

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

10-2019

Detecting cyberattacks in industrial control systems using online learning algorithms

Guangxia LI

Yulong SHEN

Peilin ZHAO

Xiao LU

Jia LIU

See next page for additional authors

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Databases and Information Systems Commons](#), and the [Theory and Algorithms Commons](#)

Citation

LI, Guangxia; SHEN, Yulong; ZHAO, Peilin; LU, Xiao; LIU, Jia; LIU, Yangyang; and HOI, Steven C. H..
Detecting cyberattacks in industrial control systems using online learning algorithms. (2019).

Neurocomputing. 364, 338-348.

Available at: https://ink.library.smu.edu.sg/sis_research/5132

This Journal Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Author

Guangxia LI, Yulong SHEN, Peilin ZHAO, Xiao LU, Jia LIU, Yangyang LIU, and Steven C. H. HOI

Detecting Cyberattacks in Industrial Control Systems Using Online Learning Algorithms

Guangxia Li, Yulong Shen, Peilin Zhao, Xiao Lu, Jia Liu, Yangyang Liu,
Steven C. H. Hoi

Abstract

Industrial control systems are critical to the operation of industrial facilities, especially for critical infrastructures, such as refineries, power grids, and transportation systems. Similar to other information systems, a significant threat to industrial control systems is the attack from cyberspace—the offensive maneuvers launched by “anonymous” in the digital world that target computer-based assets with the goal of compromising a system’s functions or probing for information. Owing to the importance of industrial control systems, and the possibly devastating consequences of being attacked, significant endeavors have been attempted to secure industrial control systems from cyberattacks. Among them are intrusion detection systems that serve as the first line of defense by monitoring and reporting potentially malicious activities. Classical machine-learning-based intrusion detection methods usually generate prediction models by learning modest-sized training samples all at once. Such approach is not always applicable to industrial control systems, as industrial control systems must process continuous control commands with limited computational resources in a nonstop way. To satisfy such requirements, we propose using online learning to learn prediction models from the controlling data stream. We introduce several state-of-the-art online learning algorithms categorically, and illustrate their efficacies on two typically used testbeds—power system and gas pipeline. Further, we explore a new cost-sensitive online learning algorithm to solve the class-imbalance problem that is pervasive in industrial intrusion detection systems. Our experimental results indicate that the proposed algorithm can achieve an overall improvement in the detection rate of cyberattacks in industrial control systems.

I. INTRODUCTION

Modern industrial control systems are microprocessor-equipped devices and associated communication networks used to monitor and operate physical equipment in the industrial environment. Such systems are designated to collect sensor measurements and operational data from the physical world, display information to human operators, perform decisions based upon the detected events, and issue control commands to the controlled equipment. The commands are used to turn on or off power switches, open or close hydraulic valves, adjust motor speed, shut down engines in emergencies, etc. Although such operations are routine, they are crucial in industrial processes as any misoperation can cause incidents that may lead to devastating consequences in terms of financial loss, acute health effects, or environmental impacts. Modern digit-controller-based industrial control systems exhibit many advantages compared to their predecessors such as mechanical-based and electromechanical-based systems in terms of performance, reliability, and cost. In fact, modern industrial control systems have been applied widely in practice, and are the de-facto standard configuration of almost every industrial sector.

Figure 1 shows a notional topology of an industrial control system. As shown, the sensors measure physical quantities (e.g., flow, pressure, speed) and convert them into signals that are transmitted to the controllers. The controllers process the sensor signals

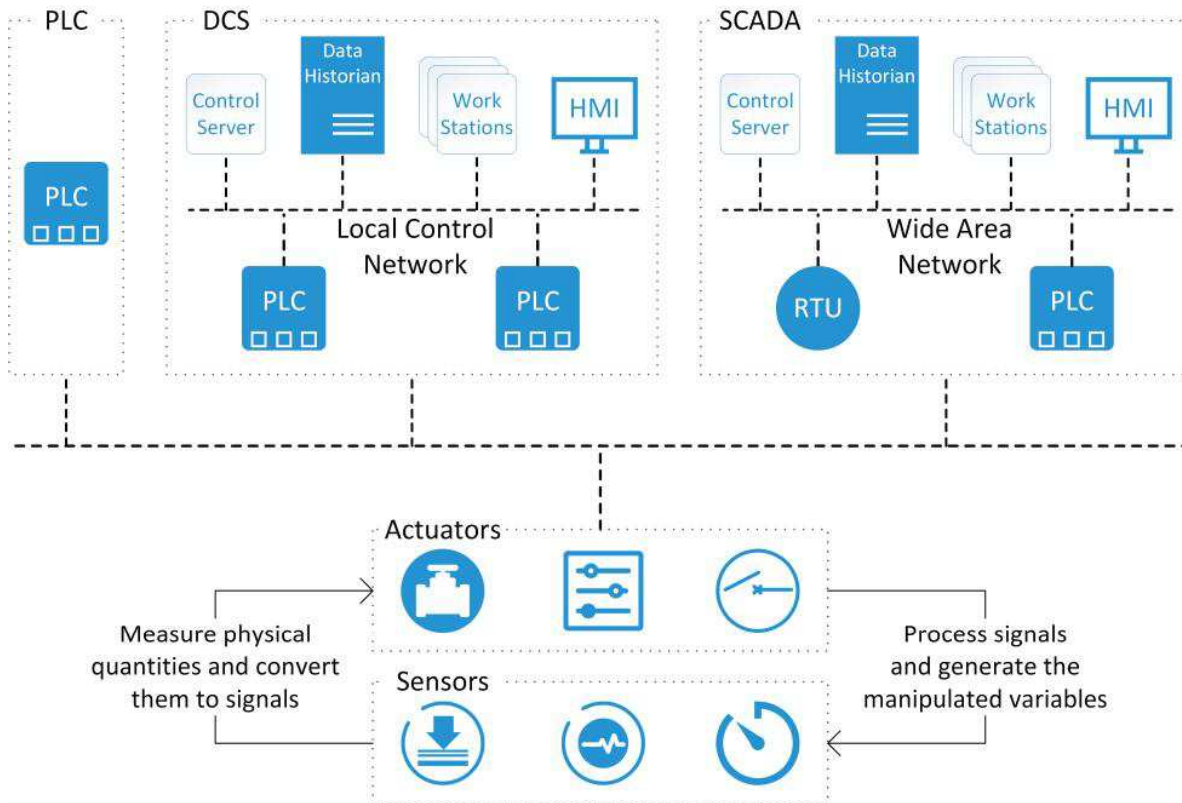


Fig. 1: Notional topology of an industrial control system.

