swift and Stephen Lamar Farlett, representatives of the TVA, for facilitating NPP data access for the authors. Additionally, the authors would like to thank Klaus Blache and the Reliability and Maintainability Center at the University of Tennessee, Knoxville for providing the necessary equipment for testbed data collection, and Darrell Russell for the testbed data curation. Neither the U.S. Government, nor any agency thereof, nor any of their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

Appendix A

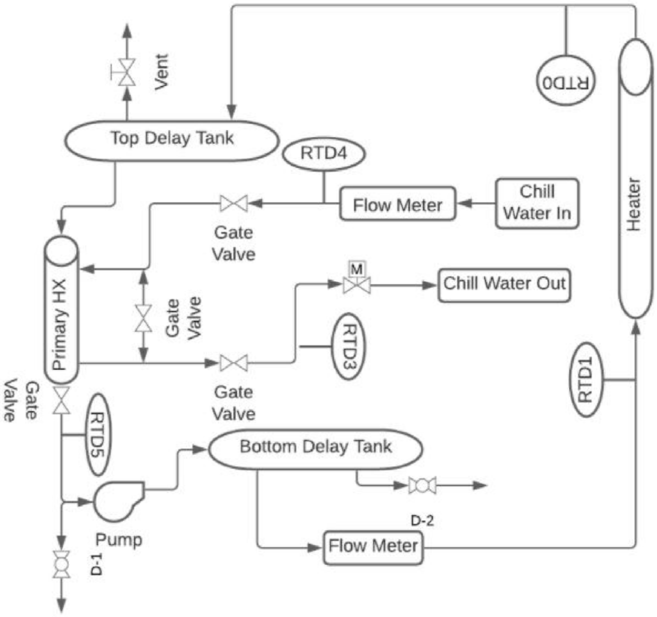

Fig. A.1. Testbed flow loop diagram

Table A.1
1D GAN Architecture

Layer	Kernel size	Stride size	Output size
Generator's Encoder			
Input			(11, 26, 1)
Conv (LeakyReLU/Batchnorm)	(1,5)	(1,1)	(11, 26, 32)
Conv (LeakyReLU/Batchnorm)	(1,3)	(2,2)	(6, 13, 64)
Conv (LeakyReLU/Batchnorm)	(1,3)	(2,2)	(3, 7, 128)
Conv (LeakyReLU/Batchnorm)	(1,3)	(2,2)	(2, 4, 128)
Global Average Pooling			(None, 128)
Generator's Decoder			
Input			(None, 128)
ConvTran (ReLU)	(1,3)	(1,1)	(3, 8, 128)
ConvTran (ReLU)	(1,3)	(1,1)	(5, 20, 64)
ConvTran (ReLU)	(1,3)	(1,1)	(7, 22, 32)
ConvTran (ReLU)	(1,3)	(1,1)	(9, 24, 32)
ConvTran (Tanh)	(1,3)	(1,1)	(11, 26, 1)
Encoder			
Input			(11, 26, 1)
Conv (LeakyReLU/Batchnorm)	(1,5)	(1,1)	(11, 26, 32)
Conv (LeakyReLU/Batchnorm)	(1,3)	(2,2)	(6, 13, 64)
Conv (LeakyReLU/Batchnorm)	(1,3)	(2,2)	(3, 7, 128)
Conv (LeakyReLU/Batchnorm)	(1,3)	(2,2)	(2, 4, 128)
Discriminator			
Input	(1, 5)	(1,1)	(11, 26, 1)
Conv (LeakyReLU/Batchnorm)	(1, 3)	(2,2)	(11, 26, 32)
Conv (LeakyReLU/Batchnorm)	(1, 3)	(2,2)	(6, 13, 64)
Conv (LeakyReLU/Batchnorm)	(1, 3)	(2,2)	(3, 7, 128)
Dense (Sigmoid)			(2, 4, 128)
Optimizer	Adam		
Learning rate	0.002		

