Reinforcement Learning-Based Adaptive Feature Boosting for Smart Grid Intrusion Detection

Chengming Hu, *Graduate Student Member, IEEE*, Jun Yan, *Member, IEEE*, and Xue Liu, *Fellow, IEEE*

***Abstract*—Intrusion detection systems (IDSs) are crucial in the security monitoring for the smart grid with increasing machine-to-machine communications and cyber threats thereafter. However, the multi-sourced, correlated, and heterogeneous smart grid data pose significant challenges to the accurate attack detection by IDSs. To improve the attack detection, this paper proposes Reinforcement Learning-based Adaptive Feature Boosting, which aims to leverage a series of AutoEncoders (AEs) to capture critical features from the multi-sourced smart grid data for the classification of normal, fault, and attack events. Multiple AEs are utilized to extract representative features from different feature sets that are automatically generated through a weighted feature sampling process; each AE-extracted feature set is then applied to build a Random Forest (RF) base classifier. In the feature sampling process, Deep Deterministic Policy Gradient (DDPG) is introduced to dynamically determine the feature sampling probability based on the classification accuracy. The critical features that improve the classification accuracy are assigned larger sampling probabilities and increasingly participate in the training of next AE. The presence of critical features is increased in the event classification over the multi-sourced smart grid data. Considering potential different alarms among base classifiers, an ensemble classifier is further built to distinguish normal, fault, and attack events. Our proposed approach is evaluated on the two realistic datasets collected from Hardware-In-the-Loop (HIL) and WUSTIL-IIOT-2021 security testbeds, respectively. The evaluation on the HIL security dataset shows that our proposed approach achieves the classification accuracy with 97.28%, an effective 5.5% increase over the vanilla Adaptive Feature Boosting. Moreover, the proposed approach not only accurately and stably selects critical features on the WUSTIL-IIOT-2021 dataset based on the significant difference of feature sampling probabilities between critical and uncritical features, i.e., the probabilities greater than 0.08 and less than 0.01, but also outperforms the other best-performing approaches with the increasing Matthew Correlation Coefficient (MCC) of 8.03%.**

***Index Terms*—Adaptive feature boosting, feature extraction, intrusion detection systems, reinforcement learning, smart grids.**

I. Introduction

A. Background

The traditional power grid is transforming into the cyber-physical smart grid with increasing two-way communications. National Institute of Standards and Technology (NIST) introduces the two-way information flow in the cyber-physical smart grid [1], as shown in Fig. 1. Sensors are devices to produce measurements of some physical properties, and then send these measurements to control centers, Intrusion Detection Systems (IDSs), asset management systems as well as historians. Actuators are devices that receive control commands generated from control centers and execute corresponding actions to the physical system. The Security Information and Event Management (SIEM) receives data streams from historians and log collectors that store measurements (e.g., voltage and current phase magnitudes) and log data (e.g., detected events from IDSs), respectively. These data are analyzed in SIEM to detect suspicious events and potential cyber incidents that are eventually reported to operators via visualization tools.

With the increasing communication and complexity of grid modernization, a surge of multi-sourced, correlated, and heterogeneous smart grid data can record various synchrophasor measurement data (e.g., current and voltage phase components) and descriptive event logs (e.g., Snort logs and control panel logs) [2], [3], [4]. For instance, IDSs deployed over an IEC-61850 [5] substation network monitor data streams from merging units, Phasor Measurement Units (PMUs), intelligent relays, bay controllers, GPS clocks, backend offices, among others. Besides, the data can be various formats from a single device. An intelligent relay equipped with a PMU currently already reports synchrophasors, GPS timestamps, relay status, system logs, and IEC 62351-7 management information bases [6] for the network and power system management.

B. Related Work

Despite efficiency and reliability promised by the cyber-physical smart grid, the growing threats and incidences of hostile attacks targeting critical power and energy infrastructures have exposed severe vulnerabilities. To detect malicious attacks in the cyber-physical smart grid, the ensemble learning-based IDSs have been generally introduced to monitor the smart grid against possible hostile attacks originating from both outside and inside the system.

Manuscript received 2 March 2022; revised 23 August 2022 and 5 December 2022; accepted 15 December 2022. Date of publication 20 December 2022; date of current version 21 June 2023. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2018-06724, and in part by the Fonds de Recherche du Québec–Nature et Technologies (FRQNT) under Grant 2019-NC-254971. Paper no. TSG-00288-2022. *(Corresponding author: Chengming Hu.)*

Chengming Hu and Xue Liu are with the School of Computer Science, McGill University, Montreal, QC H3A 0G4, Canada (e-mail: chengming.hu@mail.mcgill.ca; xueliu@cs.mcgill.ca).

Jun Yan is with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 1M8, Canada (e-mail: jun.yan@concordia.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2022.3230730.

Digital Object Identifier 10.1109/TSG.2022.3230730

1949-3053 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

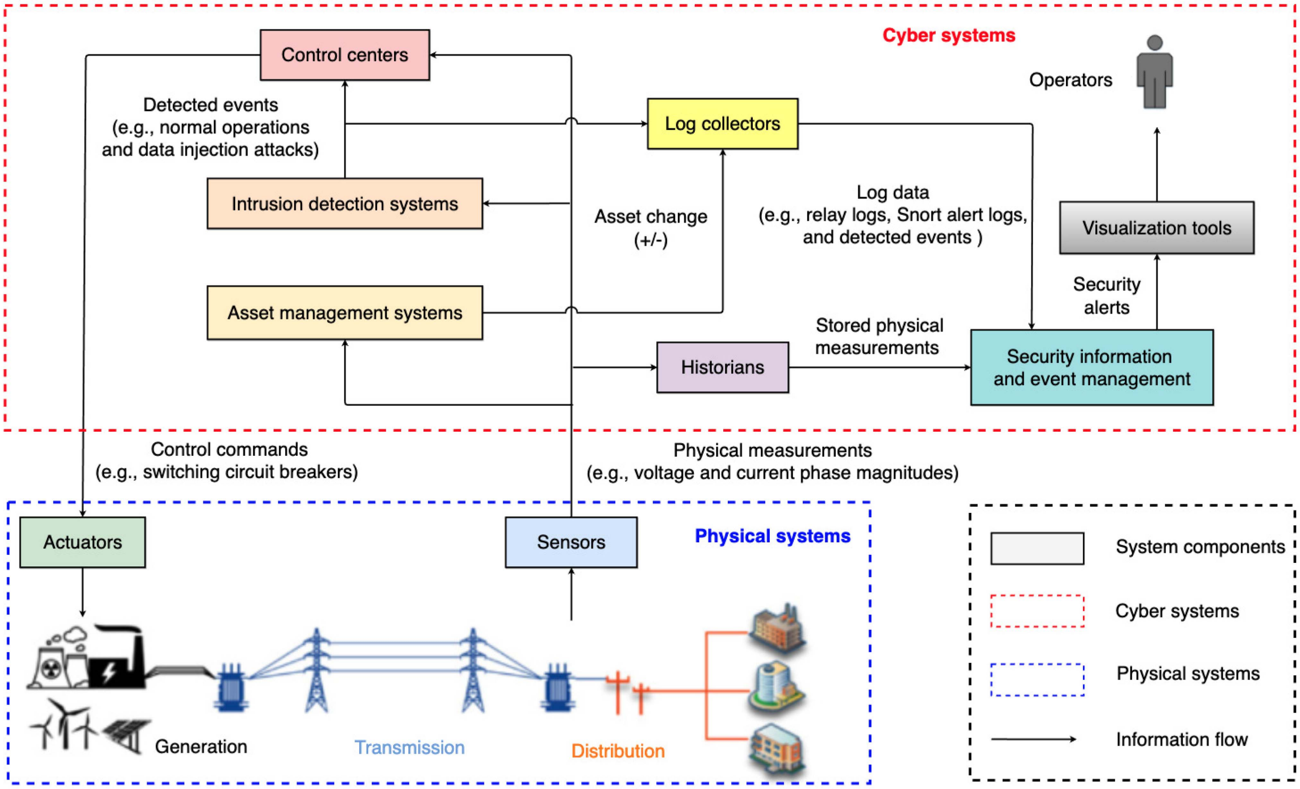


Fig. 1. The information flow in the cyber-physical smart grid.

The ensemble learning-based IDSs embrace a series of homogeneous or heterogeneous base detectors, such as Logistic Regression [7], Density-based spatial clustering [7], Extreme Learning Machine [8], Isolation Forest [9], among others, which aim to improve the detection on challenging samples. The mis-detected samples are commonly assigned larger importance weights, such that new base detectors can focus on the detection on these challenging samples. However, a surge of multi-sourced and heterogeneous smart grid data, which record various concurring synchrophasor measurements and status logs, can include largely redundant and correlated features [2], [3], [4]. Consequently, these feature information leads to critical patterns being hidden in the attack detection, which poses significant challenges to the accurate IDSs in the cyber-physical smart grid [10].

Recently, numerous efforts have been dedicated to the feature extraction and feature boosting to realize the full potential of smart grid data and provide better supports for decision making by IDSs [11]. The existing feature boosting works commonly generate critical features through one feature importance model, which assign each feature an importance score and then select critical features based on the ranking of importance scores. For example, Upadhyay et al. [12] enhance the Recursive Feature Elimination (RFE) model and the XGBoost Classification model, where RFE model scores the feature importance and then the most promising features are recursively selected as the input of XGBoost classifier. Pearson Correlation Coefficient (PCC) is utilized to select the important features that are further transformed into the representative features as the input of Kalman Filter (KF) to discover cyber-attack events [13]. Gradient Boosting-based feature selection approaches are widely regarded as powerful metrics to calculate the importance score (e.g., Gini Index) of each feature while constructing the boosted tree. The effectiveness of selected features is evaluated through the classification performance of boosted tree on various power system attacks [14], [15], [16]. These works manually select critical features through the ranking of importance scores, while it is hard to determine a proper cut-off score to ensure that non-selected features cannot carry away important information and negatively impact the following classification task.

