sciendo

JAISCR, 2025, Vol. 15, No. 1, pp. 5 – 24
10.2478/jaiscr-2025-0001

AN UNSUPERVISED ANOMALY DETECTION IN ELECTRICITY CONSUMPTION USING REINFORCEMENT LEARNING AND TIME SERIES FOREST BASED FRAMEWORK

Jihan Ghanim, Mariette Awad*

Department of Electrical and Computer Engineering, American University of Beirut, Beirut, Lebanon

**E-mail: mariette.awad@aub.edu.lb*

Submitted: 9th March 2024; Accepted: 29th September 2024

Abstract

Anomaly detection (AD) plays a crucial role in time series applications, primarily because time series data is employed across real-world scenarios. Detecting anomalies poses significant challenges since anomalies take diverse forms making them hard to pinpoint accurately. Previous research has explored different AD models, making specific assumptions with varying sensitivity toward particular anomaly types. To address this issue, we propose a novel model selection for unsupervised AD using a combination of time series forest (TSF) and reinforcement learning (RL) approaches that dynamically chooses an AD technique. Our approach allows for effective AD without explicitly depending on ground truth labels that are often scarce and expensive to obtain. Results from the real-time series dataset demonstrate that the proposed model selection approach outperforms all other AD models in terms of the F1 score metric. For the synthetic dataset, our proposed model surpasses all other AD models except for KNN, with an impressive F1 score of 0.989. The proposed model selection framework also exceeded the performance of GPT-4 when prompted to act as an anomaly detector on the synthetic dataset. Exploring different reward functions revealed that the original reward function in our proposed AD model selection approach yielded the best overall scores. We evaluated the performance of the six AD models on an additional three datasets, having global, local, and clustered anomalies respectively, showing that each AD model exhibited distinct performance depending on the type of anomalies. This emphasizes the significance of our proposed AD model selection framework, maintaining high performance across all datasets, and showcasing superior performance across different anomaly types.

Keywords: anomaly detection, reinforcement learning, model selection, time series forest, time series power consumption

1 Introduction

Anomaly detection (AD) finds widespread use in various applications [1], including smart grids [2], Internet of Things (IoT) [3], medical systems [4], financial fraud detection systems [5], and blockchain networks [6]. With the prevalence of cloud infrastructure and IoT-enabled devices, there is a vast availability of time series data [7]. Consequently, AD for time series data is gaining at-

traction, particularly in data-driven applications [8]. Anomalies, also called outliers, represent observations that deviate from the rest of the dataset [9] or exhibit deviations from most of the sequence's distributions or patterns [10]. For example, anomalies in power grid meter measurements may indicate potential cyber-attacks or malfunctions [11, 12]. Similarly, anomalies in time series power consumption patterns can lead to errors in power consumption predictions, resulting in underproduction or overproduction of electricity [13]. In finance, outliers in time series can signify "illegal activities like fraud, risk, network intrusions, account takeover, and money laundering" [5]. Thus, time series AD plays a crucial role in ensuring operational efficiency, reliability, and security in real-world applications [8].

However, AD often falls short of achieving expected performance in data-driven real-world systems. This is primarily due to the scarcity, cost, and impracticality of obtaining labeled data in large quantities [8]. In real-world applications, a limited quantity of labeled anomalies can be provided through methods like active learning or identification by domain experts [14, 15, 16]. Therefore, it is preferable to employ unsupervised AD techniques that do not rely on labeled data [8].

Furthermore, given the complexity of data in real-world systems, data abnormalities manifest in various forms [17]. AD techniques that assume specific abnormalities may tend to rely on certain aspects more than others, making them sensitive to specific types of anomalies while potentially overlooking others. Consequently, there is a need for a universal AD model that outperforms other AD models on different types of data [1]. To address this, leveraging multiple AD models at different time steps can be beneficial when applied to a particular time series data. This can be achieved by designing an AD selection model that chooses the optimal AD model from a pool of candidate models at each time step [18]. The predicted label (normal or anomalous) at each time step depends on the output of the currently selected anomaly detector. Reinforcement learning (RL), a machine learning domain that maximizes numerical rewards by mapping situations to actions at each time step, can be employed for this purpose [19].

In a RL setting, an agent aims to learn the optimal decision-making policy to maximize the overall reward. RL has demonstrated success in various model selection applications, such as RL model selection for wind speed prediction [20] and RL-based model combination for time series forecasting [21].

This study proposes an RL-based model selection approach for AD (RLAD) in time series datasets. To address the challenge of gathering labeled data which is both expensive and time-consuming [22], we focus on six unsupervised AD techniques that do not require labeled data for training. To implement the RL agent without explicit reliance on ground truth labels in its reward function, we utilize six time series forest (TSF) models. Each TSF is trained to classify a specific AD model's incorrect and correct predictions. Our proposed AD model selection framework utilizes the classifications from these six classifiers for the majority of data points and ground truth labels for the remaining data points [23]. In other words, our approach requires a portion of ground truth labels instead of explicit reliance on all ground truth labels. We compare our RL-based model selection approach with an RL model selection that solely relies on ground truth labels in its reward function and another approach that relies only on the classifications of the TSFs in its reward function.

Our proposed RLAD framework yields promising results, outperforming all AD candidate models on the real dataset and all AD models except for KNN on the synthetic dataset. Additionally, we conduct a study on different reward scenarios, including adaptive rewards with varying exploration epsilons, in the context of our AD model selection approach. The results indicate that the original reward function with a decaying exploration rate produces the best scores in terms of precision, recall, and F1 score evaluation metrics. We also examine the impact of different types of anomalies on the performance of the AD techniques by testing them on three synthetic datasets: one with local anomalies, another with global anomalies, and a third with clustered anomalies. The performance of each AD model varies across different types of anomalies, whereas our proposed AD model selection approach consistently achieves high scores

across global, local, and clustered datasets, respectively.

2 Related Work

Due to the emerging spread of time series problems [24], the field of AD has garnered substantial attention for the analysis of time series data [25]. It can be categorized into three main groups: unsupervised, semi-supervised, and supervised methods [26]. While supervised or semi-supervised methods can utilize previously identified abnormalities to label instances as normal or abnormal, the practicality of supervised techniques is limited by the need for a large amount of labeled data containing both normal and abnormal instances, which is often not feasible in real-world settings [27]. Semi-supervised AD techniques typically stem from supervised algorithms by incorporating a bias term to account for unlabeled data [28, 29]. In light of this, unsupervised AD categories, such as distance-based, statistical, ensemble-based, and reconstruction-based approaches, have recently gained favor.

In this section, we comprehensively understand these prominent categories of unsupervised AD techniques and their applicability in various scenarios. After exploring the strengths and limitations of these techniques, we discuss the recent advancements in this field, with the goal of tackling the existing AD challenges effectively.

2.1 Distance-based Approaches

Among the distance-based AD techniques, [30] considers a point (p) as an (π, ε)-anomaly if at most $(\pi)\%$ of the entire data points are away from p by a distance less than (ε). This was further improved in [31] by proposing an outlier score which is obtained by calculating the distance to the k^{th} nearest neighbor (KNN) of a data point. After that, the data point is classified as normal or anomalous based on a threshold. [32] suggested a better variant with a KNN-based method where the outlier score represents the aggregate distance of the data point (p) to the k nearest neighbors of p. Another distance-based AD algorithm, the Local Outlier Factor (LOF), considered the ratio of the average of the k nearest neighbors densities to the data point density itself [33]. While LOF and KNN have both been shown to outperform state-of-the-art (SOTA) AD techniques, they yet failed to preserve a high performance with large high-dimensional and seasonal datasets [34].

2.2 Statistical Approaches

Numerous seminal probabilistic AD methods compute the anomaly score relying on the estimates of the marginal likelihood $\mathbb{P}_\theta(X)$ derived from the data generation model [35, 36, 37]. One of the probabilistic techniques that fits a number of Gaussians on the data is Gaussian Mixture Model (GMM). To estimate the models' parameters, the expectation-maximization algorithm (EM) is utilized [38]. Support vector machine (SVM) has been extensively used for novelty detection tasks [39]. One-class SVM (OSVM) is a support-vector-based approach that assumes a probability density (like GMMs fitting) to learn a hyper-plane that separates the low-density region from the region with a high-density of data [40, 41]. One limitation of generative models lies in the complexity of obtaining marginal likelihoods, which necessitates computing integrals with exceedingly high dimensions. $\mathbb{P}_\theta(X) = \int \mathbb{P}_\theta(z)\mathbb{P}_\theta(X \mid z)\,dz$ [8]. More recent non-parametric statistical methods for AD are Empirical Cumulative Distribution for Outlier Detection (ECOD) [42] and Copula-Based Outlier Detection (COPOD) [43]. The ECOD approach involves computing an empirical cumulative distribution to predict tail probabilities for individual data points, which are then used to calculate the outlier score. In contrast, COPOD utilizes an empirical copula to estimate tail probabilities for the entire set of data instances, enabling the computation of the outlier score. Although these novel techniques provide valuable contributions, their scalability is constrained by their dependence on the data's shape [44].