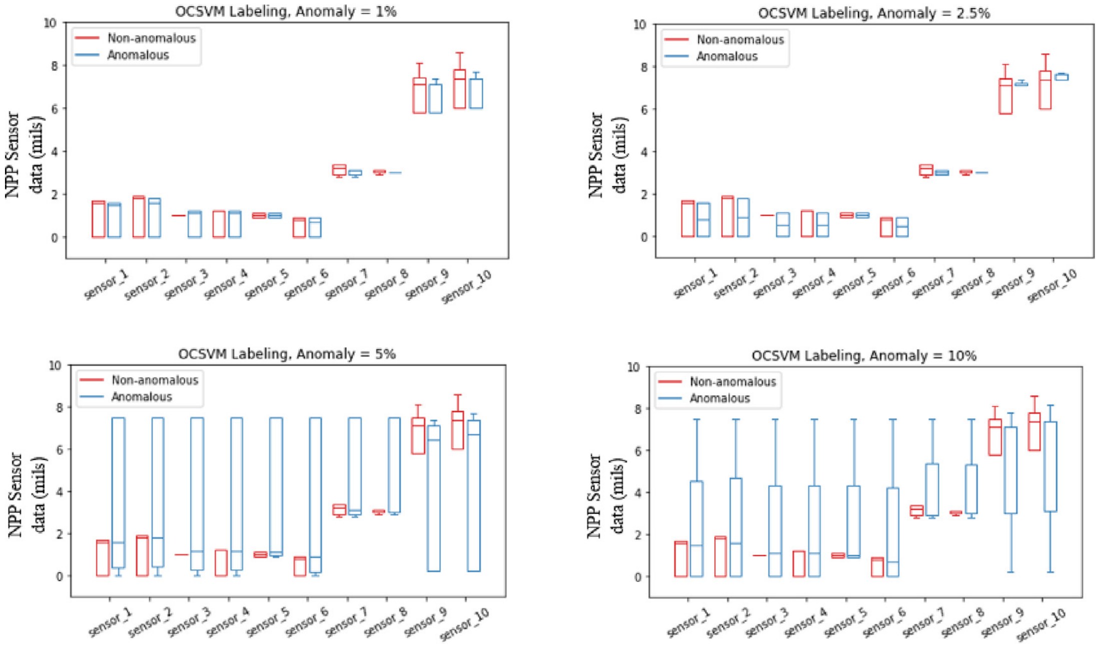

Fig. A.2. Boxplots of normal and anomalous NPP sensor data (mils) under anomaly rates $\xi \in \{1\%, 2.5\%, 5\%, 10\%\}$, as labeled by OCSVM