The other common strategy is to develop a single feature extractor through machine learning based methods, which automatically extracts deep and abstract representations from raw data to make the decision. For example, a Bloom filter-based classifier is built on the features extracted through Component Analysis techniques for the anomaly detection in Supervisory Control and Data Acquisition (SCADA) networks [17]. Artificial Feed-forward Network (AFN) includes a Convolution Neural Network to extract representative features, followed by a classifier to identify the false data cyber-attacks in the smart grid [18]. In addition, stacked autoencoders are designed to develop machine-learned features against malicious attacks in the power transmission system [19] and the cyber network [20]. The common path mining-based intrusion detection [21], [22] approaches capture the critical system states to classify disturbances, normal operations and cyber attacks.

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

Sadhasivan and Balasubramanian [23] apply Linear Weighted Cuckoo Search Optimization (LWCSO) feature extraction in the detection of SCADA attacks. Hoeffding Adaptive Tree (HAT) [24] is able to achieve the real-time event classification on heterogeneous data through the drift detection method and adaptive windowing. However, a single feature extractor extracts one abstract feature set over the heterogeneous smart grid data, which may only work well with a certain type of features, such as continuous measurements, and descriptive event logs, etc. Consequently, the extracted feature set may provide partial and limited decision supports for the identification of some events in the complex cyber-physical smart grid.

To provide comprehensive supports and secure the smart grid, our previous work, named Adaptive Feature Boosting (AFB) [25], is proposed to leverage a series of feature extractors and base classifiers to automatically and adaptively extract several feature sets from heterogeneous data for the event classification in the smart grid. In this way, although "poorly-reconstructed" features are well reconstructed with small errors, the extracted feature sets may only include the information that is most sensitive to the reconstruction error and may lose the ability to properly represent all critical information for the event classification, leading to potential degradation of identification performance of some events. Hence, we are still calling for an advanced technique that better extracts discriminative and critical features from the multi-sourced, correlated and heterogeneous smart grid data to support the event classification.

In recent years, Reinforcement Learning has witnessed many successes in addressing the attack detection in the smart grid. For instance, Kurt et al. [26] propose an online cyber-attack detection algorithm using the framework of model-free reinforcement learning, where the defender learns the mapping from observations (i.e., system states) and actions (i.e., stop and continue) by trial-and-error. A reinforcement learning-based attack recovery strategy is introduced to determine the optimal reclosing time by generating the optimal recovery actions after observing the power system dynamics under cyber-attacks [27]. The data integrity attack detection is formulated as an Markov Decision Process in Alternating Current power systems, and further addressed via learning the optimal defending strategy in a Deep-Q-Network Detection (DQND) scheme [28]. With the decision-making ability of Deep-Q-Network, an abnormal flow detection model is proposed to extract features from original data and learn the optimal detection strategy in industrial control systems [29].

C. Contributions

To this end, this paper proposes Reinforcement Learning-based AFB that utilizes a reinforcement learning algorithm to self-adaptively and automatically optimize the feature sampling probability based on the classification accuracy. When the classification accuracy is improved on the extracted feature set, the features in the input of feature extractor have greater chances to be sampled and participate in the next training iteration. Since it is difficult to design an explicit formulation between the feature sampling probability and the classification accuracy, a reinforcement learning approach, named Deep Deterministic Policy Gradient (DDPG) [30], is introduced to dynamically update the feature sampling probability in an implicit way. After training, the critical features that improve the classification accuracy can be assigned large sampling probabilities. With possible diverse alarms generated among all base classifiers, ensemble learning [31] is also introduced to adaptively weight the importance of base classifiers in making the final decision. The results show that our proposed Reinforcement Learning-based AFB can significantly enhance the performance of intrusion detectors in the distinction among normal, fault, and attack events recorded in an open-source dataset from the Hardware-In-the-Loop (HIL) testbed at the Oak Ridge National Lab [32].

The **main contributions** of this paper can be summarized as follows.

- We propose Reinforcement Learning-based AFB, which automatically and adaptively captures critical features from multi-sourced, correlated, and heterogeneous smart grid data through the weighted feature sampling approach, for the purpose of improved event classification.
- We propose the weighted feature sampling approach to formulate the search for critical features as the optimization of dynamic feature sampling probability based on the classification accuracy.
- We employ a reinforcement learning algorithm to adaptively optimize the feature sampling probability toward critical features under the AFB framework.
- We conduct experiments on two realistic security datasets, which show the effectiveness and scalability of our proposed Reinforcement Learning-based AFB over the vanilla AFB and the other state-of-the-art approaches.

The rest of paper is organized as follows: Section II describes the feature extraction, weighted feature sampling process, and ensemble classification in the AFB framework. Section III introduces our Reinforcement Learning-based AFB and DDPG network design. Section IV first describes the benchmark testbed and dataset as well as experiment setup, then the experimental analysis about the feature sampling probability, classification performance and base classifiers, and further compares the performance over the state-of-the-art approaches, and finally demonstrates the experimental results on a more generic dataset. The conclusions and future works are drawn in Section V.

II. The Adaptive Feature Boosting Framework

A. AFB Overview

Many single feature extractors are designed over multi-sourced and heterogeneous data, which may only work well with a certain types of features. As a result, the extracted feature set may only provide partial decision supports for succeeding tasks. To tackle the challenge, the proposed AFB utilizes a series of feature extractors to extract multiple feature sets from different feature subsets; and each extracted feature set is further applied to build a base classifier. Specifically, critical features have greater chances to be present in new feature

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Adaptive Feature Boosting (AFB)

```
Require: Dataset $D$ with $M$ features, feature set $F_t$, maximum episode $N$, maximum timestep $T$.
Initialization:
    Feature sampling probability $P_1(m) = 1/M$;
    Normalize the dataset $D$;
Training:
while not converge do
  for $n = 1, 2, \dots, N$ do
    Set the feature set $F_1 = D$;
    for $t = 1, 2, \dots, T$ do
      Train the feature extractor $X_t$ on $F_t$;
      Extract the feature set $E_t$ using $X_t$;
      Train the base classifier $h_t$ on $E_t$;
      Calculate the final classification accuracy $\varepsilon_t$ of $h_t$ during the training;
      Calculate the base classifier weight $\alpha_t$ of $h_t$;
      Obtain the updated feature sampling probability $P_{t+1}(m)$ according to Algorithm 2;
      Conduct the weighted feature sampling (with replacement) process on $F_t$ using $P_{t+1}(m)$;
      Generate the new feature set $F_{t+1}$ that only includes the sampled features;
    end for
    Integrate all base classifiers into the ensemble classifier $H(x)$ according to Eqn. (1);
  end for
end while
Return: The trained feature extractors $\{X_t\}$, base classifiers $\{h_t\}$, sampled feature sets $\{F_t\}$, and the final ensemble classifier $H(x)$.
```

subsets (i.e., inputs of new feature extractors), so that new feature extractors can focus on the extraction of critical features that improve the classification performance. The boosting is achieved through a weighted feature sampling (with replacement) process, where features are randomly sampled according to the dynamic feature sampling probability determined by the classification accuracy. The critical features can be assigned the larger sampling probabilities during such the weighted sampling process. To produce the final output, an ensemble classifier is eventually built to integrate potential different decisions among all base classifiers. The overall process of AFB is summarized as Algorithm 1.

B. Feature Extraction

Feature extraction has been commonly applied to learn highly-representative features for better situational awareness and decision making [10], [11], [19]. Instead of training a single feature extractor on all features, the proposed AFB iteratively generates multiple feature subsets to train an array of feature extractors individually, during which each feature set is a subset of the previous one. The first feature extractor X_1 is trained on the dataset D, so that all features participate in the training at this point. After training X_1, the extracted feature set E_1 is utilized to train the base classifier h_1. The second feature extractor X_2 is trained on a new feature set F_2 generated through the weighted feature sampling process. From F_2, X_2 extracts the latent feature set E_2, on which the base classifier h_2 is trained. The above recurrent procedure continues until the maximum timestep T is reached. Considering that AutoEncoders (AEs) have been commonly used as feature extractor in the attack detection studies [19], [33], [34], [35], AE is also chosen as our feature extractor in the proposed AFB framework.

C. Weighted Feature Sampling

The feature sampling probability is determined based on the reconstruction errors of all features in [25]. Although the reconstruction errors of "poorly-reconstructed" features are minimized via multiple feature extractors, some hidden yet informative features may be ignored in the extracted feature sets, leading to the potential degradation of classification performance on some classes. To address such the issue, our feature sampling probability is dynamically determined based on the classification accuracy. The features that contribute to the better classification accuracy are assigned large sampling probabilities and increasingly sampled to participate in the training of succeeding feature extractors.

For a dataset D with M features and Y labels, the sampling probabilities of all features are initialized to be identical, i.e., $P_1(m) = \frac{1}{M}$. To avoid dominating features, each feature is normalized into the range of [0, 1] via min-max normalization [36]. Once the feature extractor X_1 and the base classifier h_1 are completed, the feature extractor X_2 is trained on the new feature set F_2, of which the features are sampled from the feature set F_1. The idea of AFB is to capture critical features by increasing their presence in the new feature sets and the training of other more focused feature extractors.

The idea is achieved by a weighted random feature sampling (with replacement) approach based on the classification accuracy. When the classification accuracy is improved on the extracted feature set, the features in the corresponding feature set (i.e., the input of feature extractor) have greater chances to be sampled and present in the new feature set (i.e., the input of next feature extractor). The sampling with replacement means that given the sampling probability $P_2(m)$ for M features, we consecutively and independently sample the feature set F_1 for M times with the same probability $P_2(m)$. Since all features participate in the training of X_1, the sampling probabilities of all features are updated to be equal as $P_2(m)$, which suggests that all features still have same chance to be present in F_2. Ideally, the distribution of new feature set is proportional to the distribution of feature sampling probability, and thus the features with larger sampling probabilities have greater chances to be present in the new feature set.