to generate manipulated variables that are sent to the actuators (e.g., breakers, switches, valves) to manipulate the controlled process directly. Sensors, actuators, and controllers, together with some external components such as human machine interfaces (HMIs) and remote maintenance tools compose a typical industrial control system. Regarding the actual implementation of the system, many variants exist and their boundaries can be blurry. Still, there are several types of widely used control systems, such as supervisory control and data acquisition (SCADA) systems [1], distributed control systems (DCSs) [2], and programmable logic controllers (PLCs). Specifically, a SCADA system comprises a control center, and one or more geographically distributed field sites consisting of PLCs and/or remote terminal units (RTUs) used to command actuators and sensors. It is generally used to control geographically dispersed assets. A DCS, by contrast, is always applied to control production systems within a local area using the supervisory and regulatory control mechanism. As for PLCs, except for serving as the local controllers in SCADA and DCS configurations, they can also be implemented as the primary controllers in some smaller control systems to provide closed-loop control with no direct human involvement. For details about SCADA, DCS, PLC, and other types of control systems, refer to Stouffer et al. [3].

Industrial control systems are critical to the operation of industrial facilities, particularly to national critical infrastructures, such as refineries, chemical plants, electrical power grids, oil and natural gas pipelines, and transportation systems. Their incidents can cause significant risk to human lives and serious damage to the environment. Industrial control systems

had been thought to be immune to outsider threats because they were originally designed as isolated systems running proprietary control protocols using specialized hardware and software. This could be true in the past, but is no longer applicable nowadays. Modern industrial control systems do not operate in isolation anymore, but tend to be connected to wider networks (e.g., Internet, Internet-of-things [4], sensor networks [5], [6], smart grid systems [7]–[9], cloud systems [2], communication systems [10]–[13], corporate networks [14]–[17], and mobile social networks [18]). The proprietary protocols once unfamiliar to the public are gradually being replaced by open standards such as the Ethernet, TCP/IP [19], [20], and web services. The merging of typical information technologies into industrial control systems reduces the dubious protective barrier of “security by obscurity”, and thus increases the possibility of cybersecurity vulnerabilities and incidents [21]–[24]. In fact, cyberattacks to industrial control systems have occurred at an alarming pace in the last decade. Recent records include Stuxnet, Davis-Besse Nuclear Plant, Maroochy in Australia, Flame, and Aurora [3].

Among the recent cyberattacks is the famous Stuxnet worm known for its unverified but highly possible intent to compromise Iran’s nuclear program. Uncovered in 2010, Stuxnet is the first identified malware that targets SCADA systems [25]. It is believed to be introduced to Iran’s industrial sites via an infected USB flash drive. Subsequently, it propagates across the network using Microsoft Windows flaws. Stuxnet’s spread is indiscriminate, but its attack is designated to target only Siemens S7-300 PLC systems with particular variable frequency drives (VFD) attached. In particular, it monitors the frequency of motors controlled by VFD, and only attacks systems that spin between 807 Hz and 1210 Hz. The industrial applications of motors with these parameters include high-speed centrifuges that are essential for uranium enrichment. Stuxnet periodically modifies the frequency of VFD, and thus causes the rotational speed of connected motors to change in an unusual manner. It fakes the sensor signal to monitor systems, rendering the deed unbeknownst to human operators. Consequently, the fast-spinning centrifuges become destabilized initially and finally break down. Such sophisticated plots, together with the unprecedented complexity of the code, strongly suggest that Stuxnet is not a hacker’s sabotage, but a state-sponsored cyberattack. Because Stuxnet’s design is not domain specific, it could be tailored as a platform for attacking any SCADA or PLC system. This is proven by a number of new worms found subsequently, considered to be related to Stuxnet.

The recognition of Stuxnet has intensified the public’s awareness of the information security of industrial control systems. However, securing them is not easy. Owing to their long life span, legacy systems with great vulnerability are still active currently. It is not unusual for outdated devices without security patch to be manipulated by unalerted technicians in industrial sites. Because of the real-time, continuity, and constrained environment of the industrial control system, many methods used in traditional computer security (such as virus database update) are difficult to apply. As incidents and malicious actions are inevitable, detecting their occurrences timely is important to industrial control system administrators, and automatic devices such as firewalls and anti-virus software. This can be achieved by adapting solutions for traditional information technology environments to develop industrial-control-system-specific intrusion detection systems.

Artificial-intelligence-based approaches such as machine learning have been employed widely by intrusion detection systems. Among various machine learning methods, online learning represents a family of efficient algorithms that can build the predictor incrementally by processing the training data in a sequential manner, as opposed to batch learning

algorithms that train the predictor by learning the entire dataset all at once [26]. Specifically, online learning algorithms perform on a sequence of data by processing them one by one. On each round, the learner receives an input, makes a prediction using an internal hypothesis that is retained in memory, and subsequently learns the true label. It uses the new example to modify its hypothesis according to some predefined rules. The goal is to minimize the total number of rounds with incorrect predictions. In general, online learning algorithms are fast, simple, and require few statistical assumptions, rendering them applicable to a wide range of applications. They can scale well to a large amount of data, and are particularly suitable for real-world applications where data arrive continuously.

We herein study the problem of detecting cyberattacks in industrial control systems using online learning algorithms. We evaluate several state-of-the-art online learners in terms of their ability to identify malicious control commands from normal ones using testbeds provided by the Critical Infrastructure Protection Center of the Mississippi State University. The experimental results on a power system and a gas pipeline testbed indicate that online learning algorithms can discriminate the intrusions effectively. Furthermore, we focus on the so-called class-imbalanced problem that troubles most intrusion detection systems in real-world applications: relatively large numbers of normal events exist that can easily overwhelm a few attacks and distract classifiers. To address the class-imbalanced problem, we propose a cost-sensitive online learning method, namely the adaptive regularized cost-sensitive multiclass online learning, such that the classifier can focus on the minority classes that are more important. The proposed algorithm is a combination of the second-order online learning technique and the cost-sensitive learning approach. It differs from traditional multiclass online learners who are only concerned about the performance in terms of prediction mistake rate by taking the misclassification costs into consideration. We demonstrate experimentally that the proposed algorithm can discriminate attacks precisely and efficiently.

The remainder of this paper is organized as follows. Section II reviews the related work on industrial control system security. Section III begins with an introduction to classical online learning algorithms. Subsequently, our adaptive regularized cost-sensitive multiclass online learning algorithm is presented. Section IV gives experimental results and discussion. Section V concludes this paper.