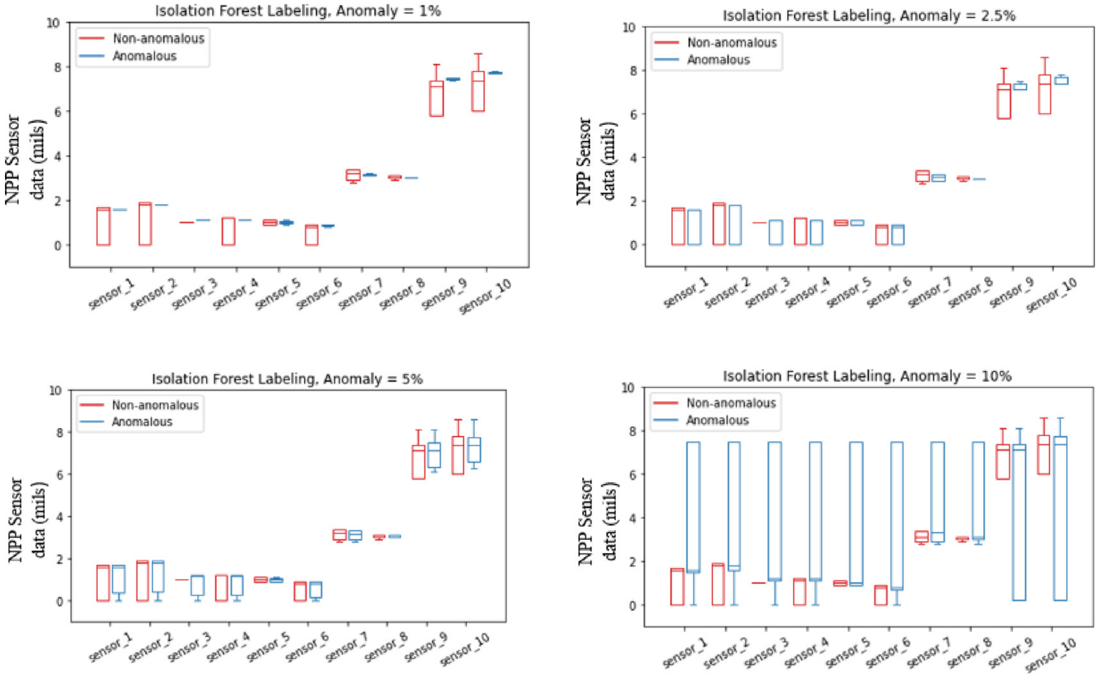

Fig. A.3. Boxplots of normal and anomalous NPP sensor data (mils) under anomaly rates $\xi \in \{1\%, 2.5\%, 5\%, 10\%\}$, as labeled by iForest

Table A.2
Training, validation, and testing set split for NPP dataset [N: Non-anomalous; A: Anomalous]

	ξ	Train set		Validation set		Test set	
		N	A	N	A	N	A
OCSVM Labeled Dataset	1%	129	2	28	1	28	1
	2.5%	128	4	28	1	27	1
	5%	125	7	27	2	27	1
	10%	118	13	26	3	25	3
iForest Labeled Dataset	1%	130	1	29	1	28	1
	2.5%	128	3	28	1	28	1
	5%	125	7	27	2	27	1
	10%	118	13	26	3	25	3

Table A.3
Training, validation, and testing split for testbed dataset, either location

ξ	Train set		Validation set		Test set	
	N	A	N	A	N	A
10%	22	2	5	1	5	1
20%	22	5	5	2	5	1

Table A.4
Training, validation, and testing split for testbed dataset, location 1

ξ	Train set		Validation set		Test set	
	N	A	N	A	N	A
10%	23	2	6	1	5	1
20%	23	4	6	1	5	1

Table A.5
Mean and standard deviation of the performance metrics for NPP I: OCSVM Labeled Anomaly Detection under window $\mathcal{W}_2$ (Results bolded if statistically better at $\alpha = 0.05$)

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
1%	iForest	**0.966 ± 0.000**	**1.000 ± 0.000**	0.000 ± 0.000	0.000 ± 0.000
	GAN-based	0.776 ± 0.121	0.768 ± 0.125	**1.000 ± 0.000**	**0.874 ± 0.071**
2.5%	iForest	**0.936 ± 0.028**	**0.970 ± 0.029**	0.000 ± 0.000	0.000 ± 0.000
	GAN-based	0.854 ± 0.064	0.848 ± 0.067	**1.000 ± 0.000**	**0.920 ± 0.036**
5%	iForest	**0.939 ± 0.041**	**0.963 ± 0.030**	0.300 ± 0.483	0.298 ± 0.480
	GAN-based	0.789 ± 0.184	0.781 ± 0.191	**1.000 ± 0.000**	**0.878 ± 0.108**
10%	iForest	0.882 ± 0.029	**0.976 ± 0.028**	0.100 ± 0.161	0.171 ± 0.275
	GAN-based	0.850 ± 0.093	0.888 ± 0.123	**0.533 ± 0.233**	**0.666 ± 0.114**

Table A.6
Mean and standard deviation of the performance metrics for NPP-II: iForest Labeled Anomaly Detection under window $\mathcal{W}_2$ (Results bolded if statistically better at $\alpha = 0.05$)

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
1%	OCSVM	**0.714 ± 0.213**	**0.732 ± 0.216**	0.200 ± 0.422	0.183 ± 0.387
	GAN-based	0.824 ± 0.162	0.825 ± 0.171	**1.000 ± 0.000**	**0.900 ± 0.094**
2.5%	OCSVM	**0.859 ± 0.093**	**0.879 ± 0.103**	0.300 ± 0.483	0.269 ± 0.435
	GAN-based	0.662 ± 0.130	0.650 ± 0.135	**1.000 ± 0.000**	**0.802 ± 0.083**
5%	OCSVM	**0.932 ± 0.043**	**0.956 ± 0.042**	0.300 ± 0.483	0.292 ± 0.471
	GAN-based	0.736 ± 0.150	0.726 ± 0.155	**1.000 ± 0.000**	**0.848 ± 0.092**
10%	OCSVM	0.829 ± 0.067	0.892 ± 0.068	0.300 ± 0.246	0.428 ± 0.307
	GAN-based	0.654 ± 0.153	0.657 ± 0.186	0.633 ± 0.246	0.616 ± 0.110