It is important to note that the feature sampling process is non-deterministic. AFB is not directly ranking and selecting features based on the order of sampling probabilities, since it is difficult to determine a proper cut-off threshold on the sampling probability and the classification accuracy. Instead, we utilize a recurrent and stochastic approach to ensure that the important information is not brought way by the non-sampled features. To maintain the stable architecture of feature extractors in the implementation, non-sampled features are not removed directly. Instead, the values of non-sampled features are set to zeros, which indicates that these features do not participate in the training of new feature extractor X_2. The values of sampled features are retained as in F_1. Without the loss

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

of generality, at the t-th timestep, the new feature set F_{t+1} is sampled from the feature set F_t according to the sampling probability $P_{t+1}(m)$ determined by the classification accuracy of base classifier h_t.

D. Ensemble Classification

Once the weighted feature sampling process is completed, a number of representative extracted feature sets are generated by multiple feature extractors. Each extracted feature set represents a unique subset of information and allows us to train a base classifier at each timestep. Thus, base classifiers can fully explore the diversity and full potential in all extracted feature sets, of which outputs may be consistent or conflicting. To produce the final output, ensemble learning [31] is introduced to integrate potential different decisions from multiple base classifiers. The common choice for a base classifier includes K-Nearest Neighbors (KNN), Artificial Neural Network (ANN), Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF), among others. In this paper, we utilize RF [37] as the base classifier to be trained on the extracted feature set, since RF can learn from both continuous and discrete feature values (e.g., synchrophasor measurements and status logs) and provide more interpretability of decisions, which is essential in the smart grid security analysis [38].

After training all T base classifiers, an ensemble classifier $H(x)$ makes the final decision based on the weighted majority vote of all base classifiers:

$$H(x) = \arg\max_{y \in Y} \sum_{t=1}^{T} \alpha_t \cdot I(h_t(x) = y), \tag{1}$$

where $I(\cdot)$ is the indicator function of base classifier h_t. Each base classifier h_t has an importance weight α_t that depends on the classification accuracy of h_t during the training [25]. Note that h_t with large classification accuracy obtains large α_t, which suggests that more accurate base classifier is more significant in the final combination. By combining multiple base classifiers systematically, the ensemble classifier can integrate the decisions from base classifiers to boost the final classification performance.

III. Reinforcement Learning-Based AFB

A. Reinforcement Learning-Based AFB

Although the feature sampling probability can be determined based on the classification accuracy, it is still difficult to design an explicit formulation between the feature sampling probability and the classification accuracy. In this paper, we formulate the proposed AFB as an Markov Decision Process (MDP) [39] to automatically and dynamically update the feature sampling probability based on the classification accuracy via an implicit mapping.

MDP [39] is a model for the sequential decision making, defined by a 5-tuple $\langle S, A, P, R, \gamma \rangle$, where S is a finite set of states $s_t \in S$ and A is a finite set of actions $a_t \in A$. $P : S \times A \times S \to [0, 1]$ is a transition probability model, where $p(s_{t+1}|s_t, a_t)$ represents the transition probability from s_t to s_{t+1} after a_t is completed. $R : S \times A \to \mathbb{R}$ is a reward function, where $r(s_t, a_t)$ is the immediate reward received when action a_t is taken at state s_t. $\pi : S \to A$ is a policy function, where $\pi(a_t|s_t)$ specifies the action a_t chosen at state s_t. The objective of MDP model is to find an optimal policy π^* that maximizes the cumulative reward $\sum_{t=0}^{\infty} \mathbb{E}[\gamma^t r(s_t, a_t)]$, where $\gamma \in [0, 1]$ is a discount factor. Note that the next state s_{t+1} depends on the current state s_t and the current action a_t.

Algorithm 2 Reinforcement Learning-Based AFB

```
Require: Feature set $F_t$, classification accuracy $\varepsilon_t$, maximum episode $N$, maximum timestep $T$.
Initialization:
    Critic network $Q(s, a|\theta^Q)$ and actor network $\mu(s|\theta^\mu)$ with parameters $\theta^Q$ and $\theta^\mu$;
    Target networks $Q'$ and $\mu'$ with parameters $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$;
Training:
while not converge do
    for $n = 1, 2, \ldots, N$ do
        for $t = 1, 2, \ldots, T$ do
            Calculate the feature sampling probability $P_{t+1}(m)$ according to the current actor network policy $\mu(s_t|\theta^\mu)|_{s_t=F_t}$;
            Obtain the newly sampled feature set $F_{t+1}$ according to Algorithm 1;
            Store the transition $\{F_t, P_{t+1}(m), \varepsilon_t, F_{t+1}\}$ in the replay buffer;
            Randomly sample a minibatch of $K$ transitions $\{F_i, P_{i+1}(m), \varepsilon_i, F_{i+1}\}$ from the replay buffer;
            Calculate $q_i$ according to Eqn. (3);
            Update $Q(s, a|\theta^Q)$ by minimizing $L$ according to Eqn. (2);
            Update $\mu(s|\theta^\mu)$ by using the sampled policy gradient according to Eqn. (4);
            Update $\theta^{Q'}$ according to Eqn. (5);
            Update $\theta^{\mu}$ according to Eqn. (6);
        end for
    end for
end while
Return: The final feature sampling probability $P_{t+1}(m)$, the trained critic network $Q(s, a|\theta^Q)$, and actor network $\mu(s|\theta^\mu)$.
```

In the case of our proposed AFB, the state is the feature set on which the feature extractor is trained; and the action is to update the feature sampling probability toward the critical features; and the immediate reward is the classification accuracy of base classifier. The proposed Reinforcement Learning-based AFB is summarized as Algorithm 2, and Fig. 2 shows the overall architecture. Concretely, our variables are defined as follows:

- *State:* The state s_t at the t-th timestep is defined as the feature set F_t that includes the features sampled (with replacement) from the previous feature set F_{t-1}. The feature extractor X_t is trained on the feature set F_t. Moreover, the dataset D is used as the initial state F_1 at each episode, which suggests that all features participate in the training of first feature extractor X_1.
- *Action:* The action a_t taken by the agent at the t-th timestep is to calculate the feature sampling probability $P_{t+1}(m)$ according to the classification accuracy ε_t of base classifier h_t. Note that when ε_t is better than the

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

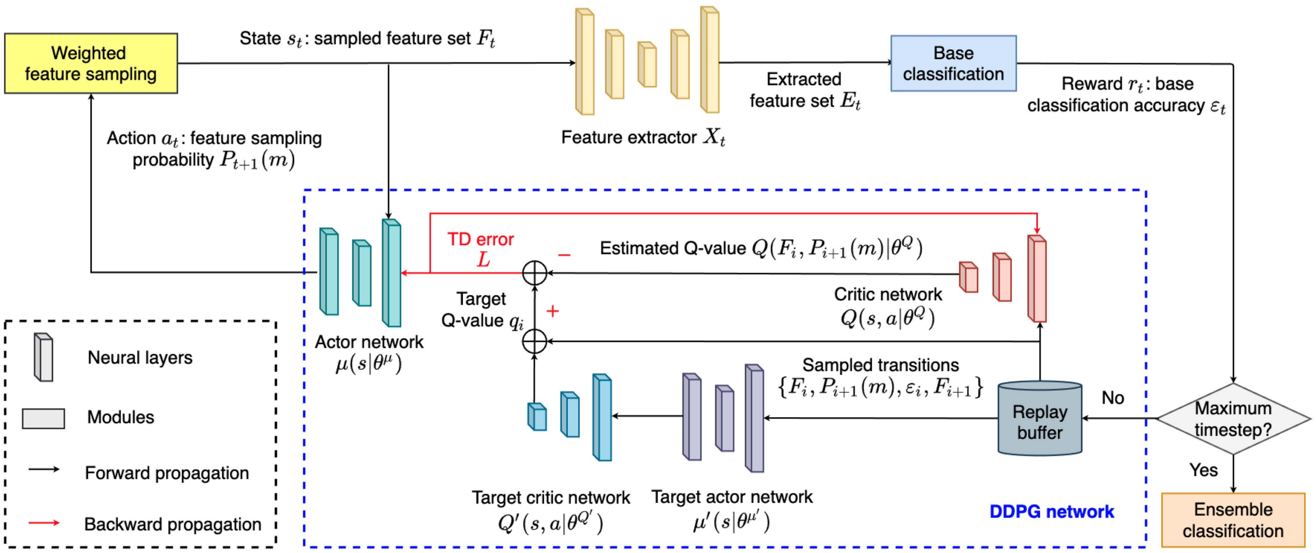

Weighted feature sampling
State s_t: sampled feature set F_t
Feature extractor X_t
Extracted feature set E_t
Base classification
Reward r_t: base classification accuracy ε_t
Action a_t: feature sampling probability $P_{t+1}(m)$
TD error L
−
+
Estimated Q-value $Q(F_i, P_{i+1}(m)|\theta^Q)$
Target Q-value q_i
Critic network $Q(s, a|\theta^Q)$
Actor network $\mu(s|\theta^\mu)$
Sampled transitions $\{F_i, P_{i+1}(m), \varepsilon_i, F_{i+1}\}$
Replay buffer
No
Maximum timestep?
Yes
Ensemble classification
Target critic network $Q'(s, a|\theta^{Q'})$
Target actor network $\mu'(s|\theta^{\mu'})$
DDPG network
Neural layers
Modules
Forward propagation
Backward propagation

Fig. 2. The overall architecture of Reinforcement Learning-based AFB.

accuracy of previous base classifiers, the features in the feature set F_t (i.e., the input of X_t) are updated with the large sampling probabilities. After a_t is taken, the next state s_{t+1} is obtained through the weighted feature sampling (with replacement) process on the current state s_t using the updated sampling probability $P_{t+1}(m)$.

- *Reward:* The immediate reward r_t at the t-th timestep is defined as the classification accuracy ε_t of base classifier h_t. In this way, the features in the feature set F_t that contribute to the improved accuracy are assigned the large immediate rewards, which suggests that these features can provide more discriminative supports for the classification and are updated with the large sampling probabilities. Once the maximum timestep T is reached, all base classifiers are integrated to build an ensemble classifier $H(x)$ to make the final decision.