II. RELATED WORK

In recent years, industrial control system security has garnered increasing attention, particularly that pertaining to critical infrastructures. Several academic studies have been performed to understand the characteristics of industrial control systems, analyze their vulnerabilities to cyberattacks, simulate real systems with testbeds, and demonstrate the importance of cybersecurity in industrial control systems. Cárdenas et al. [27] discussed the differences between the security of control systems and traditional IT systems, and analyzed the reasons for the current control systems being more vulnerable than before to cyberattacks. Companies, organizations, and government agencies are involved in the industrial control system security initiatives as well, primarily in the form of publishing guidelines, standards, and best practices. As an example, the U.S. National Institute of Standards and Technology (NIST) published a number of guidances to cybersecurity risk management, among which is the NIST Special Publication 800-82 providing cross-industry guidelines for securing industrial control systems [3]. This publication highlights the typical

threats and vulnerabilities to these systems and provides the recommended the security countermeasures to mitigate the associated risks. In addition to the guidance to cross industry, there are a number of publications targeting specific industries such as chemical, oil and gas. The surveys of such guidance, together with methodologies for measuring and managing threats, can be found in [28], [29].

Most of the conventional methods for protecting control systems have focused on increasing their reliability and maintainability. However, an urgent growing concern has emerged for protecting control systems against attacks launched in cyberspace. To detect such attacks, one can rely on the profile of anomaly patterns [30]–[33]. As an example, Carcano et al. [30] proposed an intrusion detection method for SCADA system based on tracking the so-called critical states that correspond to dangerous or unwanted situations in the monitored system. Their approach assumed that cyberattacks are always performed by forcing a transition of the system from a safe state to a critical state. Because the critical states are generally well known and limited in number, one can enumerate them formally beforehand and predict the criticality by tracking the changes in the distance between the current system state and the critical states. Likewise, Pan et al. [31] processed a sequence of critical system states using a sequential pattern mining algorithm to detect disturbances and cyberattacks in power systems. In contrast to exploiting the abnormal patterns, one can also specify the acceptable behaviors of a system, and subsequently detect attacks that cause violation to them [34], [35]. The so-called specification-based intrusion detection approach monitors the system according to policies specified by valid sequences of system behaviors. Any sequence of behaviors outside the predefined specifications is regarded as an abnormal behavior. As a representative work, Cheung et al. [34] presented three model-based detection techniques for monitoring SCADA networks: (1) specifying the expected characteristics of network request/response according to the Modbus protocol; (2) defining the expected communication patterns among network components; and (3) detecting the changes in server/service availability.

The aforementioned rule-based intrusion detection methods generally rely on human efforts to transform expert knowledge into machine-executable rules. Manually constructing such rules, however, can be a laborious and expensive endeavor. Machine learning based methods prevail in this situation as they can automatically generate rules from the existing examples without human efforts. For performance testing, Beaver et al. [36] evaluated several classical machine learning algorithms including the decision tree, naïve Bayes classifier, and support vector machine (SVM) in terms of their ability to identify cyberattacks using a dataset of RTU communications in a gas pipeline system. A similar evaluation was performed on a power system to demonstrate the feasibility of applying machine learning algorithms to discriminate types of power system disturbances, especially those caused by malicious attacks [37]. Terai et al. [38] built a discriminant model between a normal operation and attack packets on a laboratorial fluid system equipped with an actual industrial controller using SVM with packet transmission intervals and length as features. Schuster et al. [39] applied the one-class classification technique to implement a self-configuring anomaly detection for industrial network data. They identified the one-class SVM as a promising learning method for this task, as no sample of attacks or other anomalous traffic is required to construct the training set. Other classical intrusion detection methods employing machine learning techniques include exemplar-based classifier (e.g., [40]), k-nearest neighbors (e.g., [41], [42]), neural network (e.g., [43], [44]), SVM (e.g., [43]–[45]),

and naïve Bayes (e.g., [46]). Sommer et al. [47] identified a few unique challenges and corresponding guidelines regarding the use of machine learning for anomaly detection in a general network environment. Mantere et al. [48] narrowed down the scope from general networks to industrial control systems, and argued that the diversity of network traffic, while prevails in general networks and tends to disturb machine learning algorithms, is decreased significantly in industrial control systems. They thus considered machine learning a promising tool for intrusion detection in industrial control systems.

III. ONLINE LEARNING FOR INTRUSION DETECTION

A. Problem Setting

We tackle the intrusion detection problem with supervised machine learning techniques. Specifically, the task of detecting malicious actions from normal actions can be cast as a binary classification problem, in which we use a positive class to denote the malicious actions, and a negative class for the normal actions. To further distinguish malicious action types, the task can be transformed into a multiclass classification problem. To learn a classifier with machine learning techniques, a training set consisting of samples whose class labels are known is required. The training set consisting of samples of attributes and associated class labels is used to build a classification model, which is subsequently applied to samples with unknown class labels. A learning algorithm is employed to build a model that estimates the relationship between the attributes and class label of the training data. The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of samples that it has never seen.

Several well-known machine learning algorithms such as the decision tree, neural network, and SVM belong to the batch learning paradigm which assumes that all training samples are available before the learning process occurs. In contrast to batch learning, online learning algorithms operate on a stream of data by deciding the present instance based on past knowledge together with the latest available information. Formally, at each step t , the learner receives an incoming sample (\mathbf{x}_t, y_t) , where $\mathbf{x}_t \in \mathbb{R}^d$ is a d -dimensional vector representing the data, and y_t refers to its class label. For binary classification, $y_t \in \{-1, 1\}$; for multiclass classification, $y_t \in \{1, \dots, k\}$ and $k \geq 3$. The classification model to learn is parameterized by a weight vector $\mathbf{w} \in \mathbb{R}^d$. The learner first predicts the class label of the incoming instance as \hat{y}_t according to some criterion $\sigma(\mathbf{w} \cdot \mathbf{x})$. After the prediction, the true label y_t is revealed. It subsequently computes the loss $\ell(\hat{y}_t, y_t)$ according to the difference between the prediction \hat{y}_t and the revealed true label y_t , and updates the model by a certain strategy. The goal is to learn a model to minimize the online regret measured as the difference between the cumulative loss of the online learning algorithm and the cumulative loss of the best model, i.e., $\sum_{t=1}^T \ell(\sigma(\mathbf{w}_t \cdot \mathbf{x}_t), y_t) - \min_{\mathbf{w}} \sum_{t=1}^T \ell(\sigma(\mathbf{w} \cdot \mathbf{x}_t), y_t)$. Different updating strategies lead to different online learning algorithms. We elaborate some representative algorithms as follows.