2.3 Ensemble-based Approaches

In [45], an ensemble-based AD technique considered Local Outlier Factor (LOF) with multiple hyper-parameter sets and combined the results to obtain outlier scores. Another ensemble-based technique, known as Isolation Forest (IForest), was proposed in [46]. IForest consists of a forest with random binary trees in such a way that outliers are relatively near the tree's root (within a shal-

low depth of the tree). The underlying principle of IForest is based on the notion that outliers are more prone to isolation.

2.4 Reconstruction-based Approaches

Several contemporary AD algorithms capitalize on using synthetic data reconstruction. These algorithms are based on the insight that outliers, when projected into a lower-dimensional space, tend to lose information, leading to less efficient reconstruction and an increased reconstruction error for anomalous instances [8]. Principal Component Analysis (PCA) is a popular approach for data reconstruction that permits linear reconstruction only [47]. The non-linear variant of PCA, known as Kernel PCA, applies the kernel trick to map the data into a feature space [48]. Deep learning-based AD algorithms, such as those employing Recurrent Neural Networks (RNN), have garnered significant attention, especially in time series applications [49, 50]. However, these techniques can be computationally expensive and time-consuming [51, 52]. A more recent approach is Unsupervised AD on Multivariate Time Series (USAD) [53]. It relies on an encoder-decoder pair that is trained in an adversarial manner. The reconstruction error in the testing phase represents the anomaly score.

2.5 Large Language Models-based Approaches

Large Language Models (LLMs) have lately shown remarkable success in the AD task. LLMs are capable of processing and interpreting huge amounts of data having temporal dependencies which provide more complicated and precise AD compared to traditional statistical methods. Motivated by the fact that anomaly detection of time series data resembles text classification and that BERT [54] has proven effective in handling text classification, Dang et al. [55] suggested a time series anomaly detection technique relying on BERT model. Their model has the same architecture as the BERT model, however, the main distinctions are the input representation and additional output layer. Simulation results reveal that this technique only requires a small amount of labeled data for training the BERT model to achieve better results than the SOTA methods. Yet, the model scores on the KPI dataset are still not high enough recording an F1 score of 0.634. Recently, Dong et al. [56] studied the performance of zero-shot learning of LLMs in time series AD problems and their explanatory abilities. They also introduced a synthesized dataset to fine-tune LLMs and improve their performance in time series AD. Their work states that LLMs exhibit encouraging potential for time series AD, while customized prompts and instructions remain necessary. Despite that GPT-4 showed good results with minimal instructions, the present absence of public fine-tuning capabilities on GPT-4 stopped them from investigating whether fine-tuning on GPT-4 could record SOTA performance.

Leveraging LLMs for AD holds significant promise but still faces several critical challenges. One main concern is their dependence on historical datasets, which can be a matter of concern related to data availability, quality, and model bias. Moreover, these models often have trouble with generalizability, making it hard for them to use learned patterns with novel scenarios. LLMs may also generate inaccurate or misleading outputs under certain conditions, due to hallucinations, which question their reliability. At last, the computational efficiency of these large models imposes a barrier, as their resource-intensive nature can limit scalability and accessibility [57].

2.6 Model Selection for AD

Most work in AD has attempted to build novel and enhanced models to perform outlier detection on distinct data types. However, very limited work exists on the AD model selection task. Research work in [58] proposed an automated outlier detection method that applied neural architecture search to find the best neural network model. The limitation of this technique is that it is restricted to auto-encoder-based AD methods and relies on annotated data for evaluation. Another set of automated-based AD model selection techniques are TODS [59] and PyODDS [60] which also rely on labeled data. A more recent automated model selection for unsupervised outlier detection proposed a meta-learning approach without needing ground truth labels in the testing phase [61]. Yet, their approach requires ground truth labels in the meta-train datasets. These model selection methods tend to choose a single AD model after evaluation. However, a recent model selection approach applied to the Secure Water Treat-

ment (SWaT) dataset chooses dynamically an unsupervised AD technique among five different AD models using RL [1]. However, their AD pool of models does not contain a distance-based AD model. Additionally, their RL framework does not consider the anomalous dataset in its state space. Another drawback is that their RL approach relies explicitly on ground truth labels in its reward function. Due to the difficulty of obtaining a huge amount of labeled data in real-world applications, we propose a dynamic model selection for time series AD without excessively relying on ground truth labels. Our approach uses a time series forest and RL-based framework relying on a limited amount of ground truth labels to select an AD algorithm from six unsupervised AD techniques at each time step. Specifically, we will consider six various and popular AD algorithms from the above-mentioned categories: KNN, COPOD, ECOD, OSVM, IFOREST, and USAD.

3 Methodology

3.1 Unsupervised AD for Time Series

This research focuses on building a model selection for time series unsupervised AD. Figure 1 illustrates our proposed RLAD model selection. Six distinct unsupervised AD techniques are considered in our framework; KNN [32], COPOD [43], ECOD [42], OSVM [40], IForest [46], and USAD [53]. First, the training set D_{normal}, containing a time series of only normal data points, is separately fed to each unsupervised AD model. Every AD model is separately trained on the training set D_{normal}. Then, every trained model is tested on the testing set $D_{anomaly}$ having anomalous data points. Each model will output a set of anomaly scores; a score for each instance of the testing set $D_{anomaly}$. For every AD model, an anomaly threshold is empirically computed. Depending on the produced anomaly scores and thresholds, each AD model yields a predicted anomaly label for every instance of $D_{anomaly}$.

3.2 The Confidence Scores

Two confidence scores, initially used by [1], were considered in our proposed model selection framework for AD techniques. The first confidence score is derived from the concept that the greater the anomaly score, the more likely the corresponding instance is anomalous. AD models generate anomaly scores when tested on new data. The higher the predicted anomaly score exceeds the model's threshold, the more likely it is that the tested data point is an anomaly. Therefore, the extent to which the anomaly score surpasses its threshold serves as a reasonable indicator of the prediction's reliability for the given AD model. The Distance-to-Threshold Confidence Score is equivalent to the following:

$$\frac{\text{Anomaly score} - \text{Threshold}}{\max(\text{Anomaly score}) - \min(\text{Anomaly score})} \tag{1}$$

It is worth mentioning that the anomaly scores are scaled using the min-max scale to a range between zero and one before computing the Distance-to-Threshold Confidence score. Inspired by the principle of majority voting in ensemble learning, the Prediction Consensus Confidence Score indicates that as more AD models in the pool produce the same prediction (predicted label), the likelihood of the current prediction being correct increases. Precisely, the Prediction Consensus Confidence Score is determined as follows:

$$\frac{\text{Number of models predicted the same label}}{\text{Total number of models}} \tag{2}$$

3.3 Evaluation Metrics

We annotated the normal data as "negative" and the anomalous one as "positive". To evaluate the performance of our proposed approach, we apply the well-known precision, recall, and F1score metrics.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4}$$

$$\text{F1score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

where the elements of the confusion matrix are:

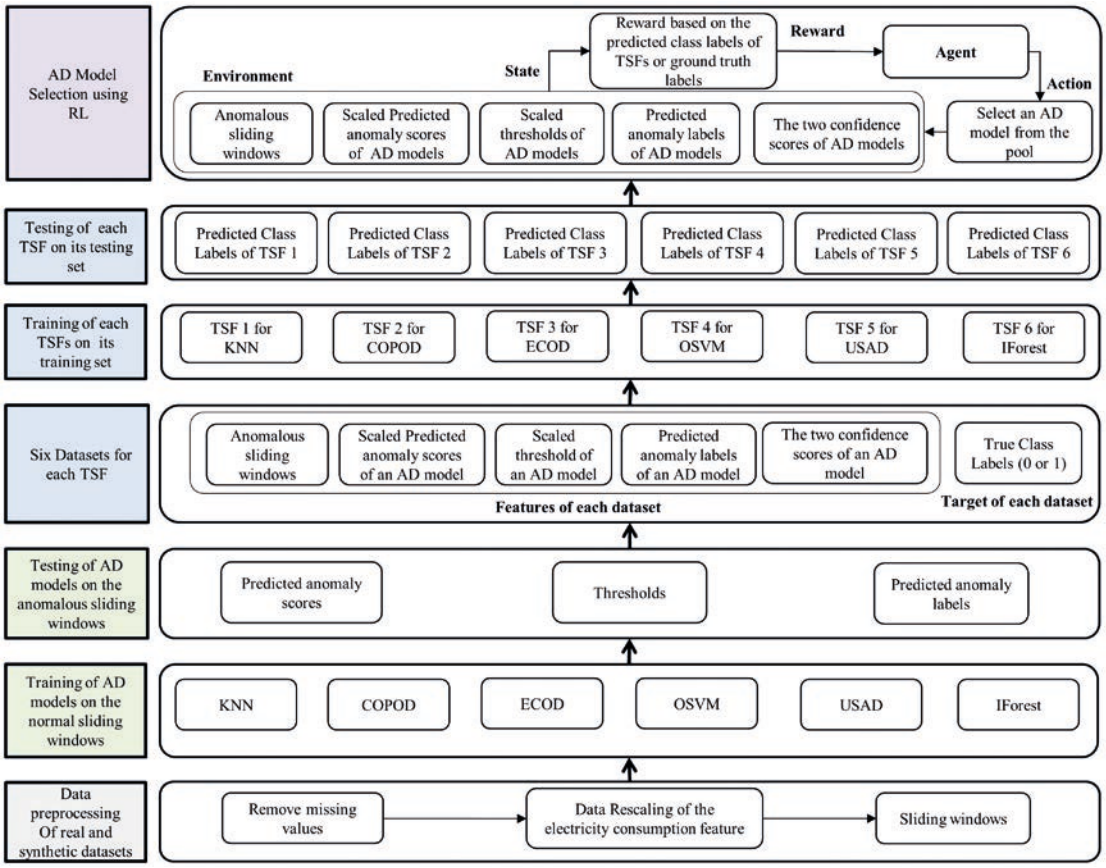


Figure 1. Illustration of the Proposed Framework

True Positive (TP): represents the case where the anomalous instance is correctly predicted.

False Positive (FP): stands for the case where the anomalous instance is predicted as a normal instance.

True Negative (TN): refers to the case where the normal instance is correctly predicted as a normal instance.

False Negative (FN): refers to the case where the normal instance was predicted as an anomalous one.

3.4 Time Series Classification Using TSF

Six TSFs are implemented; a TSF for the predicted anomaly labels of KNN, COPOD, ECOD, OSVM, IForest, and USAD, respectively. Each TSF classifies the predicted anomaly labels of a specific trained AD model (from the previous step) into two classes, C={0,1} where C=0 indicates wrongly predicted anomaly labels and C=1 indicates correctly predicted anomaly labels. In other words, if the anomaly-predicted label of a certain AD model is equivalent to the ground truth label, then it is classified as 1 otherwise it is classified as 0.

Since each TSF corresponds to a specific AD model, then each TSF is applied to a different dataset containing the anomaly predictions of its corresponding AD model. The dataset of each TSF consists of a set of features and a target. The features of each dataset represent the anomalous instances $D_{anomaly}$, along with the corresponding predicted anomaly scores (after rescaling them) obtained from the corresponding AD model, the empirical scaled threshold of the corresponding AD model, the predicted anomaly labels of the corresponding AD model, and the corresponding two confidence scores. The target of each dataset consists of the true class labels (0 or 1) belonging to its corresponding AD model. The target in each dataset of the TSFs is obtained by comparing the predicted anomaly label of its corresponding AD model to the ground truth label.

During training, each of the TSFs is trained on its corresponding training set where 20% of its corresponding dataset is considered for training. In the testing phase, each TSF is tested on its corresponding testing set, having the remaining 80% of its corresponding dataset. Each of the six TSF classifiers predicts the class labels (0 or 1). The target in the training set of each TSF is obtained by comparing the anomaly-predicted label of its corresponding AD model to the ground truth label. Thus, 20% of the anomalous instances and their ground truth labels are required to train each TSF classifier. The set of anomalous instances and ground truth labels used for all six datasets should not exceed 30% of the anomalous dataset to ensure that a limited amount of ground truth labels is only needed.

3.5 AD Model Selection Using Deep Q Network (DQN)

To tackle our model selection challenge AD, we adopt an RL framework. RL is a type of machine learning that involves an agent learning to make decisions by interacting with an environment. The agent receives feedback in the form of rewards or penalties based on its actions and uses this information to optimize its decision-making strategy over time. RL is particularly suitable for problems where the agent must learn from trial and error to maximize cumulative rewards, such as in robotics, game playing, and autonomous systems. RL can thus be conceptualized as a Markov Decision Process (MDP) where the RL agent maximizes the total expected return by learning the decision policy. MDP contains five major elements; an action-space consisting of the actions set, a state-space containing the states set, a transition matrix representing the probability of state-transition, a reward function, and a discount factor that refers to a value between 0 and 1 that is used in calculating the reward. The future return in an MDP consists of the immediate reward along with the discounted future rewards. The policy represents a probability distribution that maps the present state to the possibility of choosing a particular action.

In this work, the DQN algorithm of RL is applied. DQN is a form of RL that utilizes deep neural networks to approximate the action-value function, known as Q-function, in a high-dimensional state-action space. It combines the power of deep learning with Q-learning to enable the efficient handling of complex environments. The agent uses experience replay, storing past experiences in a memory buffer, and updates the neural network using batches of experiences, breaking the correlation between consecutive experiences. This technique allows for more stable and effective learning, en-

abling the agent to make informed decisions by selecting actions that yield the maximum expected future rewards. In our MDP modeling, the DQN agent was trained to learn a policy to select the suitable AD model among the pool at each time step. At each time step, the agent was rewarded based on its action (selection).

Given the context of a time series application, the transitions between states adhere to the chronological order of the time series dataset. As a result, the state transition is considered deterministic, eliminating the necessity to incorporate a transition matrix. Moreover, we set the discount factor to one to appropriately handle the time series nature of the problem. The remaining components of the MDP are described below;

State-space: The entire testing set $D_{anomaly}$, the scaled predicted anomaly scores from all the considered AD models, the obtained scaled empirical thresholds, and the predicted anomaly labels for these six AD models, as well as the two confidence scores of these AD models.

Action-space: The action-space is a discrete action-space. The DQN agent can select one AD model from the pool of the six AD models at each time step. Thus, the size of the action-space is the same as the number of candidate AD models in the pool.

Reward Function (R): The reward function is obtained by comparing the predicted anomaly label obtained from the selected AD model in the pool of models to the predicted class labels obtained the corresponding TSF in most of the testing set $D_{anomaly}$ (80% of $D_{anomaly}$). This represents the class-based portion of the reward. In the remaining 20%, the reward function is obtained by comparing the predicted anomaly label obtained from the selected AD model in the pool of models and its corresponding ground truth labels. This refers to the ground truth-based portion of the reward. It should be mentioned that the same 20% of ground truth labels used during the training of the TSFs, are used here in the reward function.

The reward function of equation 6 is used as the original reward, where a TP is given a higher reward compared to a TN reward ($r_{TP} > r_{TN}$). A TP is an instance where the agent selects an AD model that correctly predicts an anomaly. However, TN means that the AD model selected by the agent at this time step correctly predicts this data point as normal. Since the majority of instances are usually normal while the anomalous instances are rare and we are interested in detecting anomalies, thus the reward for true positive should be set to be greater than the TN reward $r_{TP} > r_{TN}$. On the other hand, a higher penalty should be applied to FN compared to a FP. A FN represents a case where the RL agent selects an AD model that predicted this data point to be normal when this point is actually anomalous. A FP refers to a case where the agent chooses an AD model that predicts this instance as anomalous when it is, in reality, normal. In real-world applications, neglecting anomalies (FN case) is more detrimental than giving false alarms (FP case), so $|r_{FN}| > |r_{FP}|$. As such, we reflected it in equation 6 a reward function that consists of r_{TN} that is set to 0.5, r_{TP} is set to 1, r_{FN} is set to -1.5, and r_{FN} is set to -3.

$$R = \begin{cases} r_{TN} = 0.5 & \text{TN} \\ r_{FN} = -3 & \text{FN} \\ r_{TP} = 1 & \text{TP} \\ r_{FP} = -1.5 & \text{FP} \end{cases} \tag{6}$$

4 Dataset Description and Experimental Settings

4.1 Datasets Description

In this work, we considered two datasets; one real and one synthetic dataset. Each of these datasets consists of the hourly time-steps, the electricity consumption, and the ground truth labels. The real dataset is an annotated subset of the Large-scale Energy AD dataset (LEAD1.0) [62], and it is made up of hourly meter readings obtained from 1,413 smart electricity meters over one year and gathered from sixteen distinct sites around the world. The LEAD1.0 dataset contains point and sequential (collective) anomaly types. For the sake of this work, we used a subset of LEAD1.0 containing hourly meter readings from twenty-two smart electricity meters. We split this real data into a normal dataset consisting of only normal instances and an anomalous dataset with normal and anomalous instances. The normal dataset consisting of hourly electricity meter readings (electricity consumptions) has a total of 18696 time-steps. How-

ever, the anomalous hourly electricity meter readings dataset has readings of 20424 time-steps with around 5.7% of anomalies and their corresponding ground truth labels.