Table A.7
Mean and standard deviation of the performance metrics for NPP-I: OCSVM Labeled Anomaly Detection under window $\mathcal{W}_3$ (Results bolded if statistically better at $\alpha = 0.05$)

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
1%	iForest	**0.966 ± 0.000**	**1.000 ± 0.000**	0.000 ± 0.000	0.000 ± 0.000
	GAN-based	0.790 ± 0.171	0.861 ± 0.167	**0.921 ± 0.167**	**0.879 ± 0.100**
2.5%	iForest	**0.950 ± 0.025**	**0.985 ± 0.026**	0.000 ± 0.000	0.000 ± 0.000
	GAN-based	0.767 ± 0.142	0.785 ± 0.165	**0.974 ± 0.082**	**0.785 ± 0.165**
5%	iForest	**0.922 ± 0.033**	**0.955 ± 0.034**	0.000 ± 0.000	0.000 ± 0.000
	GAN-based	0.803 ± 0.156	0.797 ± 0.162	**1.000 ± 0.000**	**0.797 ± 0.162**
10%	iForest	**0.879 ± 0.035**	**0.972 ± 0.038**	0.100 ± 0.161	0.171 ± 0.274
	GAN-based	0.689 ± 0.069	0.684 ± 0.091	**0.734 ± 0.211**	**0.684 ± 0.091**

Table A.8
Mean and standard deviation of the performance metrics for NPP-II: iForest Labeled Anomaly Detection under window $\mathcal{W}_3$ (Results bolded if statistically better at $\alpha = 0.05$)

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
1%	OCSVM	0.740 ± 0.099	**0.750 ± 0.101**	0.444 ± 0.527	0.385 ± 0.457
	GAN-based	**0.816 ± 0.132**	0.809 ± 0.137	**1.000 ± 0.000**	**0.896 ± 0.077**
2.5%	OCSVM	**0.805 ± 0.113**	**0.845 ± 0.129**	0.212 ± 0.421	0.194 ± 0.386
	GAN-based	0.763 ± 0.188	0.754 ± 0.194	**1.000 ± 0.000**	**0.862 ± 0.113**
5%	OCSVM	**0.877 ± 0.098**	**0.893 ± 0.105**	0.444 ± 0.527	0.411 ± 0.490
	GAN-based	0.790 ± 0.131	0.782 ± 0.135	**1.000 ± 0.000**	**0.881 ± 0.078**
10%	OCSVM	**0.829 ± 0.056**	**0.907 ± 0.066**	0.185 ± 0.242	0.266 ± 0.323
	GAN-based	0.595 ± 0.056	0.582 ± 0.053	**0.704 ± 0.111**	**0.639 ± 0.073**

Table A.9
Mean and standard deviation of the performance metrics for Testbed-III when using all data from either location using FFT (Results bolded if statistically better at $\alpha = 0.05$)

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
10%	iForest	0.817 ± 0.146	0.880 ± 0.139	0.900 ± 0.316	0.467 ± 0.497
	GAN-based	**0.850 ± 0.160**	**0.820 ± 0.220***	**1.000 ± 0.000**	**0.899 ± 0.112**
	OCSVM	0.450 ± 0.177	0.440 ± 0.207	0.500 ± 0.527	0.307 ± 0.342
20%	iForest	0.833 ± 0.166	0.820 ± 0.220	0.900 ± 0.316	0.467 ± 0.497
	GAN-based	**0.883 ± 0.158**	**0.860 ± 0.189**	**1.000 ± 0.000**	**0.922 ± 0.107**
	OCSVM	0.467 ± 0.153	0.540 ± 0.165	0.100 ± 0.316	0.078 ± 0.245

* and ‡ denote significance with respect to OCSVM and iForest, respectively.