B. Online Binary Classification

The seminal work of Frank Rosenblatt [49] proposed a simple model called perceptron. The perceptron operates by assigning weights to incoming connections. At each learning round, it takes the dot-product of each incoming value with a weight, and subsequently verifies if it is over or below a certain threshold. It compares the predicted label $\hat{y}_t =$

$\text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$ with the true label $y_t \in \{-1, 1\}$. If $\hat{y}_t \neq y_t$, the perceptron updates the model as $\mathbf{w}_{t+1} = \mathbf{w}_t + y_t \mathbf{x}_t$.

As improvements to perceptron-like algorithms, many modern online learning methods [50]–[53] have been proposed over the past decades. They are partly inspired by the maximum margin learning principle that has been applied successfully to batch mode learning. Specifically, for the incoming example (\mathbf{x}_t, y_t) and the algorithm’s weight vector \mathbf{w}_t , the term $y_t(\mathbf{w}_t \cdot \mathbf{x}_t)$ is referred to as the (signed) margin. Whenever the margin is a positive number, we say that the algorithm has predicted correctly. However, we are not satisfied with a positive margin value; we would prefer for the algorithm to predict correctly and with a larger margin. Therefore, our goal is to achieve a margin of at least 1, as often as possible. On rounds where the algorithm attains a margin less than 1, it suffers an instantaneous loss. Typically, this loss is defined by the following hinge-loss function:

$$\ell(\mathbf{w}; \mathbf{x}_t, y_t) = \max(0, 1 - y_t(\mathbf{w} \cdot \mathbf{x}_t)) \quad (1)$$

An example of such approach is the passive-aggressive (PA) algorithm [53]. In addition to employing the maximum margin principle, PA maintains a trade-off between the amount of progress achieved on each training round and the information retained from the previous rounds. On one hand, the classifier should be updated whenever it misclassifies a new instance. On the other hand, the classifier should not be changed too rapidly especially if it predicts most of the previous instances correctly. Formally, it is formulated as the following optimization problem:

$$\begin{aligned} \mathbf{w}_{t+1} &= \underset{\mathbf{w} \in \mathbb{R}^d}{\text{argmin}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 \\ \text{s.t.} \quad &\ell(\mathbf{w}; \mathbf{x}_t, y_t) = 0 \end{aligned}$$

where $\ell(\mathbf{w}; \mathbf{x}_t, y_t)$ is the hinge loss defined in Eq. (1).

In addition, some variants of PA are proposed to use the soft-margin technique to handle the non-separable and noisy cases. As an example, a variant named PA-I is formulated as follows:

$$\begin{aligned} \mathbf{w}_{t+1} &= \underset{\mathbf{w} \in \mathbb{R}^d}{\text{argmin}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 + C \mathbf{x}_i \\ \text{s.t.} \quad &\ell(\mathbf{w}; \mathbf{x}_t, y_t) \leq \mathbf{x}_i \quad \text{and} \quad \mathbf{x}_i \geq 0 \end{aligned}$$

where C is a positive parameter that controls the influence of the slack term \mathbf{x}_i on the objective function.

The online learning algorithms above belong to the family of first-order methods, as they only depend on the first-order information of the example. Additionally, the machine learning community has studied the second-order online learning algorithms that use parameter confidence information to guide the learning process. A family of confidence-weighted learning algorithms [54]–[56] assumes that the weight vector follows a Gaussian distribution $\mathbf{w} \sim \mathcal{N}(\mu, \Sigma)$ with mean vector $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$. To classify an instance \mathbf{x} , it draws a parameter vector $\mathbf{w} \sim \mathcal{N}(\mu, \Sigma)$ and predicts the label according to $\text{sign}(\mathbf{w} \cdot \mathbf{x})$. In practice, however, the average weight vector $E(\mathbf{w}) = \mu$ is used for the prediction. The model parameters, including both μ and Σ are updated appropriately with the effect of controlling the direction and scale of parameter updates. The learner performs online updates based on its confidence in the current parameters, generating larger changes

in the weights of infrequently observed features. Our empirical evaluation in the following indicates the advantages of learning with the second-order information.

As an example, the adaptive regularization of weight vectors (AROW) algorithm [56] maintains a probabilistic measure of confidence in each component of its weight vector using a Gaussian distribution. The weight distribution is updated by minimizing the Kullback–Leibler (KL) divergence between the new and old weight distributions under the constraint that the probability of correct classification is greater than a threshold. At round t , when receiving (\mathbf{x}_t, y_t) , the model is updated by minimizing the following objective:

$$\operatorname{argmin}_{\mu \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d \times d}} D_{KL}D(\mathcal{N}(\mu, \Sigma) \parallel \mathcal{N}(\mu_t, \Sigma_t)) + \frac{1}{2\gamma} \ell^2(\mu; (\mathbf{x}_t, y_t)) + \frac{1}{2\gamma} \mathbf{x}_t^T \Sigma \mathbf{x}_t$$

where $D_{KL}D(\mathcal{N}(\mu, \Sigma) \parallel \mathcal{N}(\mu_t, \Sigma_t))$ is the KL divergence.

The above minimization problem can be solved with a closed-form solution as in [56]. This makes AROW quite fast as it does not need to invoke any optimization routine for updating.

C. Online Multiclass Classification

The intrusion detection task occasionally requires the discrimination of attack type, instead of only distinguishing attacks from normal actions. Compared to the aforementioned binary classification, online multiclass classification operates over the same sequence of data samples $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$, but differs in that the choice of labels has more than two odds, i.e., $y_t \in \{1, \dots, k\}$ and $k \geq 3$. Unlike binary classification that represents the model with a weight vector $\mathbf{w} \in \mathbb{R}^d$, the multiclass model contains a $k \times d$ matrix \mathbf{W} , whose i th row can be considered as the model for the i th class. Specifically, the compound weight \mathbf{W} is composed as follows:

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_k \end{bmatrix} \quad (2)$$

Given an input \mathbf{x}_t , the online multiclass algorithm predicts the label as the index associated with the largest prediction value, i.e.,

$$\hat{y}_t = \operatorname{argmax}_{i \in [k]} \mathbf{W}_{t,i} \cdot \mathbf{x}_t$$

where $\mathbf{W}_{t,i}$ is the i th row of the matrix \mathbf{W} as shown in Eq. (2).

After generating a prediction, the learner computes the loss based on the true label y_t and the highest-ranked irrelevant label as follows:

$$\ell_{mc}(\mathbf{W}_t; \mathbf{x}_t, y_t) = \max(0, 1 - (\mathbf{W}_{t,y_t} \cdot \mathbf{x}_t - \max_{i \in [k], i \neq y_t} \mathbf{W}_{t,i} \cdot \mathbf{x}_t))$$

The learner subsequently updates the model parameter \mathbf{W}_t based on certain rules as those described in [57], [58].