B. DDPG Network Design

DDPG is chosen as our approach to maximize the cumulative reward, one of the most commonly-used model-free reinforcement learning algorithms, since it shows the advantage of learning policies in the continuous action space and the environment without accurate models [39], [40]. Although model-free reinforcement learning algorithms commonly require a large number of training episodes to learn policies [40], our DDPG can efficiently converge given the relatively small search space in our state-action definition, as demonstrated later in the experimental analysis.

Considering that the samples may not be independently and identically distributed in the environment, DDPG applies the replay buffer as the same approach in Deep Q-Networks [41]. The transition $\{F_t, P_{t+1}(m), \varepsilon_t, F_{t+1}\}$ is stored in the replay buffer, and the actor and critic networks are updated by randomly sampling a mini-batch of transitions $\{F_i, P_{i+1}(m), \varepsilon_i, F_{i+1}\}$ from the replay buffer.

DDPG is an actor-critic algorithm based on Deterministic Policy Gradient [30], [40]. The critic network $Q(s, a|\theta^Q)$ is a Q-value function to evaluate the updated feature sampling probability at each timestep, and is learned using the Bellman equation as the same way in Q-learning [42]. Both the states and the rewards sampled from the replay buffer are utilized to calculate the estimated Q-value $Q(F_i, P_{i+1}(m)|\theta^Q)$ in the critic network. The critic network is updated by minimizing the loss L between the target Q-value q_i and the estimated Q-value $Q(F_i, P_{i+1}(m)|\theta^Q)$. The loss L and the target Q-value q_i are given as follows:

$$L = \frac{1}{K}\sum_{i=1}^{K}\Big(q_i - Q\Big(F_i, P_{i+1}(m)|\theta^Q\Big)\Big)^2, \tag{2}$$

$$q_i = \varepsilon_i + \gamma Q'\Big(F_{i+1}, \mu'\Big(F_{i+1}|\theta^{\mu'}\Big)|\theta^{Q'}\Big), \tag{3}$$

where $Q'(F_{i+1}, \mu'(F_{i+1}|\theta^{\mu'})|\theta^{Q'})$ is the Q-value calculated from the target critic network Q', and K is the number of transitions sampled from the replay buffer.

The actor network $\mu(s|\theta^\mu)$ specifies the policy by mapping a state to a particular action at each timestep. Concretely, the actor network is utilized to calculate the feature sampling probability for a given state at each timestep, and is updated by applying the chain rule to the expected return from the start distribution J with respect to the actor network parameters:

$$\begin{aligned}\nabla_{\theta^\mu} J &\approx \mathbb{E}\Big[\nabla_{\theta^\mu} Q\Big(s, a|\theta^Q\Big)|_{s=F_t, a=\mu(F_t|\theta^\mu)}\Big],\\ &= \mathbb{E}\Big[\nabla_a Q\Big(s, a|\theta^Q\Big)|_{s=F_t, a=\mu(F_t|\theta^\mu)} \nabla_{\theta^\mu}\mu\Big(s|\theta^\mu\Big)|_{s=F_t}\Big],\\ &= \frac{1}{K}\sum_{i=1}^{K} \nabla_a Q\Big(s, a|\theta^Q\Big)|_{s=F_i, a=\mu(F_i|\theta^\mu)} \nabla_{\theta^\mu}\mu\Big(s|\theta^\mu\Big)|_{s=F_i},\end{aligned} \tag{4}$$

where θ^Q and θ^μ are the parameters of critic and actor networks, respectively.

Rather than directly copying the parameters from the critic and actor networks (i.e., the hard update strategy), the target networks are updated via the soft update strategy. Specifically,

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

the target network parameters are updated by slowly tracking the critic and actor network parameters as:

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}, \tag{5}$$

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}, \tag{6}$$

where the soft-update factor $\tau \ll 1$, and $\theta^{Q'}$ and $\theta^{\mu'}$ are the parameters of target critic and actor networks, respectively. With the soft update strategy, the output of target networks (i.e., the target Q-value q_i) is constrained to change slowly, thereby improving the stability of the learning process [30].

IV. Experimental Study

A. Testbed Architecture

This paper utilizes the HIL security testbed set up by the Oak Ridge National Laboratory of the U.S. [32], which represents the representative transmission network with rich operations and attack scenarios in realistic setting, including normal operation, data injection, command injection, relay setting change, line maintenance, among others. Thus, this testbed has generated many representative attack event scenarios targeting the smart grid, which is widely used to develop proof-of-concepts for IDSs in the industrial control systems [12], [13], [14], [15]. Specifically, the testbed [32] is a three-bus two-machine system with four intelligent relays that are wired to a real-time digital simulator to simulate transmission lines, circuit breakers, generators, and load. The relays adopt a distance protection scheme to trip the circuit breakers whenever a fault is detected on a transmission line. All relays are integrated with PMU functionality to measure transmission line states, and a phasor concentrator unit aggregates the synchrophasor measurement data from the four PMU relays and transmits the data to SCADA for the system monitoring, decision controlling, and intrusion detection. In addition, a signature-based IDS (i.e., Snort) runs on a PC to detect remote tripping command activities, and a control panel computer simulates energy management system functionality to disconnect a transmission line for maintenance by remotely tripping relays via a Modbus/TCP network packet.

B. Dataset Description

The benchmark dataset was generated from the HIL security testbed, set up by the Oak Ridge National Laboratory of the U.S. [32]. To generate this dataset, Pan et al. up-sampled the data from sensors with lower sampling rates to match the highest sampling rate of sensor [21]. Specifically, the current phase magnitude measurements are sampled with the highest rate of 120 samples/second in this testbed. Relay status, Snort alerts, and control panel log data should be up-sampled to the nearest sample period after the measurement. All samples without feature values take the non-asserted values.

The descriptions of 128 multi-sourced features are shown in Table I. There are in total 116 PMU measurements and 12 system logs. Each of the four PMU relays provides 29 multi-sourced measurements, including voltage and current sequence components, frequency, and the rate of change of frequency (ROCOF), among others. For example, R1-PA1:VH and R1-PM1:V represent the first voltage phase angle and magnitude measured by the PMU Relay 1, respectively.

TABLE I
Original Features in the HIL Security Dataset [32]

Feature Name	Description	Range
VH	Voltage phase angle	$(-180°, 180°)$
V	Voltage phase magnitude	[0, 152 kV]
IH	Current phase angle	$(-180°, 180°)$
I	Current phase magnitude	[0, 1.78 kA]
F	Frequency on relays	[0, 89.55 Hz]
DF	ROCOF	[-4.01 Hz/s, 5.12 Hz/s]
Z	Apparent impedance	(0.1 Ω, 10.36 kΩ)
ZH	Apparent impedance angle	$(-3.15°, 3.15°)$
S	Status flag of relays	Discrete values
relay_log, control_log, snort_log	Relay logs, Control panel logs, Snort alert logs	Binary: {0, 1}

In addition, the open-source benchmark dataset [21] includes six event scenarios categorizing as three classes: normal operation (Class 0, including one scenario), natural fault (Class 1, including two scenarios), and attack event (Class 2, including three scenarios). Each non-normal event scenario starts from normal operation, and then event occurs, and finally the system state returns to normal operation, during which the load is randomly changed from 200-400 MW. The event scenarios are listed below.

1) *Normal operation (Class 0):* Normal load changes without other attacks, disturbances, or control actions.
2) *Single line-to-ground fault (Class 1):* A short can occur in various locations along one transmission line after the line is grounded. The auto-reclosing scheme models a high speed three-phase scheme that closes the breaker after one second.
3) *Line maintenance (Class 1):* One or more relays are disabled on the transmission line due to the operator maintenance.
4) *Data injection (Class 2):* The attackers imitate valid single line-to-ground faults by manipulating sensor measurement values (e.g., current and voltage phase components) followed by sending illicit trip commands to relays, which could blind operators and cause a blackout. To launch the attacks, the attackers need the configurations of the power system (e.g., the system topology) and the knowledge of the state estimation function, so that the attackers can mislead the state estimation process by injecting malicious PMU measurements [43].
5) *Remote tripping command injection (Class 2):* The attackers remotely send unexpected tripping commands from nodes on the communications network with a spoofed legitimate IP address to relays at the transmission lines, leading breakers to open. To launch the attacks, the attackers first need to find vulnerable and critical breakers, and then need to know the duration that attackers should follow to send malicious commands to periodically open and close these breakers, leading to the serious power system oscillation [44].
6) *Relay setting change (Class 2):* The attackers change relay settings (e.g., timer values) to disable a distance

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

protection scheme by accessing internal registers with Modbus/TCP commands. To launch the attacks, the attackers need to gain the access to the HMI system, so that the attackers can know the configuration information of SCADA system and take illicit control actions to manipulate the relay operations [21].

Besides, data injection attacks and relay setting change attacks can occur in various locations along the transmission lines, and the locations are indicated by the percentage range. For example, a data injection attack occurs at 85% of the first transmission line, which maliciously changes the values of measurements (e.g., current phase magnitude R2:PM4:I, voltage phase magnitude R2:PM2:V, relay frequency R2:F) collected from PMU Relay 2. A relay setting change attack can be launched to disable Relay 3 located at 10% - 20% of the second transmission line, which maliciously manipulates the relevant feature values, such as R3:F (relay frequency), R3:DF (ROCOF) and R3:relay_log (relay log) collected from PMU Relay 3. Moreover, remote tripping command injection attacks can occur at single or two relays. For instance, a command injection attacker simultaneously attacks both Relay 1 and Relay 2, during which the features R1:S and R2:S represent the statuses of two relays under the attack.

C. Experiment Setup

The number of samples in Class 0, 1, and 2 are 4,405, 18,309, and 55,663, respectively; and 80% of each class are used for training set and the remaining 20% for testing in our experiments. As an imbalanced dataset, the normal and fault samples are oversampled to balance the attack samples in the training set, while the testing set is not oversampled and remains imbalanced. Moreover, data normalization is essential to be performed to improve the uniformity of features by adjusting all feature values into the range of [0, 1].