The second dataset considered in this work is a synthetic dataset in which anomalies are injected randomly into the dataset. We used the time series of hourly electricity consumption data from the residential areas of Germany provided by ENTSO-E Transparency, a European electricity data platform [63]. Similarly, this dataset was divided into normal and anomalous datasets. The normal dataset has hourly power-consumption data from the year 2015 until 2018 making up a total of 35064 time-steps. The anomalous dataset consists of 1008 hourly power-consumption data points of 2019 with 5% of randomly injected anomalies and the corresponding ground truth labels.

4.2 Data Preprocessing

Initially, data preprocessing was performed on the datasets. First, any missing values in the data were removed. Then, the electricity consumption data points in the normal and anomalous datasets (synthetic and real datasets) were rescaled. In the synthetic normal and anomalous datasets, we applied min-max scaling to scale the electricity consumption data into values between zero and one. However, in the real datasets, we applied logarithmic scaling similar to the original paper of LEAD1.0 [62]. After that, each normal and anomalous dataset was transformed into sliding windows of size six with a step size of one. The chosen size of the sliding window and the step size were found the most suitable after heuristically trying sliding sizes six, twelve, and twenty-four with different step sizes of one, two, three, and zero overlaps.

4.3 Settings of the AD Candidate Models in the Pool

We considered six different AD models in our pool of models, which are KNN [32], COPOD [43], ECOD [42], OSVM [40], IForest [46], and USAD [53]. OSVM and IForest are implemented from the Scikit-Learn library [64]. COPOD, ECOD, and KNN are imported from the PyOD toolbox [65]. As for the USAD model, it was implemented from the GitHub repository of USAD [66].

For each of KNN, OSVM, and IForest, we performed hyperparameter optimization using Hyperopt library [67] to tune their hyperparameters and select the best configuration of hyperparameters. The list of hyperparameter choices for each of KNN, OSVM, USAD, and IForest AD models are available in the Appendix. COPOD and ECOD do not require hyperparameter tuning since they are deterministic and have no hyperparameters. As for USAD, it has two hyperparameters, alpha and beta, such that their sum must be equal to one. They are utilized for parameterizing the trade-off between true and false positives, which varies the model's sensitivity. We used the original implementation of USAD having neutral detection sensitivity where alpha equals beta equals 0.5 to avoid having a sensitive model biased towards true positives or false positives.

A specific criterion is necessary to compute the empirical threshold for each AD model. Given that we know the percentage of anomalies in both datasets, we set the criterion to be equal to this percentage (5%).

To elaborate further, for each AD model, if a data point's predicted anomaly score ranks among the highest 5%, within all the predicted anomaly scores generated by that model, the data point is classified as an anomaly.

4.4 Settings of the Proposed RL Framework for AD Model Selection

In the implementation of the RL framework, we used the DQN model imported from Stable-Baseline3. Stable-Baseline3 is a RL toolbox based on PyTorch [68]. The default settings of DQN provided by [68], having a decaying exploration epsilon, were utilized. However, the exploration fraction of the decaying exploration epsilon was set to 0.7 instead of the default value of 0.1 to allow the DQN agent to explore more, and the seed number was set to 1. In the synthetic anomalous dataset case, the RL agent was trained for 600,000 time steps. The agent was trained for 3,000,000 time steps in the real anomalous dataset case. This is because the real dataset is larger than the synthetic one. During the evaluation phase of the DQN agent, we set the agent's actions to be deterministic. The exploration epsilon is set to be a decaying exploration epsilon over 600,000 time steps.

5 Experiments and Results

5.1 AD Using the Candidate AD Models in the Pool

To perform AD using the six unsupervised AD techniques, each AD model was trained on the normal sliding windows having normal electricity consumption instances. After that, each of these pretrained AD models was tested on the sliding windows of the anomalous electricity consumption instances. The predicted anomaly label (anomalous or normal instance) was compared to the ground truth labels.

The results of the unsupervised AD models on the synthetic and real datasets are presented in Tables 1 and 2, respectively. In the synthetic data, KNN achieved the highest performance having precision, recall, and F1score of 1. USAD and IForest models also achieved high performance, recording a score of 0.925 and 0.902, respectively in all the evaluation metrics. OSVM scored a lower performance having a value of 0.624 in all metrics. COPOD and ECOD yielded the lowest values of 0.361 and 0.414 in the evaluation metrics.

Table 1. Performance of AD Candidate models on the synthetic anomalous dataset.

AD Model	Precision	Recall	F1score
COPOD	0.361	0.361	0.361
ECOD	0.414	0.414	0.414
KNN	1	1	1
OSVM	0.624	0.624	0.624
IForest	0.902	0.902	0.902
USAD	0.925	0.925	0.925
Proposed	**1**	**0.977**	**0.989**

Table 2. Performance of AD Candidate models on the real anomalous dataset.

AD Model	Precision	Recall	F1score
COPOD	0.532	0.532	0.532
ECOD	0.532	0.532	0.532
KNN	0.702	0.702	0.702
OSVM	0.003	0.003	0.003
IForest	0.579	0.579	0.579
USAD	0.003	0.003	0.003
Proposed	**0.977**	**0.579**	**0.727**

As shown in Table 2 for the real data, KNN also recorded the highest of 0.702 in all the evaluation metrics. The results showed that the KNN model outperformed the other AD models' performance in both datasets. IForest model came next with a 0.579 precision, recall and F1score. Following this, COPOD and ECOD models recorded a value of 0.532 in all metrics. However, OSVM and USAD both failed to detect anomalies.

Thus, KNN seemed to preserve the highest performance in both datasets, however, it recorded lower results in the real data. The remaining AD models showed a different performance on different datasets. These results further illustrate the need for a dynamic AD model selection that can guarantee accurate AD in time series applications, especially with the absence of large quantities of ground truth labels in real-world applications.

5.2 Time Series Classification Using TSF

To avoid relying on large quantities of ground truth labels that are not usually available in real-world applications, TSFs were used to classify the predicted anomaly labels into correct or wrong labels. Six TSFs were implemented corresponding to the six AD models. Each TSF was fed with a different dataset corresponding to the AD model.

Table 3. Results of each of the TSFs on their corresponding testing set having anomalous sliding windows from the synthetic dataset.

TSF based on predictions from	Precision	Recall	F1score
COPOD	0.99	0.993	0.991
ECOD	0.983	0.997	0.99
KNN	-	-	-
OSVM	0.989	0.996	0.993
IForest	0.9738	1	0.987
USAD	0.998	0.996	0.997

The results of each of the TSFs on their corresponding testing set are shown in Tables 3 and 4 using anomalous sliding windows from the synthetic and real datasets respectively. It is worth mentioning that since KNN when tested on the synthetic anomalous data achieved a superior performance of 1, as shown in Table 1, in all precision, recall, and F1score metrics, then no need to apply a TSF to

classify the predictions of KNN. Table 3 illustrates that all TSFs achieved remarkable performance of F1scores ranging between 0.987 and 0.997 in the case of anomalous synthetic instances. Similarly, in Table 4, in the case of the anomalous real dataset, they recorded F1scores ranging between 0.913 and 0.979.

Table 4. Results of each of the TSFs on their corresponding testing set having anomalous sliding windows from the real dataset.

TSF based on predictions from	Precision	Recall	F1score
COPOD	0.978	0.956	0.9667
ECOD	0.978	0.962	0.969
KNN	0.977	0.975	0.976
OSVM	0.974	0.86	0.913
IForest	0.971	0.986	0.979
USAD	0.975	0.856	0.911

5.3 AD Model Selection Based on an RL Framework

In this work, we proposed an AD model selection using a RL framework. The original reward function, shown in equation 6, was used in our proposed RLAD model selection framework. The performance of the proposed model selection framework for AD the synthetic dataset is stated in Table 1. Comparing the results of the six AD models in Table 1; shows that the proposed AD model selection provided a precision of 1, the same as KNN outperforming all the remaining AD models. In terms of the F1score, our suggested AD model selection recorded a value of 0.989 higher than the entire candidate AD models except for KNN having a slightly greater F1score of 1. In the case of the real dataset, results in Table 2 show that the proposed AD model selection, having a precision of 0.977 and an F1score of 0.727, outperformed all the candidate AD techniques of Table in terms of the precision and F1score.