Table A.10
Mean and standard deviation of the performance metrics for Testbed-III when using the data from location 1 only using FFT (Results bolded if statistically better at $\alpha = 0.05$)

ξ	Model	Accuracy	Sensitivity	Specificity	G-Mean
10%	iForest	**0.833 ± 0.111**	**0.860 ± 0.135**	0.700 ± 0.483	0.635 ± 0.443
	GAN-based	0.817 ± 0.123	0.779 ± 0.148	**1.000 ± 0.000**	**0.879 ± 0.084**
	OCSVM	0.550 ± 0.209	0.580 ± 0.220	0.400 ± 0.516	0.308 ± 0.402
20%	iForest	**0.833 ± 0.176**	**0.820 ± 0.199**	0.900 ± 0.316	0.809 ± 0.308
	GAN-based	0.780 ± 0.111	0.740 ± 0.135	**1.000 ± 0.000**	**0.857 ± 0.078**
	OCSVM	0.550 ± 0.209	0.580 ± 0.220	0.400 ± 0.516	0.308 ± 0.402

* and ‡ denote significance with respect to OCSVM and iForest, respectively.

References

[1] Jeffrey Thomas, Clifford C. Baker, Thomas B. Malone, John T. Malone, Christina L. Hard, Ivan C.L. Rezende, Sally Caruana, Mark Witten, Application of Human Factors in Reducing Human Error in Existing Offshore Facilities, 34, United States Department of Transportation - Publication & Papers, 2002. https://digitalcommons.unl.edu/usdot/34.

[2] Seongkeun Kang, Poong Hyun Seong, Performance shaping factor taxonomy for human reliability analysis on mitigating nuclear power plant accidents caused by extreme external hazards, Annals of Nuclear Energy 145 (2020), 107533.

[3] Jinkyun Park, Hee Eun Kim, Inseok Jang, Empirical estimation of human error probabilities based on the complexity of proceduralized tasks in an analog environment, Nuclear Engineering and Technology (2021).

[4] S Dhillon Balbir, Safety, Reliability, Human Factors, and Human Error in Nuclear Power Plants, CRC Press, 2017.

[5] KONIS (KOrea Hydro & Nuclear Power Company Nuclear Information System).

[6] David Gertman, Harold Blackman, Julie Marble, James Byers, Curtis Smith, et al., The spar-h human reliability analysis method, US Nuclear Regulatory Commission 230 (4) (2005) 35.

[7] Seong Poong Hyun, Kang Hyun Gook, Man Gyun Na, Jong Hyun Kim, Gyunyoung Heo, Yoensub Jung, Advanced mmis toward substantial reduction in human errors in npps, Nuclear Engineering and Technology 45 (2) (2013) 125–140.

[8] Cha Kab-Mun, Lee Hyun-Chul, A novel qeeg measure of teamwork for human error analysis: an eeg hyperscanning study, Nuclear Engineering and Technology 51 (3) (2019) 683–691.

[9] Ahmad Al Rashdan, Michael Griffel, Roger Boza, Donna Guillen, Subtle Process Anomalies Detection Using Machine Learning Methods, Idaho National Laboratory, Idaho Falls, ID (USA), 2019. Technical Report INL/EXT-19-55629.

[10] Jung Sung Kang, Seung Jun Lee, Concept of an intelligent operator support system for initial emergency responses in nuclear power plants, Nuclear Engineering and Technology (2022).

[11] Won Chul Cho, Tae Ho Ahn, A classification of electrical component failures and their human error types in south Korean npps during last 10 years, Nuclear Engineering and Technology 51 (3) (2019) 709–718.

[12] E. Swaton, V. Neboyan, L. Lederman, Human factors in the operation of nuclear power plants, IAEA Bulletin 29 (4) (1987) 27–30.

[13] Hardik A. Gohel, Himanshu Upadhyay, Leonel Lagos, Kevin Cooper, Andrew Sanzetenea, Predictive maintenance architecture development for nuclear infrastructure using machine learning, Nuclear Engineering and Technology 52 (7) (2020) 1436–1442.

[14] Jacques V. Hugo, David I. Gertman, A method to select human–system interfaces for nuclear power plants, Nuclear Engineering and Technology 48 (1) (2016) 87–97.

[15] Robert E. Uhrig, Use of Artificial Intelligence to Enhance the Safety of Nuclear Power Plants, Oak Ridge National Laboratory, Oak Ridge, TN (USA), 1988. Technical report.

[16] Meenu Sethu, Nesar Ahmed Titu, Dingyu Hu, Mahboubeh Madadi, Jamie Coble, Ronald Boring, Klaus Blache, Vivek Agarwal, Vaibhav Yadav, Anahita Khojandi, Using artificial intelligence to mitigate human factor errors in nuclear power plants: a review, in: 12th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPICHMIT 2021), ANS, 2021, pp. 129–141.