It is important to note that the dimensionalities of state s_t and action a_t are both the same as the number of features (128). In our DDPG framework, the actor network takes the state as input and outputs the corresponding action, and the critic network calculates the estimated Q-value output based on the input of state and action. The actor and critic networks both include two hidden layers that contain 80 and 60 hidden neurons, respectively. The capacity size of replay buffer is set to 20,000, and the actor and critic networks are updated by randomly sampling K (5,000) transitions from the replay buffer. Note that we will start the training of actor and critic networks when the replay buffer is full, and the replay buffer is updated by discarding the oldest transitions. The discount factor γ and the soft-update factor τ are set to 0.95 and 0.001, respectively. In addition, each AE extracts 30 latent features and includes three hidden layers that contain 64, 30, and 64 hidden neurons, respectively. A RF multi-class base classifier, including 10 classification and regression trees, is built on each unique AE-extracted feature set. Thus, all base classifiers can explore the diversity and full potential in all extracted feature sets, of which outputs could be consistent or conflicting. To produce the final output, a multi-class ensemble classifier is generated based on the weighted voting of all the base classifiers and makes the final decision to classify samples among normal operations, natural faults, and attack events.

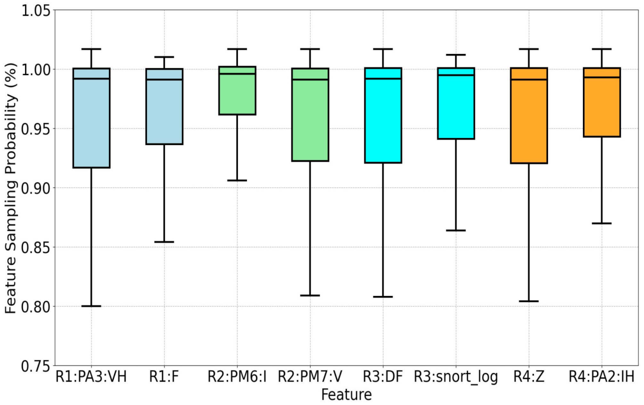


Fig. 3. The boxplot of final sampling probabilities (in %) of eight examples of critical features: R1:PA3:VH, R1:F, R2:PM6:I, R2:PM7:V, R3:DF, R3:snort_log, R4:Z, and R4:PA2:IH over 100 experiments.

D. Experimental Analysis on Feature Sampling Probability and Classification Accuracy

The sampling probabilities of all 128 features are initialized to be identical as 0.78×10^{-2}. After training, the sampling probabilities of 24 features are decreased to less than 10^{-5}, such as the frequency from Relay 2, the control panel logs from Relay 2 and Relay 3, and the voltage phase angle from Relay 4, which suggests that these features are unlikely to be sampled to participate in the training of new feature extractor. Moreover, the sampling probabilities of remaining 104 features, including 27, 26, 22 and 29 features from Relay 1, Relay 2, Relay 3, and Relay 4, respectively, of which final sampling probabilities range between 0.79×10^{-2} and 1.22×10^{-2}. These 104 features have greater chances to participate in the feature extraction and provide discriminative supports for the event classification. Fig. 3 shows the boxplot of final sampling probabilities of eight examples of critical features, including R1:PA3:VH, R1:F, R2:PM6:I, R2:PM7:V, R3:DF, R3:snort_log, R4:Z, and R4:PA2:IH. Note that the feature sampling probabilities are indicated in % on the y-axis. All these eight features report the similar median, 75% percentile as well as maximum of sampling probabilities, with the values around 0.97×10^{-2}, 1.0×10^{-2}, and 1.02×10^{-2}, respectively. The 25% percentiles of sampling probabilities range between and 0.92×10^{-2} and 0.96×10^{-2}.

Table II demonstrates the average sampling probabilities (with standard deviations) of all the 24 uncritical features and 24 examples of critical features over the different episodes. The sampling probabilities of the 24 critical features eventually stabilize around 1.01×10^{-2} after 300 episodes, which relatively increase by roughly 29.49% over the initial probability with 0.78×10^{-2}. The standard deviation is simultaneously decreased to 0.38×10^{-4} over time. For the uncritical features, the final feature sampling probabilities all decrease to less than 10^{-5} after 300 episodes. Specifically, the average sampling probabilities of all uncritical features are around 0.19×10^{-4} after 60 episodes, with the relative decreasing

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

TABLE II
AVERAGE FEATURE SAMPLING PROBABILITIES WITH STANDARD DEVIATIONS AT DIFFERENT EPISODES ON THE HIL SECURITY DATASET

Feature Name	**Episode**	**Feature Sampling Probability** ($\times 10^{-4}$)
Critical features: R1:PM1:V, R1:PA3:VH, R1:PM11:IH, R1:relay_log, R1:Z, R1:F, R2:PA2:VH, R2:PA5:IH, R2:PM6:I, R2:PM7:V, R2:PA8:VH, R2:DF, R3:PM1:V, R3:PM6:I, R3:PA8:VH, R3:DF, R3:F, R3:snort_log, R4:PA2:IH, R4:PM4:I, R4:PM7:V, R4:ZH, R4:Z, R4:control_log	0	78.13
	50	89.42 ± 0.92
	100	93.87 ± 0.87
	150	97.65 ± 1.56
	200	99.18 ± 0.67
	250	99.67 ± 0.40
	300	101.22 ± 0.38
Uncritical features: R1:PM7:V, R1:PA8:VH, R1:PM9:V, R1:snort_log, R1:relay_log, R2:PA4:IH, R2:PM9:V, R2:PM11:I, R2:S, R2:F, R2:control_log, R3:PA4:IH, R3:PM5:I, R3:PM7:V, R3:PM8:V, R3:PA11:IH, R3:PM12:I, R3:Z, R3:S, R3:control_log, R3:relay_log, R4:PA8:VH, R4:F, R4:relay_log	0	78.13
	10	39.15 ± 7.82
	20	9.09 ± 5.74
	30	1.04 ± 0.58
	40	0.43 ± 0.29
	50	0.24 ± 0.18
	60	0.19 ± 0.15

TABLE III
AVERAGE CLASS-WISE AND OVERALL CLASSIFICATION ACCURACY WITH STANDARD DEVIATIONS OF THE BASE AND ENSEMBLE CLASSIFIER ON THE HIL SECURITY DATASET

Classifier	**Class**	**Class-wise Accuracy** (%)	**Overall Accuracy** (%)
Base classifier h_{16}	0	90.49 ± 0.72	93.99 ± 0.76
	1	**92.21** ± 0.73	
	2	94.62 ± 0.71	
Base classifier h_{23}	0	**90.79** ± 0.72	94.29 ± 0.72
	1	91.83 ± 0.76	
	2	94.89 ± 0.81	
Base classifier h_{35}	0	90.70 ± 0.70	**94.41** ± 0.82
	1	92.09 ± 0.75	
	2	**95.29** ± 0.71	
Ensemble classifier $H(x)$	0	**94.47** ± 0.58	**97.28** ± 0.60
	1	**96.23** ± 0.61	
	2	**97.94** ± 0.62	

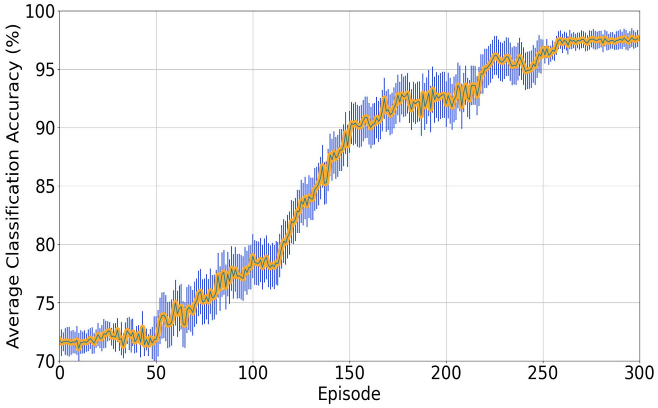


Fig. 4. The average and standard deviation of classification accuracy (in %) over 100 experiments. The orange solid line indicates the average classification accuracy at each episode, and the blue shaded region indicates the standard deviation of classification accuracy over 100 experiments.

probabilities of 99.97%. This indicates that the sampling probability learning process of uncritical features is notably faster than that of critical features. In addition, it is observed that the importances of features with the same measurement property can be similar or diverse. For instance, the final sampling probabilities of features R1:F and R3:F are 0.99×10^{-2} and 1.0×10^{-2}, which are significantly larger than the probabilities of features R2:F and R4:F. Thus, R1:F and R3:F are identified as critical features and have greater chances to participate in the feature extraction to provide discriminative information for the event classification. The closer comparison between the feature sampling probabilities (with the same measurement property) suggests the potential relation of the critical features and the benchmark topology, which will be further investigated in the future work of this paper.

The average classification accuracy over 100 repeated experiments is shown in Fig. 4, which reports an error band with the average and the standard deviation of classification accuracy over 100 repeated experiments at each episode. Note that the shaded region indicates the standard deviation over 100 experiments, and the solid line indicates the mean. Each RF multi-class base classifier h_t is built on an AE-extracted feature set E_t at each timestep, and 40 base classifiers are eventually combined as a multi-class ensemble classifier $H(x)$ to simultaneously classify normal, fault and attack events at each episode. The average classification accuracy is not only improved from 71.54% to 97.28% after 300 episodes, but the standard deviation is also decreased to 0.60% over time.

E. Experimental Analysis on Different Base Classifiers

Table III shows the average class-wise and overall classification accuracy with standard deviations over 100 repeated experiments. The best average class-wise and overall accuracy on the base and ensemble classifiers are highlighted in bold. Specifically, the base classifier h_{16} has the best average accuracy in Class 1 with 92.21%, and the base classifier h_{23} is the best one in Class 0 with the average accuracy of 90.79%. In addition, the base classifier h_{35} achieves the best average class-wise accuracy in Class 2 and overall accuracy, with the values of 95.29% and 94.41%, respectively. After integrating all 40 base classifiers, the ensemble classifier $H(x)$ can outperform all the base classifiers in terms of average class-wise and overall accuracy, of which values are 94.47%, 96.23%, 97.94%, and 97.28%, respectively. This suggests that ensemble learning can effectively integrate potentially different decisions from base classifiers to improve the classification performance.