5.4 Comparison of AD Model Selection Based on an RL Framework to an LLM-based AD model

Considering the synthetic dataset, which is a relatively small dataset, the proposed RLAD model selection framework was compared to an LLM-based AD model. Leveraging Chain-of-Thought (CoT) prompting, as illustrated in Figure 2, we motivated the ChatGPT4o-mini model to perform an AD task and to break down the complex thoughts or ”processes” into smaller steps before providing a response on any detected anomalies. The LLM model yielded encouraging results on the AD task scoring an F1 score of 0.913 and a recall of 0.84 as shown in Table 5. Yet, our proposed AD model framework, recording an F1 score of 0.989 and recall of 0.977 still outperforms the LLM-based AD model.

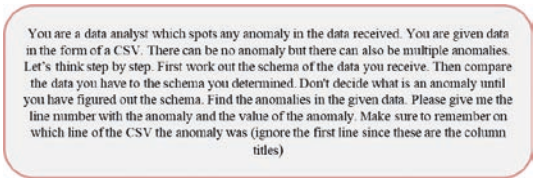


Figure 2. Template for the prompt strategy for the LLM to perform an AD task

Table 5. Performance of the LLM-based AD model on the synthetic dataset in comparison of the proposed RLAD model selection framework.

AD Model	Precision	Recall	F1score
LLM	1	0.83	0.913
Proposed	**1**	**0.977**	**0.989**

5.5 Comparison of Different RL Frameworks for AD Model Selection

The proposed RLAD model selection framework, which uses a small portion of ground truth labels, was compared with two other RLAD model selection frameworks; The first framework RLAD_{Gtruth} totally relies on ground truth labels in its reward function. This means that, in all the instances, the reward was obtained by comparing the predicted anomaly label with the ground truth label. On the contrary, the second framework named RLAD_{Class} relies totally on the predicted class labels obtained from TSFs. It should be noted that the RLAD_{Class} framework also requires a portion of ground truth labels (used for TSFs training). The state-space and action-space remain the same in all frameworks. The original reward of equation 6 was applied in all of the three frameworks (r_{TN}= 0.5, r_{FN}= -3, r_{TP}= 1, r_{FP}= -1.5).

The precision, recall, and F1score results of these two AD model selection frameworks based on RL are also present in Tables 6 and 7 using the synthetic and real datasets respectively. In both datasets, all the frameworks showed nearly the same high performance. In the synthetic data, the three frameworks recorded a precision of 1; however, our proposed framework recorded the highest F1score of 0.989. In the real data, our proposed framework, having a 0.977 precision and an F1score of 0.727, showed a slightly higher performance than RLAD_{Class} but slightly lower than RLAD_{Gtruth}. It can be said that, in the case of the real dataset, the proposed AD model selection offered a compromise between RLAD_{Gtruth} that explicitly uses ground truth labels and RLAD_{Class} that relies totally on class labels in its reward function.

Table 6. Results of the three different RL Model Selection frameworks for AD on the anomalous synthetic data.

Synthetic Dataset			
AD Model	**Precision**	**Recall**	**F1score**
Proposed	**1**	**0.977**	**0.989**
RLAD_{Gtruth}	1	0.955	0.977
RLAD_{Class}	1	0.925	0.961

Table 7. Results of the Performance of the three different RL Model Selection frameworks for AD on the anomalous real data.

Real Dataset			
AD Model	**Precision**	**Recall**	**F1score**
Proposed	0.977	0.579	0.727
RLAD_{Gtruth}	0.971	**0.67**	**0.793**
RLAD_{Class}	**0.979**	0.549	0.703

5.6 Different Constant Reward Functions with a Decaying Epsilon

To study the effect of the different reward values on the proposed RL framework, we considered two extreme scenarios of reward functions R1 and R2 (Table 8) and evaluated them on the synthetic dataset. It can be observed that in comparison with the original reward function (r_{TN}= 0.5, r_{FN}= -3, r_{TP}= 1, r_{FP}= -1.5), the TN reward in R1 was decreased from 0.5 to 0.15. The FP reward in the original reward which represented a high penalty of -3 was changed into a low positive reward of 0.1. However, the FN reward and TP rewards in both functions remained the same (r_{FN}= -3 and r_{TP}= 1). Applying a lower positive reward on TN and a low positive reward on FP instead of a high penalty has led to a decrease in the model's performance in the case of R1. In this regard, the precision decreased sharply from 1 to 0.636, the recall decreased from 0.977 to 0.88, and the F1score also decreased from 0.989 to 0.738 as shown in Table 9.

As shown in Table 8, another extreme scenario was investigated in the R2 function; in which the TN reward in R2 was set to be greater than the TN reward in the original function. Similarly, the FP penalty in R2 was set to be greater than the FP penalty in the original function. However, the penalty applied on FN in the original reward function was modified to a low reward in the case of R2. The TP reward in R2 was set to be less than the TP reward in the original function. The results of the original reward function in comparison to R2 reward function, in Table 9, reflected a slight decrease in the precision from 1 to 0.987 but a significant decrease in the recall from 0.977 to 0.564, and in the F1score from 0.989 to 0.718. This is due to decreasing the TP reward and giving the agent a low reward in case of a FN instead of penalizing it with a negative reward. As a result, the reward functions play a significant role in the performance of the proposed AD model selection approach in which the original reward function offered the most suitable and logical formulation of the AD model selection problem.

5.7 Adaptive Rewards with Different Epsilons

We also inspected the effect of two adaptive reward functions; an increasing adaptive reward AdapInc and a decreasing adaptive reward AdapDec. In the AdapInc, we designed a reward function that increases iteratively with the time steps of the anomalous data. In this regard, the TN and TP rewards were intended to increase in a step-wise parabolic manner as a function of the time steps (or length) of the anomalous data. Similarly, the FN and FP penalties were set to rise in a step-wise parabolic form as a function of the time steps of the anomalous data. An illustration of the

AdapInc reward is demonstrated in Figure 3. The AdapInc reward is formulated in the equation below:

$$R_{\text{AdapInc}} = \begin{cases} r_{\text{TN}} = 5 \times 10^{-2} \times C^2 & \text{TN} \\ r_{\text{FN}} = -3 \times 10^{-2} \times C^2 & \text{FN} \\ r_{\text{TP}} = 1 \times 10^{-2} \times C^2 & \text{TP} \\ r_{\text{FP}} = -1.5 \times 10^{-2} \times C^2 & \text{FP} \end{cases} \quad (7)$$

In both adaptive rewards, the counter C was initially set to 1, for every 100 time-steps the counter would be incremented by 1. When reaching the final time-step (final sliding window of the anomalous dataset), C would be reset to 1. Alternatively, as shown in Figure 4, the reward in the AdapDec is designed to decrease as a function of the time-steps of the anomalous data. Towards this, the TN and TP rewards would decrease iteratively step-wise hyperbolically. Also, the FN and FP penalties would decline in a step-wise hyperbolic manner as a function of the time steps (or length) of the anomalous data. The formulation of the AdapDec reward is stated in the equation below:

$$R_{\text{AdapDec}} = \begin{cases} r_{TN} = \frac{0.5}{C^2} & \text{TN} \\ r_{FN} = \frac{-3}{C^2} & \text{FN} \\ r_{TP} = \frac{1}{C^2} & \text{TP} \\ r_{FP} = \frac{-1.5}{C^2} & \text{FP} \end{cases} \quad (8)$$

5.7.1 Adaptive Reward Functions with a Decaying Epsilon

In this experiment, the same decaying exploration epsilon of Figure 2 was used. The proposed model selection AD framework was evaluated on the two adaptive rewards. The AdapInc reward, having an increasing reward function, encourages the agent to explore different actions at the beginning of the iterations (time-steps of the data) which gives the agent a higher chance to select randomly an AD model at the beginning. Then later on, on higher iterations, the agent is invited to greedily select a model that would yield the largest reward. Hence, eventually, the agent would be able to learn the optimal policy of selection. This is reflected in the results of the proposed AD model selection. The original and AdapInc rewards in Table 9 both had a precision of 1. The F1score of AdapInc reward is 0.981, slightly lower than that of the original reward with an F1score of 0.989. On the other hand, the AdapDec reward, representing a decreasing reward function, prevents the agent from exploring various actions at the beginning and forces the agent at the beginning to select greedily an action (an AD model) that gives him a high reward which does not allow the agent to learn the optimal policy of selection. This is proven in the results of the proposed AD model selection in Table 10 with AdapDec having a precision of 0.763 and an F1score of 0.833. Given that the decaying exploration epsilon, which already allows the agent to explore more in the earlier time steps of the RL training phase, was applied in all cases, both the original and the AdapInc rewards achieved very similar high performance as shown in Table 10. On the contrary, the AdapDec, which prevents exploration in the earlier time instances, recorded a deteriorating performance.