D. Cost-Sensitive Online Learning for Binary Classification

A significant trait of the intrusion detection task is the skewed distribution of classes. In most cases, most of the classes are normal events. The misclassification costs of instances

from different classes can be significantly different. The classical online learning algorithms minimize the regret, or equally, maximize the accuracy. However, pursuing a maximal accuracy may be inappropriate on imbalanced datasets because a trivial learner that simply classifies all samples as negative could achieve a high accuracy, but is of little use in practice. This renders it unsuitable for class-imbalanced datasets.

As a remedy for class-imbalanced datasets, the cost-sensitive classification differs from the normal classification approach by considering the misclassification costs during the training process [59]. In principle, a better performance might be obtained if the classifier is tailored by the learning algorithm using the cost matrix. Over the past decades, substantial research efforts have been devoted to developing cost-sensitive classification algorithms. In online learning, cost-sensitive classification methods exist for the binary class case [57], [58], [60]. The key is to change the hinge loss in Eq. (1) to incorporate cost-sensitive measures and optimize such measures directly.

As an example, the cost sensitive online gradient descent (CSOGD) algorithm [60] is proposed to maximize the sum of weighted sensitivity (the proportion of positive samples that are identified correctly as such) and specificity (the proportion of negatives that are identified correctly as such). Hence, it modifies the hinge loss function as follows:

$$\ell(\mathbf{w}; \mathbf{x}_t, y_t) = \max(0, (\rho * I_{y=1} + I_{y=-1}) - y_t(\mathbf{w} \cdot \mathbf{x}_t)) \quad (3)$$

where ρ is a predefined parameter related to the ratio of the number of negative samples to the number of positive samples, and I is an indicator function.

The model is updated by:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \lambda \nabla \ell(\mathbf{w}_t)$$

where λ is a learning rate parameter and $\nabla \ell(\mathbf{w}_t)$ is the gradient of loss function in Eq. (3)

E. Cost-Sensitive Online Learning for Multiclass Classification

Despite being studied extensively for binary classification problems, cost-sensitive online learning has been rarely examined for the multiclass case, even though the imbalanced class distribution prevails in real-world applications. This particularly applies to industrial control systems where the attacks, with various subclasses, comprise only a small part of all events. As the sensitivity, specificity, and other class-sensitive metrics are defined for binary classification, the aforementioned cost-sensitive learning technique cannot be applied to the multiclass case. We thus propose a cost-sensitive online learning algorithm that can solve the multiclass classification problem.

Suppose that there are k classes. We use a $k \times d$ matrix \mathbf{W} as defined in Eq. (2) to represent the model. To define the cost of misclassification, we use a $k \times k$ matrix C in which the diagonal elements represent the cost of correct prediction (they are set to zero), and the off-diagonal elements $c(i, j) > 0, i, j \in \{1, 2, \dots, k\}, i \neq j$ denote the cost of misclassifying a sample of the i th class to the j th class. Given an example \mathbf{x} , we define the most possible misclassified class as follows:

$$p = \operatorname{argmax}_{i \in [k], i \neq y_t} \mathbf{W}_{t,i} \cdot \mathbf{x}_t$$

The loss on this example is defined as follows:

$$\ell_{mc}(\mathbf{W}_t; \mathbf{x}_t, y_t) = \max(0, c(y_t, p) - (\mathbf{W}_{t,y_t} \cdot \mathbf{x}_t - \mathbf{W}_{t,p} \cdot \mathbf{x}_t)) \quad (4)$$

Algorithm 1 Adaptive regularized cost-sensitive multiclass online learning (ARCSMC).

- 1: **Input:** A sequence of samples (\mathbf{x}_t, y_t) where $\mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \{1, \dots, k\}$ and $t \in \{1, \dots, T\}$; a cost matrix C ; a parameter $\gamma > 0$
 - 2: **Output:** A vector $\mathbf{W} \in \mathbb{R}^{k \times d}$ representing the compound model
 - 3: **Initialize:** Set $\mathbf{W}_0 = \mathbf{0}$, $\Sigma_0 = I$
 - 4: **for** $t = 1, \dots, T$ **do**
 - 5: Receive an instance \mathbf{x}_t
 - 6: Make the prediction as $\hat{y}_t = \operatorname{argmax}_{i \in [k]} \mathbf{W}_{t,i} \cdot \mathbf{x}_t$
 - 7: Receive the true label y_t
 - 8: Compute the most possible misclassified class as $p = \operatorname{argmax}_{i \in [k], i \neq y_t} \mathbf{W}_{t,i} \cdot \mathbf{x}_t$
 - 9: Compute the loss as $\ell_t = \max(0, c(y_t, p) - (\mathbf{W}_{t,y_t} \cdot \mathbf{x}_t - \mathbf{W}_{t,p} \cdot \mathbf{x}_t))$
 - 10: **if** $\ell_t > 0$ **then**
 - 11: Update the model as $\mathbf{W}_{t+1,y_t} = \mathbf{W}_{t,y_t} + \alpha_t \Sigma_t y_t \mathbf{x}_t$, and $\mathbf{W}_{t+1,p} = \mathbf{W}_{t,p} - \alpha_t \Sigma_t y_t \mathbf{x}_t$,
 where $\Sigma_{t+1} = \Sigma_t - \beta_t \Sigma_t \mathbf{x}_t \mathbf{x}_t^\top \Sigma_t$, $\alpha_t = \ell_t \beta_t$, and $\beta_t = \frac{1}{\mathbf{x}_t^\top \Sigma_t \mathbf{x}_t + \gamma}$
 - 12: **end if**
 - 13: **end for**
-

where $c(y_t, p)$ is an element extracted from the predefined cost matrix C .

Replacing the loss function of any online learning algorithms with Eq. (4) leads to a multiclass cost-sensitive online learning algorithm. We specifically employ the adaptive regularization of weights (AROW) [56] as the framework to derive a new algorithm, namely the adaptive regularized cost-sensitive multiclass online learning (ARCSMC). This is summarized in Algorithm 1. The evaluation of the ARCSMC algorithm is presented in the next section.

IV. EXPERIMENTAL RESULTS

It is difficult to conduct the security experiment on real industrial control systems because of the potential risk and downtime of services provided by the facilities controlled by them. An alternative method is to simulate their functions in an isolated environment, also known as the testbed, where experiments can be performed safely. In our experiment, we used two testbeds to evaluate the performance of online learning algorithms. We start by introducing our experimental setup, followed by discussion on the results.