Fig. 5 illustrates the average and standard deviation of classification accuracy on the ensemble classifiers with different numbers of base classifier over 100 experiments. It is observed that the classification accuracy can be boosted by systematically combining more RF base classifiers. For example, after integrating five RF base classifiers, the ensemble classifier reports the average classification accuracy with 94.49%. With the increasing number of base classifiers (from 5 to 40), the

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
CONFUSION MATRICES COMPARISON BETWEEN VANILLA AND REINFORCEMENT LEARNING-BASED AFB MODELS ON THE HIL SECURITY DATASET

Model Type	**Predict** / **Actual**	**Class 0**	**Class 1**	**Class 2**	**TPR**	**FNR**	**FPR**	**FAR**	**F1-score**	**MCC**	**Overall Accuracy** (%)
Vanilla AFB [25]	Class 0	767.3	15.8	99.0	0.871	0.129	0.003	0.132	0.904	0.899	91.78
	Class 1	5.8	2,990.1	667.1	0.817	0.183	0.039	0.222	0.84	0.887	
	Class 2	43.0	459.5	10,631.5	0.955	0.045	0.169	0.214	0.944	0.801	
Reinforcement Learning-based AFB	Class 0	832.2	5.6	44.9	**0.944**	**0.056**	**0.002**	**0.058**	**0.95**	**0.945**	**97.28**
	Class 1	8.6	3,517.1	137.1	**0.960**	**0.040**	**0.017**	**0.057**	**0.952**	**0.937**	
	Class 2	28.8	204.5	10,901.5	**0.979**	**0.021**	**0.040**	**0.061**	**0.981**	**0.936**	

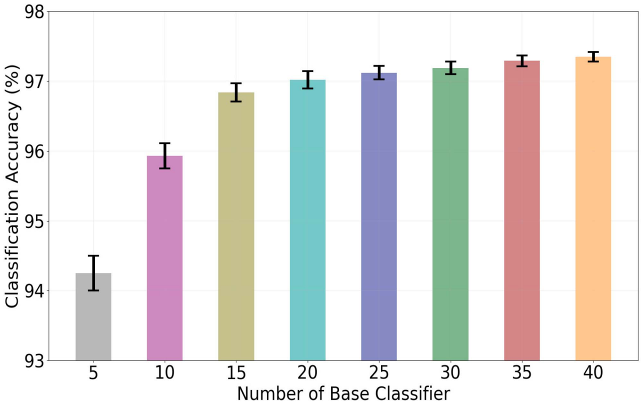


Fig. 5. The average and standard deviation of ensemble classification accuracy (in %) with different numbers of base classifier over 100 experiments.

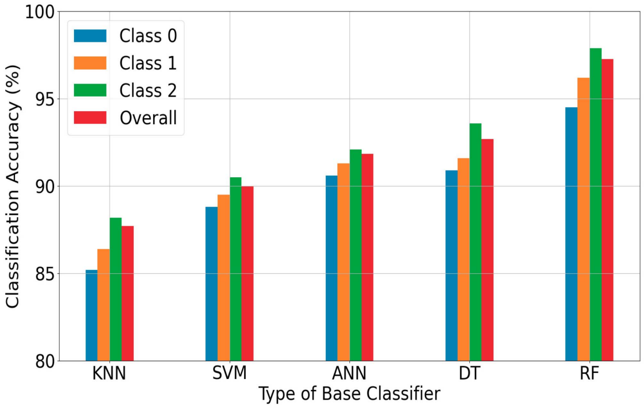


Fig. 6. The bar plots of class-wise and overall classification accuracy (in %) on the ensemble classifiers with different types of base classifier.

average classification accuracy is increased from 94.49% to 97.28%; and the robustness against outliers is also improved over time, with the relative decreasing standard deviation of classification accuracy of 71.84%.

To demonstrate the effectiveness of RF-based ensemble classifier, Fig. 6 shows the class-wise and overall classification accuracy on the different ensemble classifiers with different types of base classifiers (i.e., KNN, SVM, ANN, DT, and RF). The number of neighbors is set to 10 in KNN, and ANN is built by applying a softmax function after AE. For each type of base classifier, an ensemble classifier is built based on the weighted majority voting of 40 base classifiers. It is seen that our RF-based ensemble classifier achieves the best overall accuracy with the value of 97.28%, which can outperform the other ensemble classifiers by 9.57% (KNN), 7.27% (SVM), 5.44% (ANN), and 4.58% (DT), respectively. Besides, DT-based ensemble classifier achieves the second-best class-wise and overall accuracy, with the values of 90.93% (Class 0), 91.61% (Class 1), 93.64% (Class 2), and 92.7% (Overall), respectively. This indicates that the tree-based approaches (i.e., DT and RF) can show the stronger ability to monitor and detect prominent threats in the smart grid security analysis.

F. Performance Comparison Over State-of-the-Art Approaches

Table IV shows the confusion matrices that compare the vanilla AFB [25] and the proposed Reinforcement Learning-based AFB over 100 experiments. To demonstrate the performance of each event in the imbalanced classification problem, we decompose the class-wise performance into the following matrices:

1) *True Positive Rate (TPR):* the fraction of samples that are correctly classified.
2) *False Negative Rate (FNR):* the fraction of samples that are misclassified as other events.
3) *False Positive Rate (FPR):* the fraction of samples in other events that are misclassified as this event.
4) *False Alarm Rate (FAR):* the addition of FNR and FPR.
5) *F1-score:* the harmonic mean of precision and recall.
6) *Matthew Correlation Coefficient (MCC):* the Pearson Correlation Coefficient between the actual and predicted event.

Note that the positive and negative rates are all measured as One-Versus-All performances in the multi-class classification problem, such as normal (Class 0) vs. abnormal (Class 1 and Class 2).

The proposed Reinforcement Learning-based AFB exhibits the large improvement over the vanilla AFB [25], with the increasing overall classification accuracy of 5.5%. For the class-wise performance, it is observed that the proposed method outperforms the vanilla AFB with the increasing F1-scores of 0.046, 0.112, and 0.037, respectively. Compared to the vanilla AFB, the proposed method is not only the better one with the FAR values of 0.058, 0.057 and 0.061, but also increases MCCs by 0.046, 0.05 and 0.135 in the three classes, respectively. Moreover, it can be seen that the misclassification of the proposed method mainly arises from the values of FNR. For instance, the proposed approach has the largest FNR and the smallest FPR in Class 0, with the values of 0.056 and 0.002, respectively.

Table V shows the classification performance comparison between our proposed Reinforcement Learning-based AFB

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

TABLE V
CLASSIFICATION PERFORMANCE AND EXECUTION TIME COMPARISON WITH THE STATE-OF-THE-ART APPROACHES ON THE HIL SECURITY DATASET

Model Type	**Classification Task**	**TPR**	**FPR**	**F1-score**	**Accuracy** (%)	**T-score** ($\times 10^4$)	**Training Time** (second)	**Testing Time** (second)
RFE-XGBoost [12]	Binary	0.966	0.028	0.969	98.24	1.13	306.04	3.63
	Three-class	0.961	0.052	0.966	97.95	1.79	447.19	6.96
PPAD-CPS [13]	Binary	0.963	0.035	/	96.82	2.18	/	/
GBFS [15]	Binary	0.974	0.037	0.972	97.96	1.36	289.37	2.50
	Three-class	0.968	0.067	0.978	96.80	2.44	375.28	3.32
CatBoost [16]	Binary	0.959	0.033	0.974	98.09	1.62	549.62	7.89
LWCSO-PKM [23]	Binary	0.968	/	0.988	98.90	0.74	/	/
HAT [24]	Five-class	/	/	/	96.85	2.17	/	/
Reinforcement Learning-based AFB	Binary	0.994	0.003	0.996	99.98	/	265.83	2.27
	Three-class	0.981	0.024	0.985	99.22	/	314.55	2.92
	Five-class	0.963	0.037	0.972	98.06	/	390.31	3.44

TABLE VI
TRAINING AND TESTING TIME OF DIFFERENT MODULES ON THE HIL SECURITY DATASET

Classification Task	**Module**	**Training Time** (second)	**Testing Time** (second)	**Overall Training Time** (second)	**Overall Testing Time** (second)
Binary	Feature extraction	182.96	1.48	265.83	2.27
	Weighted feature sampling	48.14	/		
	Ensemble classification	34.73	0.79		
Three-class	Feature extraction	185.16	1.37	314.55	2.92
	Weighted feature sampling	50.21	/		
	Ensemble classification	79.18	1.54		
Five-class	Feature extraction	184.73	1.51	390.31	3.44
	Weighted feature sampling	47.59	/		
	Ensemble classification	157.93	1.93		

and the state-of-the-art approaches in the different classification tasks, which is based on the t-Test with 100 repeated experiments under the 5% level of significance. Note that the t-scores are calculated based on the accuracy results of the proposed method and the state-of-the-art method over 100 experiments, and the t-Test is not applicable to the proposed Reinforcement Learning-based AFB. Specifically, the classification tasks are categorized as follows:

1) *Binary classification:* the distinction between non-attack and attack events.
2) *Three-class classification:* the classification among normal, fault, and attack events.
3) *Five-class classification:* the classification among five types of events: normal operation, line maintenance, command injection attack, SLG fault replay attack, and relay disabling attack.

Our proposed method achieves the better classification performance over the state-of-the-art approaches that report the classification accuracy over 96.0% on a down-sampled dataset [12], [13], [15], [16], [23], [24]. For example, the proposed method outperforms the second-best approach (i.e., LWCSO-PKM [23]) in the binary classification task, with the increasing TPR, F1-score and accuracy values of 0.026, 0.008, and 1.08%, respectively. In the three-class classification task, the proposed method exhibits the significant performance improvement over the other boosting-based methods (i.e., RFE-XGBoost [12] and GBFS [15]), which can relatively improve TPR, FPR, F1-score and accuracy, by up to 2.08%, 64.18%, 1.97%, and 2.5%, respectively. Moreover, all *t*-scores are significantly greater than $t_{0.05,99} = 1.6$. This suggests that the alternative hypothesis can be accepted under the 5% level of significance, which is a strong evidence that the proposed Reinforcement Learning-based AFB achieves better classification performance in all the classification tasks.