5.7.2 Adaptive Reward Function with a Constant Exploration Epsilon

Considering the proposed model selection framework, we also evaluated the same original reward, adaptive increasing reward AdapInc, and adaptive decreasing reward AdapDec but instead of using a decaying exploration epsilon, a constant exploration epsilon was applied. We considered three various cases; an exploration epsilon equals 0.9, 0.5, and 0.05. As stated in Table 10, in case of a high epsilon value of 0.9, all three rewards showed high performance where the AdapInc reward recorded the highest evaluation metrics with an F1score of 0.974, followed by the original reward having an F1score of 0.933, and finally the AdapDec reward with an F1score of 0.925. Upon setting the epsilon to 0.5, the original reward and the AdapInc reward both revealed good performance in which the original reward showed the best results opposing that of an AdapDec reward which exhibited a degrading performance. The F1score of the original, AdapInc, and AdapDec rewards, as shown in Table 9, were 0.953, 0.938, and 0.743 respectively. In the case of a very low epsilon of 0.05, the original reward produced the highest F1score of 0.942, while the AdapDec reward yielded the lowest F1score of 0.826. As for the AdapInc reward, it resulted in an F1score of 0.92. As a result in all three cases, both the original and the AdapInc rewards showed better results. However, the original

reward had the highest score in the cases of 0.5 and 0.05. This implies that, in the case of low and intermediate exploration epsilons, a constant reward yielded the best performance. Whereas, in the case of a high exploration epsilon (epsilon= 0.9), the AdapInc reward gave the highest scores. All in all, the original reward and the AdapInc reward functions maintained both higher results in the different studied cases. However, the original reward using a decaying exploration epsilon produced the best overall performance.

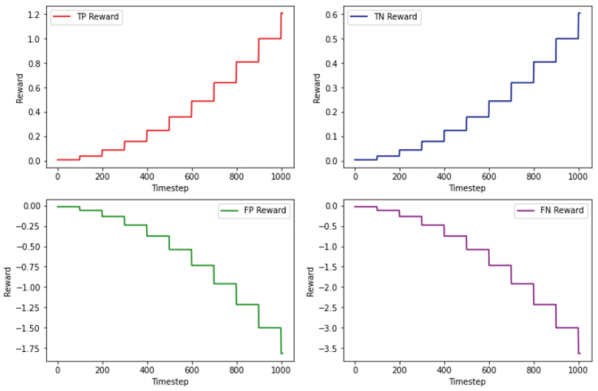


Figure 3. Adaptive Increasing Reward

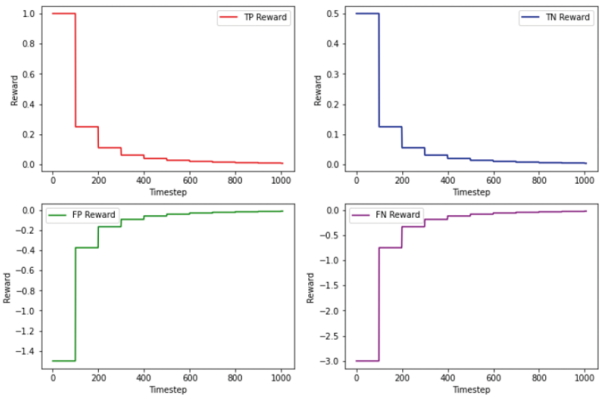


Figure 4. Adaptive Decreasing Reward

Table 8. Different Constant Reward Functions on the Proposed RL Framework for AD Model Selection.

Reward Function				
Reward	**TN**	**FN**	**TP**	**FP**
Original	**0.5**	**-3**	**1**	**-1.5**
R1	0.15	-3	1	0.1
R2	1	0.1	0.15	-3

Table 9. Results of the Performance of the Proposed RL Framework for AD Model Selection on Different Constant Reward Functions Using the Synthetic Dataset.

Decaying Epsilon			
Reward	**Precision**	**Recall**	**F1score**
Original	**1**	**0.977**	**0.989**
R1	0.636	0.88	0.738
R2	0.987	0.564	0.718

5.8 Different Types of Anomalies

In time series data, there exist different types of anomalies, consisting of global anomalies, local anomalies, and clustered anomalies.

Global anomalies: point anomalies that seem to differ from the overall time series.

Local anomalies: point anomalies that deviate from their neighborhood data points. A point anomaly takes place at any time but does not repeat.

Clustered anomalies: represent a group of anomalies that show similar characteristics. Clustered or sequential anomalies may be present once or repeatedly. Clustered anomalies can be local or global anomalies [57].

Usually, the available real-world datasets have a mixture of various anomaly types. This makes it difficult to recognize how AD techniques would behave in certain types of anomalies [16]. In this work, we attempted to use the time series of hourly electricity consumption data from the residential areas of France also provided by ENTSO-E Transparency [58]. The hourly power-consumption data from the year 2015 until 2017 having a total of 26302 time-steps is considered for the normal data. The remaining 8016 time-steps of the year 2018 will be used for the anomalous dataset. To evaluate the effect of different anomaly types, we created three separate anomalous datasets; global, local, and clustered, anomalous datasets, in which we injected a quantity of 2% anomalies to the hourly power-consumptions of 2018 of global, local, and clustered types respectively. We applied the same preprocessing procedure that was performed on the above synthetic dataset of Germany and then trained the six unsupervised AD techniques on the normal sliding windows and evaluated their performance on each of the three anomalous sliding win-

Table 10. Results of the Performance of the Proposed RL Framework for AD Model Selection on Different Reward Functions Using the Synthetic Dataset.

Reward Function	Epsilon	Precision	Recall	F1score
Original	Decaying	**1**	**0.977**	**0.989**
AdapInc		1	0.962	0.981
AdapDec		0.763	0.917	0.833
Original	Constant=0.9	0.983	0.887	0.933
AdapInc		0.977	0.97	0.974
AdapDec		0.975	0.88	0.925
Original	Constant=0.5	1	0.91	0.953
AdapInc		0.976	0.902	0.938
AdapDec		0.675	0.827	0.743
Original	Constant=0.05	0.968	0.917	0.942
AdapInc		0.944	0.895	0.92
AdapDec		0.784	0.872	0.826

dows. The precision, recall, and F1score of the AD models on each of the global, local, and clustered anomalous datasets in Tables 14 shows that the performance of AD models differed depending on the type of anomalies. Except for KNN and IForest, all AD models showed low performance in both datasets with local and global anomalies and higher performance in the dataset with clustered anomalies. KNN and IForest, on the other hand, maintained a high performance in all cases recording the highest performance in the global dataset and the lowest in the clustered dataset. To apply the proposed AD model selection on each of the three global, local, and clustered anomalous datasets, the same process of implementing six TSFs was performed to classify the predicted anomaly labels of each AD model into correct labels or wrong labels in an attempt to avoid using large quantities of ground truth labels. The results of the performance of each of the TSFs on the three different anomalous testing sets are provided in Tables 11, 12, and 13 where each TSF showed high-performance scores. After that, using the anomalous sliding windows, the predicted anomaly scores from the AD models, the obtained empirical thresholds of the AD models, and their predicted anomaly labels, as well as the two confidence scores, the proposed AD model selection framework was applied for each of the three distinct anomalous datasets. Table 14 showed the precision, recall, and F1score on each of the three different types of anomalous datasets. The proposed model selection framework achieved an F1score of 0.962, 0.931, and 0.963 in the global, local, and clustered datasets respectively outperforming the performance of the whole AD models except for KNN. KNN yielded a slightly higher performance with F1score of 1, 0.99, and 0.965. The reason behind this is that in our proposed AD model selection framework, the reward function mostly relies on the predicted class labels of the TSF instead of explicitly depending on ground truth labels. And although the classifications of the TSFs are considered accurate, it is not 100% accurate which would slightly decrease the performance of our proposed AD model selection in some cases. Nonetheless, in all the considered datasets, our proposed model selection approach for AD maintained high-performance results.

Table 11. Results of each of the TSFs on their corresponding testing set having global anomalous sliding windows from the France dataset.

TSF based on predictions from	Precision	Recall	F1score
COPOD	0.997	0.921	0.958
ECOD	0.946	0.99	0.967
KNN	-	-	-
OSVM	0.999	0.999	0.999
IForest	0.999	0.996	0.998
USAD	0.987	0.918	0.951

Table 12. Results of each of the TSFs on their corresponding testing set having local anomalous sliding windows from the France dataset.