A. Experimental Testbeds

The data used in our experiment are extracted from two testbeds developed by the Mississippi State University's Critical Infrastructure Protection Center.

a) Power System Dataset: The modern power transmission system, also known as the smart grid, relies on field sensors such as synchrophasors for remote monitoring and controlling. The synchrophasor data contain measurements such as voltage and current phasor, as well as the status of system devices including relays, breakers, and transformers. It is typically sampled at a high speed (e.g., 120 times per second [32]) and sent to a processing unit with low latency. Such a configuration causes the system to generate a large volume of data that demands real-time processing—an ideal scenario for applying online learning algorithms.

Table 1: Statistics of datasets used in experiment.

(a) Binary Datasets

	#Class	#Feature	#Sample	#Pos Sample (Attack Event)	#Pos:#Neg
Power System	2	128	78,377	55,663	2.45:1
Gas Pipeline	2	17	274,628	60,048	1:3.57

(b) Multiclass Dataset

	#Class	#Feature	#Sample	Class Ratios
Power System	3	128	78,377	Attack, Natural, No Event 71.02%, 23.36%, 5.62%

Table 1: Statistics of datasets used in experiment.

We used a testbed to explore the suitability of applying online learning methods to discriminate malicious activities from natural power system disturbances. The dataset includes the simulation of 37 event scenarios including natural disturbances (8 events), normal operations (1 event), and cyberattacks (28 events) in a two-line three-bus power system [37]. Two classification schemes are employed in the experiment: one is a binary classification where the 37 event scenarios are grouped as either the attack (28 events) or normal operation (9 events); the other is a three-class classification sharing the same setting as mentioned above. There are 78,377 samples in the dataset. Each sample consists of 128 features: 116 measurements are generated by 4 synchrophasors, and 12 measurements are from the control panel logs, relay logs, and Snort alerts (Snort is a network monitoring tool). The sample size, feature count, and class distributions of this dataset are summarized in Table 1.

b) Gas Pipeline Dataset: This dataset is a collection of labeled command/response streams from a simulated control system that models a gas pipeline used to transfer natural gas or other petroleum products [61], [62]. The physical system comprises a closed-loop gas pipeline connected to an air pump that pumps air into the pipeline, a manual release valve together with a solenoid release valve used to release air pressure from the pipeline, and a pressure sensor. Commercial PLC, RTU, and HMI are configured to control the physical system to maintain a specific pipeline pressure value.

Artifacts of normal operations and cyberattacks are mixed randomly to compose the dataset. Four categories of cyberattacks are included: response injection, reconnaissance, denial of service, and command injection. The configuration details can be found in [61]. Unlike the setting above, our experiment on this dataset only involves a binary classification task. It consists of 274,628 samples, in which 60,048 samples are attack related. Table 2 provides the statistics of this dataset.

As shown in Table 1, a disproportionate number of classes exist in all datasets. For the Power System dataset, the number of positive samples representing attack events is larger than that of normal events. This deviates from the popular belief that attacks are rare in a system. However, it is noteworthy that for this dataset, the portion of each class is determined by the testbed’s creator during simulation. In fact, regardless of the class

outnumbers, the class imbalance problem is prevalent in real-world applications. Further, the algorithms to mitigate imbalanced classes do not rely on the meaning of a specific class. Therefore, we are confident that the experimental results on testbeds described herein are applicable to real-world industrial control applications.

B. Evaluation Metrics

We adopt the following metrics to evaluate the performance of online learning algorithms for the intrusion detection task.

a) Cumulative error rate: The cumulative error rate is the ratio of the number of mistakes made by an online learner over the number of samples received to date. Despite its extensive usage in online learning studies, the cumulative error rate is inept to measure class-imbalanced datasets, as it ignores the different costs of misclassifying different classes. In an extreme case, one can create a trivial classifier on a highly imbalanced dataset (i.e., blanket prediction of the majority class) that exhibits a low error rate but is in fact of little use.

b) Sensitivity: Sensitivity, or true positive rate, measures the proportion of positives that are identified correctly as such (e.g., the percentage of attacks identified correctly by the intrusion detection system). For a binary classification problem, let P denote the number of positive samples, and N the number of negative samples. Further, let TP , TN , FP , and FN denote the true positive, true negative, false positive, and false negative, respectively. The sensitivity can be calculated as follows:

$$sensitivity = \frac{TP}{P} = \frac{TP}{TP + FN}$$

c) Specificity: Specificity, or true negative rate, measures the proportion of negatives that are identified correctly as such.

$$specificity = \frac{TN}{N} = \frac{TN}{TN + FP}$$

It is noteworthy that the sensitivity and specificity, by their definitions, are only applicable to the binary classification test.

d) Weighted sum of sensitivity and specificity: The weighted sum of sensitivity and specificity (abbreviated as “sum” hereafter) is defined as follows:

$$sum = \eta_p \times sensitivity + \eta_n \times specificity$$

where $0 \leq \eta_p \leq 1$, $0 \leq \eta_n \leq 1$, and $\eta_p + \eta_n = 1$. As a cost-sensitive metric, the weighted sum is suitable for measuring a classifier’s performance on the class-imbalanced dataset. The higher the sum value, the better the classifier. When the η_p and η_n are both equal to 0.5, the sum becomes the well-known balanced accuracy. In our experiment, we set η_p and η_n to 0.5.

C. Benchmark Setup

The intent of our work is to establish a foundation for the application of online learning to intrusion detection in industrial control systems. The benchmarks selected are thus state-of-the-art online learning algorithms with distinctive features. We employ the SOL online learning library [63] in our experiment for its good accessibility and efficiency. As SOL

includes a number of online learning methods, only representative methods are reported herein.

Specifically, the first-order online learning algorithms used in our experiment are as follows:

- Perceptron: the classical online learning algorithm [49];
- ALMA: the approximate large margin algorithm [50];
- ROMMA: the relaxed online maximum margin algorithm [51];
- OGD: the online gradient descent algorithm [52];
- PA: the passive aggressive online learning algorithm [53];
- CSOGD: the cost sensitive online gradient descent algorithm [60].

The second-order online learning algorithms include the following:

- CW: the confidence-weighted learning algorithm [54];
- AROW: the adaptive regularization of weight vectors algorithm [56];
- SCW: the soft confidence weighted learning algorithm [64];
- ARCSOGD: the adaptive regularized cost-sensitive online gradient descent algorithm [60];
- ARCSMC: the adaptive regularized cost-sensitive multiclass online learning algorithm presented in Section III-E.

The experiment is conducted on a PC with a 2.4-GHz CPU and 8-GB RAM. The key parameters for each algorithm are chosen from a small range of values (i.e., 0.001, 0.01, 0.1, 1, 10, 100, 1000) on a validation set. The elements of the cost matrix described in Section III-E are set inversely proportional to each class count. We shuffled the sample order for each dataset randomly and repeated the experiment 10 times with new shuffles. The average results and corresponding standard deviations over 10 trials are reported in Table 2.