In addition, Table V also compares the execution time (i.e., training and testing time) between our proposed Reinforcement Learning-based AFB and the state-of-the-art approaches in the binary and three-class classification tasks. Note that we were unable to provide the training and testing time of the state-of-the-art approaches [13], [23], [24], since their computational costs and codes are not openly public. It is observed that our proposed method is most computationally efficient than all the state-of-the-art approaches in the binary and three-class classification tasks. For example, according to our results, GBFS [15] costs the least training and testing time among all the state-of-the-art approaches. Compared to GBFS [15], our proposed method relatively reduces the training and testing time by 8.13% and 9.20% in the binary classification task, as well as 16.18% and 12.05% in the three-class classification task. This suggests that our proposed method not only achieves faster offline training, but also can be more suitable for real-time applications. Moreover, CatBoost [16] also builds a boosting classifier, while costing the most training and testing time (549.62 seconds and 7.89 seconds) than RFE-XGBoost [12], GBFS [15] as well as our proposed method. One potential reason is that such CatBoost classifier [16] is trained on all original features without feature extraction and selection techniques, which indicates that the computational efficiency of event classification can be improved through the capture of informative features.

To better demonstrate the computational efficiency of the proposed Reinforcement Learning-based AFB, Table VI

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

further shows the average training and testing time of different modules (i.e., feature extraction, weighted feature sampling, and ensemble classification) over 100 experiments. Note that since critical features are eventually determined after the training, the weighted feature sampling will not be conducted during the testing stage. Compared to the feature extraction and the ensemble classification, the weighted feature sampling costs the least training time in the three-class and five-class classification tasks, with the proportion of 15.95% and 12.19% in the overall training time, respectively. From the binary task to the five-class classification task, the training and testing time of ensemble classification are increased to 157.93 seconds and 1.93 seconds, while the training time of feature extraction and weighted feature sampling can stabilize around 184.28 seconds and 48.64 seconds, respectively. Hence, with the increasing difficulty of the classification task, the feature extraction and the weighted feature sampling can still remain computationally efficient during the training and testing stage, and the ensemble classification could bring more computational cost to make the decision. To further improve the computational efficiency of the proposed method, the feature extractors and the base classifiers can be concurrently trained in the feature extraction and ensemble classification, respectively, which will be investigated in the future work of this paper.

G. Scalability of Reinforcement Learning-Based AFB

1) Evaluation Dataset: To demonstrate the scalability of the proposed Reinforcement Learning-Based AFB, we also evaluate the proposed method on the widely-used and more generic Industrial Internet of Things (IIOT) security dataset WUSTIL-IIOT-2021 Dataset [45]. Specifically, Zolanvari et al. [45] developed the realistic testbed that includes an actual industrial plant, three sensors, four actuators, among others. Based on this emulated testbed, they generated the realistic WUSTIL-IIOT-2021 Dataset [45], which includes normal operation and four types of cyber-attack events (i.e., reconnaissance, command injection, DoS, and backdoor). The attack events can be summarized as follows:

1) *Reconnaissance:* The attackers gather the detailed information about the physical process, which can be considered as the first step to launch attacks.
2) *Command injection:* The attackers connect to the communication network and gain the access to the PLC register information, which aim to rewrite the PLC registers that are essential to the physical process.
3) *Denial of service (DoS):* The attackers launch sequences of malicious attempts against legitimate users to disrupt access to normal services.
4) *Backdoor:* The attackers open the ports to gain the access to the HMI system, which aim to fully explore the system information and remove important files to disrupt the HMI operations.

In addition, this dataset includes 41 features, of which the values can be dynamically changed from the normal operation stage to the attack stage, such as Tload (total bits per second), Sloss (source packets retransmitted or dropped), Drate (destination packets per second), among others. To evaluate the performance of the proposed method on the WUSTIL-IIOT-2021 dataset, the sampling probabilities of all 41 features are initialized to be identical as 0.0243, and we still remain the same network architecture and hyperparameter setting as the experiment setup on the HIL security dataset.

TABLE VII
AVERAGE FINAL SAMPLING PROBABILITIES WITH STANDARD DEVIATIONS OF CRITICAL FEATURES ON THE WUSTIL-IIOT-2021 DATASET

Feature Name	**Description**	**Final Feature Sampling Probability** ($\times 10^{-2}$)
Mean	Average duration of active flows	11.87 ± 0.58
Sport	Source port number	10.93 ± 0.47
Tload	Total bits per second	10.59 ± 0.54
Tpkts	Total transaction packet count	10.41 ± 0.48
Ploss	Percent packets retransmitted or dropped	9.79 ± 0.43
Sloss	Source packets retransmitted or dropped	9.25 ± 0.41
Dport	Destination port number	8.90 ± 0.52
Sbytes	Source bytes count	8.33 ± 0.44
Dur	Record total duration	8.17 ± 0.51

2) Experimental Result and Analysis: Table VII shows the average and standard deviation of final sampling probabilities of the critical features over 100 repeated experiments. It is seen that nine critical features can be determined with the final sampling probabilities larger than 0.08. For example, the feature Mean is assigned the largest sampling probability after the training, which is increased from 0.0243 to 0.1187. Thus, this feature has the greatest chance to participate in the feature extraction and ensemble classification. Moreover, the standard deviations of these final sampling probabilities can simultaneously stabilize between 0.41 $\times 10^{-2}$ and 0.58 $\times$ 10^{-2}, which indicates the robustness of the weighted feature sampling process. In addition, the final sampling probabilities of the remaining 32 features are all decreased to less than 0.01, which indicates that these uncritical features are unlikely to sampled for the training of feature extractors. For example, Trate (total packets per second) and Proto (transaction protocol) report the final largest and smallest sampling probabilities among all the uncritical features, with the values of 0.0098 and 6.62×10^{-5}, respectively. Hence, we can demonstrate that the proposed method has the strong capacity to accurately and stably select critical features based on the significant difference of feature sampling probabilities.

With the selected critical features, our proposed method can outperform the other state-of-the-art approaches using the same sample number of each class as these works [45], [46], [47]. After generating WUSTIL-IIOT-2021 Dataset [45], Zolanvari et al. also introduced Transparency Relying Upon Statistical Theory and Explainable AI (TRUST-XAI) [46] and Anomaly Detection using Distributed AI (ADDAI) [47] as two representative case studies using this security dataset. Specifically, TRUST-XAI [46] first leverages multi-modal Gaussian distributions to calculate the prediction probability of each class, and then provides the transparency through a surrogate explainer. ADDAI [47] is composed of AE and AdaBoost

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

TABLE VIII
CLASSIFICATION PERFORMANCE AND EXECUTION TIME COMPARISON WITH THE STATE-OF-THE-ART APPROACHES ON THE WUSTIL-IIOT-2021 DATASET

Model Type	**MCC**	**FNR** (%)	**Acc.** (%)	**T-score**	**Training Time** (second)	**Testing Time** (second)
TRUST-XAI [46]	0.909	0.23	98.98	364.07	212.65	2.57
ADDAI [47]	0.969	0.25	99.78	110.33	/	/
Our Method	0.982	0.21	99.89	/	146.81	1.39

classification modules, which aims to improve the detection performance as well as minimize the communication cost between the sensors and the cloud. Table VIII compares the binary classification performance and execution time between our proposed method and these two latest existing works [46], [47], in terms of MCC, FNR, accuracy, t-score, training time, and testing time. The t-scores are still calculated based on the average and standard deviation of accuracy over 100 experiments. Note that the authors have not provided the computational cost in ADDAI [47] and the t-Test is not applicable to the proposed Reinforcement Learning-based AFB. The proposed method achieves better classification performance over ADDAI [47], which relatively increases MCC and FNR by around 1.34% and 16.0%, respectively. Moreover, the proposed method not only exhibits the significant improvement over TRUST-XAI [46] with the relative increasing MCC of 8.03%, but also reduces the training and testing time by 65.81 seconds and 1.18 seconds, respectively. This suggests that after the faster training, our proposed method is able to more accurately and timely to detect malicious attacks on the WUSTIL-IIOT-2021 dataset. With the t-score values of 364.07 and 110.33 (greater than $t_{0.05,99} = 1.6$), there is strong evidence to indicate that our proposed method can outperform these existing works.

V. CONCLUSION AND FUTURE WORK

This paper proposes the Reinforcement learning-based AFB for intrusion detection involving the multi-sourced data in the smart grid and similar large-scale cyber-physical systems, which utilizes DDPG to adaptively and automatically optimize the feature sampling probability based on the classification accuracy. The critical features are assigned large sampling probabilities and increasingly sampled to participate in the feature extraction, so that more diverse discriminative information can be extracted to support the event classification with respect to different operational scenarios and data sources in the smart grid. With a series of AEs and RF-based ensemble classifiers, the proposed method is able to distinguish critical and uncritical features among the correlated synchrophasors, relay status, system logs, and events, by adaptively assigning distinctive sampling probabilities to informative features for better information representation and event classification.

The evaluations on the HIL security dataset demonstrate that our proposed approach achieves the classification accuracy with 97.28% on the classification among normal, fault, and attack events, which outperforms the vanilla AFB with the classification accuracy of 91.78% [25]. Compared to the state-of-the-art approaches [12], [13], [15], [16], [23], [24], our proposed approach improves the classification performance by up to 3.16%, 2.42% and 1.21% in the binary, multi-class and five-class event classification, respectively. Moreover, the proposed method can accurately and stably select critical features on the WUSTIL-IIOT-2021 dataset based on the significant difference of feature sampling probabilities between critical and uncritical features (i.e., the probabilities greater than 0.08 and less than 0.01). Compared to the other best-performing approaches [46], [47], the proposed approach not only increases MCC by up to 8.03%, but also improves the computational efficiency with the reduced training and testing time of 65.81 seconds and 1.18 seconds, respectively. Although our testbed and dataset can represent a high-level abstraction of transmission networks with rich operations and attack scenarios in realistic setting, the proposed approach will be further validated on a larger-scale security testbed with more buses and hardware. Moreover, while the proposed approach is able to identify critical features based on the sampling probabilities, the approach will be further developed to discover an optimal feature set from the critical features.