TSF based on predictions from	Precision	Recall	F1score
COPOD	0.991	0.992	0.991
ECOD	0.932	0.999	0.964
KNN	0.999	1	0.999
OSVM	0.998	0.999	0.999
IForest	0.996	0.988	0.992
USAD	0.998	0.981	0.989

Table 13. Results of each of the TSFs on their corresponding testing set having clustered anomalous sliding windows from the France dataset.

TSF based on predictions from	Precision	Recall	F1score
COPOD	0.997	0.999	0.998
ECOD	0.991	1	0.995
KNN	0.999	1	0.999
OSVM	0.998	1	0.999
IForest	0.998	0.999	0.999
USAD	0.998	1	0.999

Table 14. Results of the AD Models on Different Anomaly Types.

F1score of AD Model	Global	Local	Clustered
COPOD	0.283	0.088	0.611
ECOD	0.241	0.117	0.594
KNN	1	0.99	0.965
OSVM	0.093	0.016	0.611
IForest	0.922	0.865	0.852
USAD	0.286	0.105	0.747
Proposed	**0.962**	**0.931**	**0.963**

6 Conclusion and Future Work

In this study, we address the challenges posed by the complex and diverse nature of data abnormalities in real-world systems, coupled with the limited availability of large quantities of ground truth labels. To tackle this, we propose a model selection framework for AD that combines RL with TSF. Our approach incorporates partial reliance on ground truth labels in the reward function and dynamically selects an appropriate AD model at each time step.

The results of our proposed approach are remarkable. It surpasses all other AD models in terms of performance on real-world data and achieves outstanding results on synthetic data, outperforming almost all other models except for KNN, which attains an F1 score of 1. It also exceeded the performance the LLM-based AD model on the synthetic dataset.

The proposed AD model selection approach was examined using different reward functions. The original and AdapInc rewards functions exhibited superior results in both constant and decaying exploration epsilon scenarios. The original reward function with a decaying exploration epsilon demonstrated the best precision, recall, and F1 score outcomes.

The impact of different types of anomalies was investigated, revealing variations in the performance of each AD model. This finding further emphasizes the significance of the proposed model selection framework for unsupervised AD, enabling accurate detection of anomalies in time series data without an extensive availability of ground truth labels.

Future endeavors will involve applying the proposed AD model selection framework to larger time series datasets with higher dimensions. Additionally, incorporating additional semi-supervised AD models into the pool of candidate models is an avenue for further exploration. In this regard, we plan to incorporate additional semi-supervised AD techniques. Given the promising results of LLMs in AD tasks, we are also considering the integration of an LLM-based AD model into the pool of candidate models. By requiring only a small portion of ground truth labels and harnessing the strengths of multiple AD models, our approach raises its applicability to real-world scenarios with diverse anomaly types.

Acknowledgement

This work was supported by the Maroun Semaan Faculty of Engineering and Architecture and

the University Research Board at the American University of Beirut, Lebanon.

A Hyperparameter Choices

Table 15. Hyperparameter Choices.

Hyperparameter 1	Hyperparameter 2
n_neighbors: [1, 5, 10, 15, 20, 25, 50, 60, 70, 80, 90, 100]	method: ['largest', 'mean', 'median']
nu (train error tol): [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]	-
n_estimators: [10, 20, 30, 40, 50, 75, 100, 150, 200]	max_features: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]

References

[1] J. E. Zhang, D. Wu, and B. Boulet, Time Series Anomaly Detection via Reinforcement Learning-Based Model Selection, 2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2022, 193–199.

[2] A.N. Alkuwari, S. Al-Kuwari, M. Qaraqe, Anomaly Detection in Smart Grids: A Survey from Cybersecurity Perspective, 2022 3rd International Conference on Smart Grid and Renewable Energy (SGRE), 2022, 1-7.

[3] A. Chatterjee, B.S. Ahmed, IoT Anomaly Detection Methods and Applications: A Survey, Internet of Things, 19, 2022, 100568.

[4] A.A. Khalil, F. E Ibrahim, M.Y. Abbass, N. Haggag, Y. Mahrous, A. Sedik, Z. Elsherbeeny, A.M. Khalaf, M. Rihan, W. El-Shafai, Efficient Anomaly Detection from Medical Signals and Images with Convolutional Neural Networks for Internet of Medical Things (IoMT) Systems, International Journal for Numerical Methods in Biomedical Engineering, 38(1), 2022, e3530.

[5] W. Hilal, S.A. Gadsden, J. Yawney, Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances, Expert Systems with Applications, 193, 2022, 116429.

[6] M. U. Hassan, M. H. Rehmani, and J. Chen, Anomaly Detection in Blockchain Networks: A Comprehensive Survey, IEEE Communications Surveys & Tutorials, 25(1), 2022, 289–318.

[7] A. Singh and K. Chatterjee, Cloud Security Issues and Challenges: A Survey, Journal of Network and Computer Applications, 79, 2017, 88–115.

[8] D. Jung, N. Ramanan, M. Amjadi, S. R. Karingula, J. Taylor, and C. N. Coelho Jr, Time Series Anomaly Detection with Label-Free Model Selection, arXiv preprint arXiv:2106.07473, 2021.

[9] V. Barnett and T. Lewis, Outliers in Statistical Data, 3rd ed., Wiley, New York 1994.

[10] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, A Unifying Review of Deep and Shallow Anomaly Detection, Proceedings of the IEEE, 109(5), 2021, 756–795.

[11] M. Gunduz and A. M. A. Yahya, Analysis of Project Success Factors in Construction Industry, Technological and Economic Development of Economy, 24(1), 2018, 67–80.

[12] P. M. Tehrani, Cyber Resilience Strategy and Attribution in the Context of International Law, European Conference on Cyber Warfare and Security, 2019, 501–XVI.

[13] J. Ghanim, M. Issa, and M. Awad, An Asymmetric Loss with Anomaly Detection LSTM Framework for Power Consumption Prediction, 2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON), 2022, 819–824.

[14] N. B. Aissa and M. Guerroumi, Semi-Supervised Statistical Approach for Network Anomaly Detection, Procedia Computer Science, 83, 2016, 1090–1095.

[15] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, Ganomaly: Semi-Supervised Anomaly Detection via Adversarial Training, Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14, 2019, 622–637.

[16] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, Adbench: Anomaly Detection Benchmark, Advances in Neural Information Processing Systems, 35, 2022, 32142–32159.

[17] V. Chandola, A. Banerjee, and V. Kumar, Anomaly Detection: A Survey, ACM Computing Surveys (CSUR), 41(3), 2009, 1–58.

[18] J. P. S. Chhabra and G. P. Warn, A Method for Model Selection Using Reinforcement Learning When Viewing Design as a Sequential Decision Process, Structural and Multidisciplinary Optimization, 59, 2019, 1521–1542.

[19] R.S. Sutton and A.G. Barto, Reinforcement Learning, second edition: An Introduction, MIT Press, 2018.

[20] V. Kosana, K. Teeparthi, S. Madasthu, and S. Kumar, A Novel Reinforced Online Model Selection Using Q-learning Technique for Wind Speed Prediction, Sustainable Energy Technologies and Assessments, 49, 2022, 101780.

[21] Y. Fu, D. Wu, and B. Boulet, Reinforcement Learning Based Dynamic Model Combination for Time Series Forecasting, Proceedings of the AAAI Conference on Artificial Intelligence, 36(6), 2022, 6639–6647.

[22] K. Christophe, J. El Zini, and M. Awad, A Progressive and Cross-Domain Deep Transfer Learning Framework for Wrist Fracture Detection, Journal of Artificial Intelligence and Soft Computing Research, 12(2), 2021, 101-120.

[23] H. Deng, G. Runger, E. Tuv, and V. Martyanov, A Time Series Forest for Classification and Feature Extraction, Information Sciences, 239, 2013, 142–153.

[24] Lim, Bryan, Zohren, and Stefan, Time-Series Forecasting with Deep Learning: A Survey. Philosophical Transactions of the Royal Society A, 379(2194), 2021, 20200209.

[25] M. Braei and S. Wagner, Anomaly Detection in Univariate Time-Series: A Survey on the State-of-the-Art, arXiv preprint arXiv:2004.00433, 2020.

[26] C.C. Aggarwal, Outlier Analysis, Springer International Publishing, 2016.

[27] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, Toward Supervised Anomaly Detection, Journal of Artificial Intelligence Research, 46, 2013, 235–262.

[28] V. N. Vapnik, An Overview of Statistical Learning Theory, IEEE Transactions on Neural Networks, 10(5), 1999, 988–999.