[17] Min-Han Hsieh, Sheue-Ling Hwang, Kang-Hong Liu, Sheau- Farn Max Liang, Chang-Fu Chuang, A decision support system for identifying abnormal operating procedures in a nuclear power plant, Nuclear Engineering and Design 249 (2012) 413–418.

[18] Kwae Hwan Yoo, Ju Hyun Back, Man Gyun Na, Seop Hur, Hyeonmin Kim, Smart support system for diagnosing severe accidents in nuclear power plants, Nuclear Engineering and Technology 50 (4) (2018) 562–569.

[19] Yuyun Zeng, Jingquan Liu, Kaichao Sun, Lin-wen Hu, Machine learning based system performance prediction model for reactor control, Annals of Nuclear Energy 113 (2018) 270–278.

[20] Mario Gomez Fernandez, Akira Tokuhiro, Welter Kent, Qiao Wu, Nuclear energy system's behavior and decision making using machine learning, Nuclear Engineering and Design 324 (2017) 27–34.

[21] Varun Chandola, Arindam Banerjee, Vipin Kumar, Anomaly detection: a survey, ACM Computing Surveys (CSUR) 41 (3) (2009) 1–58.

[22] Nari S. Arunraj, Robert Hable, Michael Fernandes, Karl Leidl, Michael Heigl, Comparison of supervised, semi-supervised and unsupervised learning methods in network intrusion detection system (nids) application, Anwendungen und Konzepte der Wirtschaftsinformatik 6 (2017).

[23] Villa-Pérez Miryam Elizabeth, Miguel Á. Álvarez-Carmona, Octavio Loyola-González, Miguel Angel Medina-Pérez, Juan Carlos Velazco-Rossell, Kim-Kwang Raymond Choo, Semi-supervised anomaly detection algorithms: a comparative summary and future research directions, Knowledge-Based Systems (2021), 106878.

[24] Prasanta Gogoi, K. Dhruba, Bhattacharyya, Bhogeswar Borah, Jugal, K. Kalita, A survey of outlier detection methods in network anomaly identification, The Computer Journal 54 (4) (2011) 570–588.

[25] Ahmad Al Rashdan, St Shawn, Germain. Methods of data collection in nuclear power plants, Nuclear Technology 205 (8) (2019) 1062–1074.

[26] Hermine N. Akouemo, Richard J. Povinelli, Probabilistic anomaly detection in natural gas time series data, International Journal of Forecasting 32 (3) (2016) 948–956.

[27] Thomas B. Sheridan, Understanding human error and aiding human diagnostic behaviour in nuclear power plants, in: Human Detection and Diagnosis of System Failures, Springer, 1981, pp. 19–35.

[28] Jooyoung Park, Wondea Jung, Jonghyun Kim, Inter-relationships between performance shaping factors for human reliability analysis of nuclear power plants, Nuclear Engineering and Technology 52 (1) (2020) 87–100.

[29] Gyunyoung Heo, Jinkyun Park, A framework for evaluating the effects of maintenance-related human errors in nuclear power plants, Reliability Engineering & System Safety 95 (7) (2010) 797–805.

[30] Jaemin Yang, Jonghyun Kim, An accident diagnosis algorithm with untrained accident identification, in: Transactions of the Korean Nuclear Society Spring Meeting, Jeju, Korea, 2019.

[31] Yochan Kim, Yung Hsien James Chang, Jinkyun Park, Criscione Lawrence, Sacada and hurex part 2: the use of sacada and hurex data to estimate human error probabilities, Nuclear Engineering and Technology 54 (3) (2022) 896–908.

[32] Jinkyun Park, Yochan Kim, Wondea Jung, Calculating nominal human error probabilities from the operation experience of domestic nuclear power plants, Reliability Engineering & System Safety 170 (2018) 215–225.

[33] Wolfgang Preischl, Mario Hellmich, Human error probabilities from operational experience of German nuclear power plants, Reliability Engineering & System Safety 109 (2013) 150–159.

[34] Markus Goldstein, Seiichi Uchida, A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data, PloS One 11 (4) (2016), e0152173.

[35] Jianping Ma, Jin Jiang, Applications of fault detection and diagnosis methods in nuclear power plants: a review, Progress in Nuclear Energy 53 (3) (2011) 255–266.