REFERENCES

[1] J. McCarthy et al., "Situational awareness," NIST, Gaithersburg, MD, USA, document SP 1800-7B, 2019.

[2] H. Daki, A. El Hannani, A. Aqqal, A. Haidine, and A. Dahbi, "Big data management in smart grid: Concepts, requirements and implementation," *J. Big Data*, vol. 4, no. 1, pp. 1–19, 2017.

[3] I. Kiaei and S. Lotfifard, "Fault section identification in smart distribution systems using multi-source data based on fuzzy Petri nets," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 74–83, Jan. 2020.

[4] H. Sun, Z. Wang, J. Wang, Z. Huang, N. Carrington, and J. Liao, "Data-driven power outage detection by social sensors," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2516–2524, Sep. 2016.

[5] R. E. Mackiewicz, "Overview of IEC 61850 and benefits," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, 2006, pp. 1–8.

[6] F. M. Cleveland, "IEC 62351-7: Communications and information management technologies-network and system management in power system operations," in *Proc. IEEE/PES Transm. Distrib. Conf. Expo.*, 2008, pp. 1–4.

[7] M. Zhou, Y. Wang, A. K. Srivastava, Y. Wu, and P. Banerjee, "Ensemble-based algorithm for synchrophasor data anomaly detection," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2979–2988, May 2019.

[8] A. Khamis, Y. Xu, Z. Y. Dong, and R. Zhang, "Faster detection of microgrid islanding events using an adaptive ensemble classifier," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1889–1899, May 2018.

[9] E. Khaledian, S. Pandey, P. Kundu, and A. K. Srivastava, "Real-time synchrophasor data anomaly detection and classification using isolation forest, KMeans, and LoOP," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2378–2388, May 2021.

[10] C. Hu, J. Yan, and C. Wang, "Robust feature extraction and ensemble classification against cyber-physical attacks in the smart grid," in *Proc. IEEE Electr. Power Energy Conf. (EPEC)*, 2019, pp. 1–6.

[11] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[12] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Intrusion detection in SCADA based power grids: Recursive feature elimination model with majority vote ensemble algorithm," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2559–2574, Jul.–Sep. 2021.

[13] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Trans. Sustain. Comput.*, vol. 6, no. 1, pp. 66–79, Jan.–Mar. 2021.

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.

[14] M. H. L. Louk and B. A. Tama, "Exploring ensemble-based class imbalance learners for intrusion detection in industrial control networks," *Big Data Cogn. Comput.*, vol. 5, no. 4, p. 72, 2021.

[15] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 1104–1116, Mar. 2021.

[16] M. H. L. Louk and B. A. Tama, "Revisiting gradient boosting-based approaches for learning imbalanced data: A case of anomaly detection on power grids," *Big Data Cogn. Comput.*, vol. 6, no. 2, p. 41, 2022.

[17] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain, and A. Nawaz, "HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems," *IEEE Access*, vol. 7, pp. 89507–89521, 2019.

[18] S. Sengan, V. Subramaniyaswamy, V. Indragandhi, P. Velayutham, and L. Ravi, "Detection of false data cyber-attacks for the assessment of security in smart grid using deep learning," *Comput. Elect. Eng.*, vol. 93, Jul. 2021, Art. no. 107211.

[19] D. Wilson, Y. Tang, J. Yan, and Z. Lu, "Deep learning-aided cyber-attack detection in power transmission systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, 2018, pp. 1–5.

[20] B. Zhang, Y. Yu, and J. Li, "Network intrusion detection based on stacked sparse autoencoder and binary tree ensemble method," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2018, pp. 1–6.

[21] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3104–3113, Nov. 2015.

[22] U. Adhikari, T. H. Morris, and S. Pan, "Applying non-nested generalized exemplars classification for cyber-power event and intrusion detection," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 3928–3941, Sep. 2018.

[23] D. K. Sadhasivan and K. Balasubramanian, "A novel LWCSO-PKM-based feature optimization and classification of attack types in SCADA network," *Arab. J. Sci. Eng.*, vol. 42, no. 8, pp. 3435–3449, 2017.

[24] U. Adhikari, T. H. Morris, and S. Pan, "Applying Hoeffding adaptive trees for real-time cyber-power event and intrusion classification," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4049–4060, Sep. 2018.

[25] C. Hu, J. Yan, and X. Liu, "Adaptive feature boosting of multi-sourced deep Autoencoders for smart grid intrusion detection," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, 2020, pp. 1–5.

[26] M. N. Kurt, O. Ogundijo, C. Li, and X. Wang, "Online cyber-attack detection in smart grid: A reinforcement learning approach," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5174–5185, Sep. 2019.

[27] F. Wei, Z. Wan, and H. He, "Cyber-attack recovery strategy for smart grid based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2476–2486, May 2020.

[28] D. An, Q. Yang, W. Liu, and Y. Zhang, "Defending against data integrity attacks in smart grid: A deep reinforcement learning-based approach," *IEEE Access*, vol. 7, pp. 110835–110845, 2019.

[29] W. Wang et al., "Abnormal flow detection in industrial control network based on deep reinforcement learning," *Appl. Math. Comput.*, vol. 409, Nov. 2021, Art. no. 126379.

[30] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[31] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. New York, NY, USA: Springer, 2012.

[32] U. Adhikari, T. Morris, and S. Pan, "WAMS cyber-physical test bed for power system, cybersecurity study, and data mining," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2744–2753, Nov. 2017.

[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[34] M. Nicolau, J. McDermott, and V. L. Cao, "A hybrid autoencoder and density estimation model for anomaly detection," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2016, pp. 717–726.

[35] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proc. MLSDA Proc. 2nd Workshop Mach. Learn. Sensory Data Anal.*, 2014, pp. 4–11.

[36] Z. Liu and W. Li, "A method of SVM with normalization in intrusion detection," *Procedia Environ. Sci.*, vol. 11, pp. 256–262, Dec. 2011.

[37] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1, 1995, pp. 278–282.

[38] C. Hu, J. Yan, and C. Wang, "Advanced cyber-physical attack classification with extreme gradient boosting for smart transmission grids," in *Proc. IEEE Power Energy Soc. General Meeting (PESGM)*, 2019, pp. 1–5.

[39] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[40] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, Jun. 2014, pp. 387–395. [Online]. Available: https://proceedings.mlr.press/v32/silver14.html

[41] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," 2015, *arXiv:1507.06527*.

[42] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

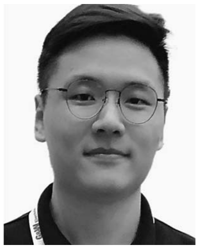
[43] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Security*, vol. 14, no. 1, pp. 1–33, 2011.

[44] T. H. Morris and W. Gao, "Industrial control system cyber attacks," in *Proc. 1st Int. Symp. ICS SCADA Cyber Security Res.*, 2013, pp. 22–29.

[45] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of Industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6822–6834, Aug. 2019.

[46] M. Zolanvari, Z. Yang, K. Khan, R. Jain, and N. Meskin, "TRUST XAI: Model-agnostic explanations for AI with a case study on IIoT security," *IEEE Internet Things J.*, early access, Oct. 21, 2021, doi: 10.1109/JIOT.2021.3122019.

[47] M. Zolanvari, A. Ghubaish, and R. Jain, "ADDAI: Anomaly detection using distributed AI," in *Proc. IEEE Int. Conf. Netw., Sens. Control (ICNSC)*, vol. 1, 2021, pp. 1–6.



Chengming Hu (Graduate Student Member, IEEE) received the M.Sc. degree in quality systems engineering from the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada, in 2019. He is currently pursuing the Ph.D. degree with McGill University, Montreal.

His research interests include machine learning with applications in cyber-physical system security and wireless communication systems.

Jun Yan (Member, IEEE) received the B.Eng. degree in information and communication engineering from Zhejiang University, China, in 2011, and the M.Sc. and Ph.D. degrees (with Excellence in Doctoral Research) in electrical engineering from the University of Rhode Island, USA, in 2013 and 2017, respectively.

He is currently an Associate Professor with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canad, where he is also a Founding Member of the Security Research Institute and a member of the Applied Artificial Intelligence Institute. His research interest includes computational intelligence and cyber–physical security, with applications in smart grids, smart cities, and other smart critical infrastructures. He was the recipient of the Best Paper Award of IEEE ICC, the Best Student Paper Award of IEEE WCCI, and the Best Readings of IEEE ComSoc.

Xue Liu (Fellow, IEEE) received the B.S. degree in mathematics and the M.S. degree in automatic control from Tsinghua University, Beijing, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2006.

He is currently a Professor and a William Dawson Scholar with the School of Computer Science, McGill University, Montreal, QC, Canada. He was also the Samuel R. Thompson Associate Professor with the University of Nebraska-Lincoln and HP Labs, Palo Alto, CA, USA. He has been granted one U.S. patent and filed four other U.S. patents, and published more than 150 research papers in major peer-reviewed international journals and conference proceedings. His research interests include computer networks and communications, smart grid, real-time and embedded systems, cyber-physical systems, data centers, and software reliability. He was a recipient of several awards, including the Year 2008 Best Paper Award from the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and the First Place Best Paper Award of the ACM Conference on Wireless Network Security in 2011. He received the Outstanding Young Canadian Computer Science Researcher Prizes from the Canadian Association of Computer Science.

Authorized licensed use limited to: Southeast University. Downloaded on July 18,2023 at 13:01:22 UTC from IEEE Xplore. Restrictions apply.