D. Evaluation of Binary Datasets

Table 2 reports the mean error rate, sum, sensitivity, and specificity, together with their standard deviations of different algorithms measured at the last learning round on the two binary datasets. Figure 2 depicts the variation in mean error rate and sum along the entire online learning process. The running-time for each algorithm, i.e., the total time (in seconds) consumed by updating the models, and generating the predictions is included in Table 2 as well. Our observations from these results are as follows.

First, the classification problem is complex as the error rates are high. This has been confirmed previously [36], [37], where the classifiers tend to make mistakes on the rare classes. As for the online learning algorithms evaluated in this experiment, their error rates are approximately 30% in most cases, not differing significantly from those of the batch models reported in [36], [37]. Considering the minimal time cost of online learning algorithms (e.g., they can process hundred thousands of samples in tens of microseconds with moderate computing resource), we conclude that online learning can detect cyberattacks of industrial control systems more effectively compared to conventional batch learning approaches.

Next, the sensitivity records are less than those of specificity. For example, the sensitivity against specificity of the ALMA algorithm is 4.62% against 95.56% on the Gas Pipeline dataset. This is to be expected, as for a class-imbalanced dataset, identifying samples of the rare class (positive class in this case) is more difficult than that of the majority class (negative class). A classifier trained under this setting is more likely to err on the positive

Table 2: Experimental results on the binary datasets showing the mean (in terms of percentage) and standard deviation (in parentheses) of different metrics over 10 trails.

(a) Power System

	Perceptron	ALMA	ROMMA	OGD	PA-I	CW	AROW	SCW	CSOGD	ARCSOGD
Error Rate	49.25 (9.43)	34.25 (5.28)	52.50 (7.91)	35.00 (5.00)	35.50 (5.37)	43.75 (4.29)	37.25 (6.61)	33.50 (4.44)	37.29 (0.09)	42.15 (0.17)
Sensitivity	42.50 (12.08)	10.00 (12.91)	38.33 (12.55)	10.83 (13.06)	10.00 (12.91)	35.00 (10.97)	15.83 (14.93)	8.33 (13.03)	32.91 (0.34)	45.50 (0.36)
Specificity	54.29 (9.79)	89.64 (7.23)	51.43 (10.54)	88.21 (7.15)	87.86 (7.38)	65.36 (6.74)	82.86 (8.55)	91.43 (3.45)	74.86 (0.21)	62.89 (0.19)
Sum	48.39 (9.74)	49.82 (6.29)	44.88 (7.74)	49.52 (6.09)	48.93 (6.33)	50.18 (4.89)	49.35 (7.71)	49.88 (6.62)	53.89 (0.10)	54.19 (0.20)
Time (s)	0.01	0.03	0.04	0.04	0.06	0.05	0.03	0.05	0.06	0.06

(b) Gas Pipeline

	Perceptron	ALMA	ROMMA	OGD	PA-I	CW	AROW	SCW	CSOGD	ARCSOGD
Error Rate	37.14 (7.06)	28.57 (2.55)	34.29 (4.49)	28.16 (3.16)	28.37 (2.96)	30.00 (10.97)	27.76 (2.92)	29.59 (4.54)	23.25 (0.03)	23.64 (0.03)
Sensitivity	15.38 (10.88)	4.62 (5.38)	17.69 (7.30)	4.62 (5.38)	5.38 (6.33)	8.46 (24.18)	2.31 (5.19)	10.77 (16.70)	36.08 (0.14)	36.69 (0.11)
Specificity	80.00 (7.94)	95.56 (3.26)	83.06 (5.15)	96.11 (3.75)	95.56 (3.51)	92.22 (23.64)	97.50 (2.76)	91.94 (10.26)	88.13 (0.06)	87.46 (0.04)
Sum	47.69 (7.41)	50.09 (2.87)	50.37 (4.76)	50.36 (3.35)	50.47 (3.49)	50.34 (0.80)	49.90 (3.44)	51.36 (4.93)	62.10 (0.05)	62.08 (0.05)
Time (s)	0.03	0.08	0.08	0.11	0.14	0.08	0.12	0.10	0.15	0.21

Table 2: Experimental results on the binary datasets showing the mean (in terms of percentage) and standard deviation (in parentheses) of different metrics over 10 trails.

samples, rendering the metric for measuring the correctly identified positive samples, i.e., the sensitivity record, low. Because an intrusion detection system focuses more on the percentage of real attacks that are correctly identified, one should focus on promoting the performance in terms of sensitivity in practice.

Finally, by examining the values of sensitivity, specificity, and their weighted sum, we discovered that the two cost-sensitive online learners, i.e., CSOGD and ARCSOGD, outperform the others that do not apply the cost-sensitive learning schema in most cases. This suggests that it is effective to apply cost-sensitive algorithms on the intrusion detection task for industrial control systems.

E. Evaluation of Multiclass Datasets

We further evaluate the performance of online learning algorithms for multiclass classification in terms of their error rates and cost-sensitive metrics. As metrics such as sensitivity and specificity are for the binary classification test, we adapted for the multiclass dataset by treating one class as positive and all other classes as negative. We calculated the cost-sensitive metrics for each class as such, and reported them in Table 3 and Figure 3.

Similar to the result of binary datasets, the cost-sensitive learning algorithm (ARCSMC) generally outperforms regular online learners in terms of the weighted sum of sensitivity and specificity. This again indicates the necessity of applying cost-sensitiveness for the