[36] Luis Martí, Nayat Sanchez-Pi, José Manuel Molina, Ana Cristina Bicharra Garcia, Anomaly detection based on sensor data in petroleum industry applications, Sensors 15 (2) (2015) 2774–2797.

[37] Prabhas Hundi, Rouzbeh Shahsavari, Comparative studies among machine learning models for performance estimation and health monitoring of thermal power plants, Applied Energy 265 (2020), 114775.

[38] Ioannou George, Thanos Tagaris, Georgios Alexandridis, Andreas Stafylopatis, Intelligent techniques for anomaly detection in nuclear reactors, in: EPJ Web of Conferences, ume 247, EDP Sciences, 2021, p. 21011.

[39] Xingang Zhao, Junyung Kim, Kyle Warns, Xinyan Wang, Pradeep Ramuhalli, Sacit Cetiner, Hyun Gook Kang, Michael Golay, Prognostics and health management in nuclear power plants: an updated method-centric review with special focus on data-driven methods, Frontiers in Energy Research 9 (2021) 294.

[40] Xin Jin, Yin Guo, Soumik Sarkar, Asok Ray, Robert M. Edwards, Anomaly detection in nuclear power plants via symbolic dynamic filtering, IEEE Transactions on Nuclear Science 58 (1) (2010) 277–288.

[41] Minhee Kim, Elisa Ou, Po-Ling Loh, Todd Allen, Robert Agasie, Kaibo Liu, Rnn-based online anomaly detection in nuclear reactors for highly imbalanced datasets with uncertainty, Nuclear Engineering and Design 364 (2020), 110699.

[42] Le Zhang, Wei Cheng, Xue Liu, Xuefeng Chen, Fengtian Chang, Junying Hong, Xiaofei Li, System-level anomaly detection for nuclear power plants using variational graph auto-encoders, in: 2021 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), IEEE, 2021,

pp. 180–185.

[43] Francesco Caliva, Fabio Sousa De Ribeiro, Antonios Mylonakis, Christophe Demazi'ere, Paolo Vinai, Georgios Leontidis, Stefanos Kollias, A deep learning approach to anomaly detection in nuclear reactors, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[44] J. Tylee, On-line failure detection in nuclear power plant instrumentation, IEEE Transactions on Automatic Control 28 (3) (1983) 406–415.

[45] Lex Poon, Siamak Farshidi, Na Li, Zhiming Zhao, Unsupervised anomaly detection in data quality control, in: 2021 IEEE International Conference on Big Data (Big Data), IEEE, 2021, pp. 2327–2336.

[46] Daiki Nishiura, Isao Nambu, Yoshiko Maruyama, Yasuhiro Wada, Improvement of human error prediction accuracy in single-trial analysis of electroencephalogram, in: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), IEEE, 2021, pp. 6179–6182.

[47] Ji Hun Park, Hye Seon Jo, Man Gyun Na, System and component anomaly detection using lstm-vae, in: 2021 5th International Conference on System Reliability and Safety (ICSRS), IEEE, 2021, pp. 131–137.

[48] Junyong Bae, Jeeyea Ahn, Seung Jun Lee, Comparison of multilayer perceptron and long short-term memory for plant parameter trend prediction, Nuclear Technology 206 (7) (2020) 951–961.

[49] Jeeyea Ahn, Junyong Bae, Seung Jun Lee, A Human Error Detection System in Nuclear Power Plant Operations. In *11th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPICHMIT)*, American Nuclear Society, 2019.

[50] Houssam Zenati, Chuan Sheng Foo, Lecouat Bruno, Gaurav Manek, Vijay Ramaseshan Chandrasekhar, Efficient gan-based Anomaly Detection, 2018 *arXiv preprint arXiv:1802.06222*.

[51] Federico Di Mattia, Paolo Galeone, Michele De Simoni, Emanuele Ghelfi, A Survey on Gans for Anomaly Detection, 2019 *arXiv preprint arXiv: 1906.11632*.

[52] Xuan Xia, Xizhou Pan, Nan Li, He Xing, Ma Lin, Xiaoguang Zhang, Ning Ding, Gan-Based Anomaly Detection: A Review, Neurocomputing, 2022.

[53] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial networks, Communications of the ACM 63 (11) (2020) 139–144.

[54] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, Georg Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in: International Conference on Information Processing in Medical Imaging, Springer, 2017, pp. 146–157.