[29] V. Sindhwani, P. Niyogi, and M. Belkin, Beyond the Point Cloud: From Transductive to Semi-Supervised Learning, Proceedings of the 22nd International Conference on Machine Learning, 2005, 824–831.

[30] E. M. Knorr, R. T. Ng, and V. Tucakov, Distance-Based Outliers: Algorithms and Applications, The VLDB Journal, 8(3), 2000, 237–253.

[31] S. Ramaswamy, R. Rastogi, and K. Shim, Efficient Algorithms for Mining Outliers from Large Data Sets, Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, 427–438.

[32] F. Angiulli and C. Pizzuti, Fast Outlier Detection in High Dimensional Spaces, European Conference on Principles of Data Mining and Knowledge Discovery, 2002, 15–27.

[33] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, LOF: Identifying Density-Based Local Outliers, Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, 93–104.

[34] G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, On the Evaluation of Unsupervised Outlier Detection: Measures, Datasets, and an Empirical Study, Data Mining and Knowledge Discovery, 30, 2016, 891–927.

[35] L. Xiong, X. Chen, and J. Schneider, Direct Robust Matrix Factorization for Anomaly Detection, 2011 IEEE 11th International Conference on Data Mining, 2011, 844–853.

[36] L. Li, J. McCann, N. S. Pollard, and C. Faloutsos, Dynammo: Mining and Summarization of Coevolving Sequences with Missing Values, Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, 507–516.

[37] N. Görnitz, M. Braun, and M. Kloft, Hidden Markov Anomaly Detection, International Conference on Machine Learning, 2015, 1833–1842.

[38] A. P. Dawid and A. M. Skene, Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm, Journal of the Royal Statistical Society: Series C (Applied Statistics), 28(1), 1979, 20–28.

[39] M. Awad, and R. Khanna, Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, Springer Nature, 2015.

[40] L. M. Manevitz and M. Yousef, One-Class SVMs for Document Classification, Journal of Machine Learning Research, 2(Dec), 2001, 139–154.

[41] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, Estimating the Support of a High-Dimensional Distribution, Neural Computation, 13(7), 2001, 1443–1471.

[42] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. H. Chen, ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions, IEEE Transactions on Knowledge and Data Engineering, 35(12), 2022, 12181–12193.

[43] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, COPOD: Copula-Based Outlier Detection, 2020 IEEE International Conference on Data Mining (ICDM), 2020, 1118–1123.

[44] A. Kharitonov, A. Nahhas, M. Pohl, and K. Turowski, Comparative Analysis of Machine Learning Models for Anomaly Detection in Manufacturing, Procedia Computer Science, 200, 2022, 1288–1297.

[45] Z. Xu, D. Kakde, and A. Chaudhuri, Automatic Hyperparameter Tuning Method for Local Outlier Factor, with Applications to Anomaly Detection, 2019 IEEE International Conference on Big Data (Big Data), 2019, 4201–4207.

[46] F. T. Liu, K. M. Ting, and Z.-H. Zhou, Isolation Forest, 2008 Eighth IEEE International Conference on Data Mining, 2008, 413–422.

[47] P. J. Rousseeuw and M. Hubert, Anomaly Detection by Robust Statistics, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(2), 2018, e1236.

[48] H. Hoffmann, Kernel PCA for Novelty Detection, Pattern Recognition, 40(3), 2007, 863–874.

[49] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, LSTM-Based Encoder-Decoder for Multi-Sensor Anomaly Detection, arXiv preprint arXiv:1607.00148, 2016.

[50] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, 2828–2837.

[51] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, Neural Computation, 9(8), 1997, 1735–1780.

[52] J. El Zini, Y. Rizk, and M. Awad, An Optimized Parallel Implementation of Non-Iteratively Trained Recurrent Neural Networks, Journal of Artificial Intelligence and Soft Computing Research, 11(1), 2021, 33-50.

[53] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, USAD: Unsupervised Anomaly Detection on Multivariate Time Series, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, 3395–3404.

[54] S. González-Carvajal and E. C. Garrido-Merchán, Comparing BERT Against Traditional Machine Learning Text Classification, arXiv preprint arXiv:2005.13012, 2020.

[55] W. Dang, B. Zhou, W. Zhang, and S. Hu, Time Series Anomaly Detection Based on Language Model, Proceedings of the Eleventh ACM International Conference on Future Energy Systems, 2020, 544–547.

[56] M. Dong, H. Huang, and L. Cao, Can LLMs Serve As Time Series Anomaly Detectors?, arXiv preprint arXiv:2408.03475, 2024.

[57] J. Su, C. Jiang, X. Jin, Y. Qiao, T. Xiao, H. Ma, R. Wei, Z. Jing, J. Xu, and J. Lin, Large Language Models for Forecasting and Anomaly Detection: A Systematic Literature Review, arXiv preprint arXiv:2402.10350, 2024.

[58] Y. Li, Z. Chen, D. Zha, K. Zhou, H. Jin, H. Chen, and X. Hu, AUTOOD: Neural Architecture Search for Outlier Detection, 2021 IEEE 37th International Conference on Data Engineering (ICDE), 2021, 2117–2122.

[59] K.-H. Lai, D. Zha, G. Wang, J. Xu, Y. Zhao, D. Kumar, Y. Chen, P. Zumkhawaka, M. Wan, D. Martinez, et al., TODS: An Automated Time Series Outlier Detection System, Proceedings of the AAAI Conference on Artificial Intelligence, 35(18), 2021, 16060–16062.

[60] Y. Li, D. Zha, P. Venugopal, N. Zou, and X. Hu, PYODDS: An End-to-End Outlier Detection System with Automated Machine Learning, Companion Proceedings of the Web Conference 2020, 2020, 153–157.

[61] Y. Zhao, R. Rossi, and L. Akoglu, Automatic Unsupervised Outlier Model Selection, Advances in Neural Information Processing Systems, 34, 2021, 4489–4502.

[62] M. Gulati and P. Arjunan, LEAD1.0: A Large-Scale Annotated Dataset for Energy Anomaly Detection in Commercial Buildings, Proceedings of the Thirteenth ACM International Conference on Future Energy Systems, 2022, 485–488.

[63] A Platform for Open Data of the European Power System, available at https://open-power-system-data.org/, Accessed on: Aug. 9, 2024.

[64] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-Learn: Machine Learning in Python, The Journal of Machine Learning Research, 12, 2011, 2825–2830.

[65] Y. Zhao, Z. Nasrullah, and Z. Li, PYOD: A Python Toolbox for Scalable Outlier Detection, Journal of Machine Learning Research, 20(96), 2019, 1–7.

[66] M. Galati, USAD: UnSupervised Anomaly Detection on Multivariate Time Series, available at https://github.com/manigalati/usad, 2020. Accessed on: Aug. 9, 2024.

[67] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, Hyperopt: A Python Library for

Model Selection and Hyperparameter Optimization, Computational Science & Discovery, 8(1), 2015, 014008.

[68] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, Stable-Baselines3: Reliable Reinforcement Learning Implementations, Journal of Machine Learning Research, 22(268), 2021, 1–8.



Jihan Ghanim is an AI Researcher with three years of experience in research, specializing in time-series load prediction, anomaly detection, reinforcement learning (RL), and natural language processing (NLP). She holds a Master's degree in Electrical and Computer Engineering from the American University of Beirut (AUB) and has previously worked as a research assistant at AUB, where she contributed to several AI-driven research projects. She was also a former data science researcher at Zayed University. Her research interests include developing AI methodologies for optimizing power systems and advancing AI applications in RL and NLP. She is actively engaged in exploring innovative solutions that address real-world challenges using AI techniques.
https://orcid.org/0009-0004-9797-4422



Mariette Awad is the director of the Artificial Intelligence, Data Science and Computing Hub at American University of Beirut (AUB), Lebanon, a tenured associate professor at the ECE department. Awad's academic contributions are recognized globally. She is among the top 2% of influential scholars in 2022 and 2023, by Stanford University and has earned the 2024 Abdul Hameed Shoman Award for Arab Researchers in the field of engineering and technological sciences - computational modeling in data analysis. Awad also won in 2024 the Intissar Ahdab award for Women's Excellence and Impact in Research.
She is an Authorized Assessor for ethical AI and has contributed to the drafting of the Munich Convention on AI, Data and Human Rights. She has published over 130 manuscripts. Her best seller book "Efficient Learning Machines" was one of the 5 most downloaded books by Springer Nature in Sept 2020. Springer Nature, is a leading authority in publishing textbooks and academic books for more than 175 years. BookAuthority listed her work as "#1 among the 41 Best-Selling Machine Theory books of all time" in 2020.
https://orcid.org/0000-0002-4815-6894