[55] Dan Li, Dacheng Chen, Lei Shi, Baihong Jin, Jonathan Goh, See-Kiong Ng, in: Mad-gan: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks, ICANN, 2019.

[56] Yeji Choi, Hyunki Lim, Heeseung Choi, Ig-Jae Kim, Gan-based anomaly detection and localization of multivariate time series data for power plant, in: 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, 2020, pp. 71–74.

[57] Seung Geun Kim, Ho Chae Young, Seong Poong Hyun, Development of a generative-adversarial-network-based signal reconstruction method for nuclear power plants, Annals of Nuclear Energy 142 (2020), 107410.

[58] Xiangyu Li, Tao Huang, Kun Cheng, Zhifang Qiu, Sichao Tan, Research on anomaly detection method of nuclear power plant operation state based on unsupervised deep generative model, Annals of Nuclear Energy 167 (2022), 108785.

[59] Anders Brandt, Noise and Vibration Analysis: Signal Analysis and Experimental Procedures, John Wiley & Sons, 2011.

[60] Mark Mitchell, Baurzhan Muftakhidinov, Winchen Tobias, B. van Schaik, A.K. Wilms, et al., Engauge digitizer software, Webpage (11, 2017). http://markummitchell.github.io/engauge-digitizer.

[61] Duygu Bayram, Serhat Şeker, Redundancy-based predictive fault detection on electric motors by stationary wavelet transform, IEEE Transactions on Industry Applications 53 (3) (2016) 2997–3004.

[62] Hocine Bendjama, Salah Bouhouche, Mohamed Seghir Boucherit, Application of wavelet transform for fault diagnosis in rotating machinery, International Journal of machine Learning and computing 2 (1) (2012) 82–87.

[63] Wahyu Caesarendra, Tegoeh Tjahjowidodo, A review of feature extraction methods in vibration-based condition monitoring and its application for degradation trend estimation of low-speed slew bearing, Machines 5 (4) (2017) 21.

[64] Salim Lahmiri, Wavelet low-and high-frequency components as features for predicting stock prices with backpropagation neural networks, Journal of King Saud University-Computer and Information Sciences 26 (2) (2014) 218–227.

[65] Samet Akcay, Amir Atapour-Abarghouei, Toby P. Breckon, Ganomaly: semi-supervised anomaly detection via adversarial training, in: Asian Conference on Computer Vision, Springer, 2018, pp. 622–637.

[66] Dan Li, Dacheng Chen, Jonathan Goh, See-kiong Ng, Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series, 2018 *arXiv preprint arXiv:1809.04758*.

[67] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, See-Kiong Ng, Mad-gan: multivariate anomaly detection for time series data with generative adversarial networks, in: International Conference on Artificial Neural Networks, Springer, 2019, pp. 703–716.

[68] Ferhat Özgür Çatak, Robust ensemble classifier combination based on noise removal with one-class svm, in: International Conference on Neural Information Processing, Springer, 2015, pp. 10–17.

[69] Katherine Heller, Krysta Svore, Angelos D. Keromytis, Salvatore Stolfo, One Class Support Vector Machines for Detecting Anomalous Windows Registry Accesses, Columbia Academic Commons, 2003, https://doi.org/10.7916/D85M6CFF.

[70] Shen Yin, Xiangping Zhu, Jing Chen, Fault detection based on a robust one class support vector machine, Neurocomputing 145 (2014) 263–268.

[71] S. Sathiya Keerthi, Chih-Jen Lin, Asymptotic behaviors of support vector machines with Gaussian kernel, Neural Computation 15 (7) (2003) 1667–1689.

[72] Yingchao Xiao, Huangang Wang, Wenli Xu, Parameter selection of Gaussian kernel for one-class svm, IEEE Transactions on Cybernetics 45 (5) (2014) 941–953.

[73] Fei Tony Liu, Kai Ming Ting, Zhi-Hua Zhou, Isolation forest, in: 2008 Eighth Ieee International Conference on Data Mining, IEEE, 2008, pp. 413–422.

[74] Chandresh Kumar Maurya, Durga Toshniwal, Gopalan Vijendran Venkoparao, Online anomaly detection via class-imbalance learning, in: 2015 Eighth International Conference on Contemporary Computing (IC3), IEEE, 2015, pp. 30–35.