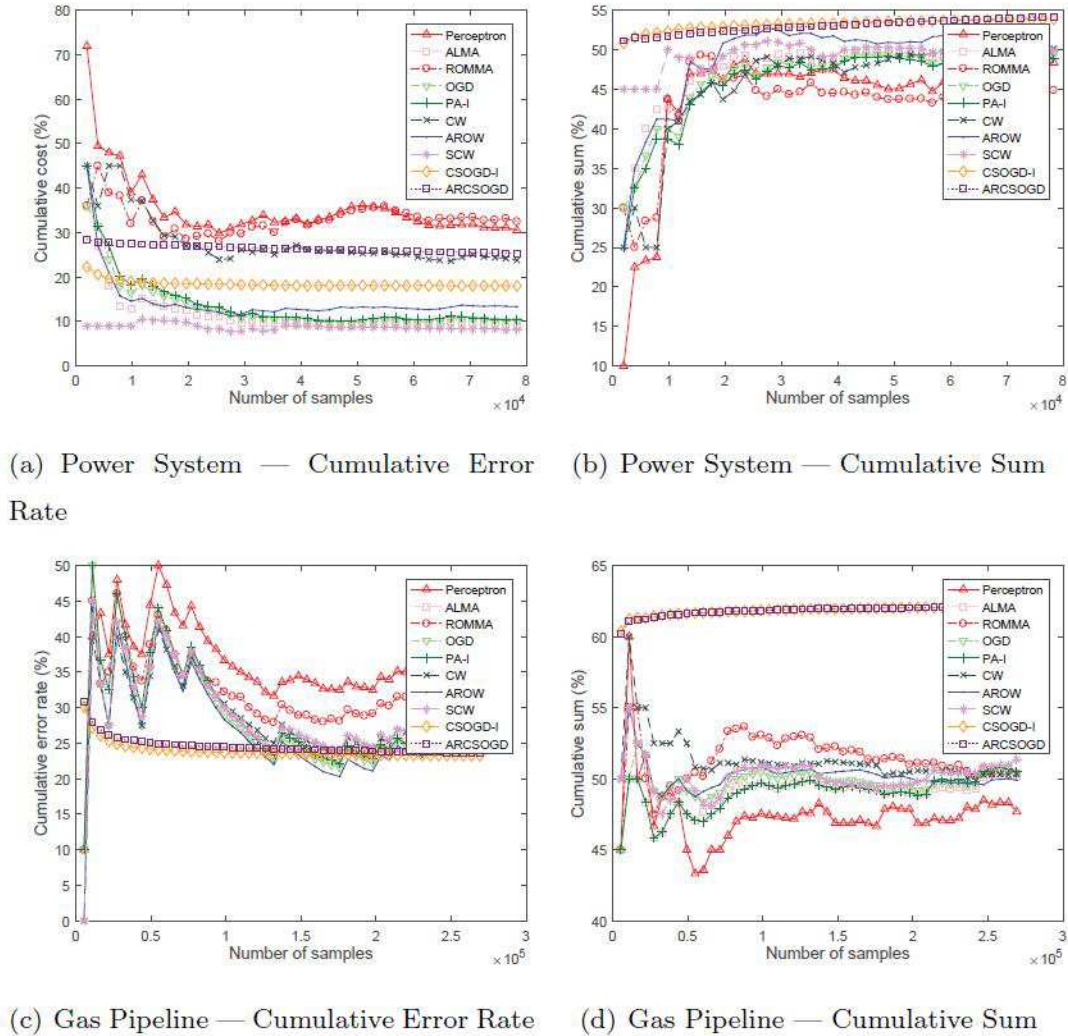


Fig. 2: The variation in evaluation metrics along the entire online learning process on the binary datasets.

class-imbalanced problem. As shown, SCW achieves the lowest error rate, at the cost of low sensitivity values on two minority classes. This indicates that SCW tends to classify all samples as the majority class. From a practical standpoint, such a classifier is not very helpful though it demonstrates a low error rate. Therefore, we conclude that ARCSMC is better than SCW when applied to detect the few but significant intrusion events in industrial control systems.

V. CONCLUSION

We herein explored the viability of applying online learning algorithms to perform intrusion detection in industrial control systems. We began by a brief review of the industrial control systems, the cyberthreats they experienced, and the intrusion detection methods, especially those based on machine learning techniques. We highlighted that because the industrial control systems require real-time response and uninterrupted operations, any

Table 3: Experimental results on the Power System multiclass dataset describing the mean (in terms of percentage) and standard deviation (in parentheses) of different metrics over 10 trails.

		Perceptron	ROMMA	OGD	PA-I	AROW	SCW	ARCSMC
Error Rate		50.89 (0.15)	53.59 (0.40)	31.13 (0.10)	31.46 (0.03)	32.45 (0.13)	29.26 (0.10)	40.48 (1.99)
Class 1 ratio: 5.62%	Sensitivity	27.59 (0.69)	34.90 (2.09)	1.25 (0.27)	1.30 (0.21)	10.28 (0.33)	34.07 (2.73)	89.40 (1.38)
	Specificity	82.84 (0.18)	79.47 (0.68)	97.62 (0.09)	97.34 (0.04)	96.26 (0.14)	97.86 (0.18)	78.44 (2.68)
	Sum	55.21 (0.34)	57.19 (0.91)	49.44 (0.12)	49.32 (0.10)	53.27 (0.18)	65.96 (1.29)	83.92 (0.93)
Class 2 ratio: 23.36%	Sensitivity	30.65 (0.30)	34.27 (0.53)	10.69 (0.18)	10.48 (0.14)	13.79 (0.18)	9.48 (0.61)	17.98 (1.22)
	Specificity	74.31 (0.16)	71.89 (0.47)	96.18 (0.11)	96.04 (0.06)	93.61 (0.14)	96.25 (0.23)	92.28 (0.73)
	Sum	52.48 (0.19)	53.08 (0.20)	53.43 (0.06)	53.26 (0.06)	53.70 (0.05)	52.87 (0.19)	55.13 (0.30)
Class 3 ratio: 71.02%	Sensitivity	56.89 (0.17)	51.31 (0.54)	93.36 (0.18)	92.96 (0.05)	89.77 (0.20)	93.79 (0.41)	70.82 (3.12)
	Specificity	48.23 (0.49)	56.28 (0.46)	10.46 (0.24)	10.60 (0.11)	17.13 (0.20)	15.91 (0.91)	50.98 (3.15)
	Sum	52.56 (0.29)	53.79 (0.20)	51.91 (0.05)	51.78 (0.05)	53.45 (0.12)	54.85 (0.27)	60.90 (0.17)

Table 3: Experimental results on the Power System multiclass dataset describing the mean (in terms of percentage) and standard deviation (in parentheses) of different metrics over 10 trails..

algorithms that they employ to detect attacks should be efficient, scalable, and suitable for processing data stream. Online learning algorithms satisfy only these requirements. We subsequently introduced several state-of-the-art online learning methods, and especially the cost-sensitive online classification that could be used to improve the prediction accuracy of the rare but significant attack events. We applied the cost-sensitive learning scheme to AROW [56] to derive a new method—the adaptive regularized cost-sensitive multiclass online learning (ARCSMC). The experimental results indicated that the cost-sensitive online learning algorithms, in particular the proposed ARCSMC, are both effective and efficient for detecting cyberattacks in industrial control systems.

For future work, we wish to extend our experiments to a more substantial size dataset and to more applications. This involves building new testbeds to mimic more industrial control processes and simulating more cyberattacks with different communication protocols and attack schemes. In addition, more online learning algorithms and classification schemes will be studied. One possible solution is the online one-class classification that is particularly suitable for problems where the majority of available data represents the normal behavior

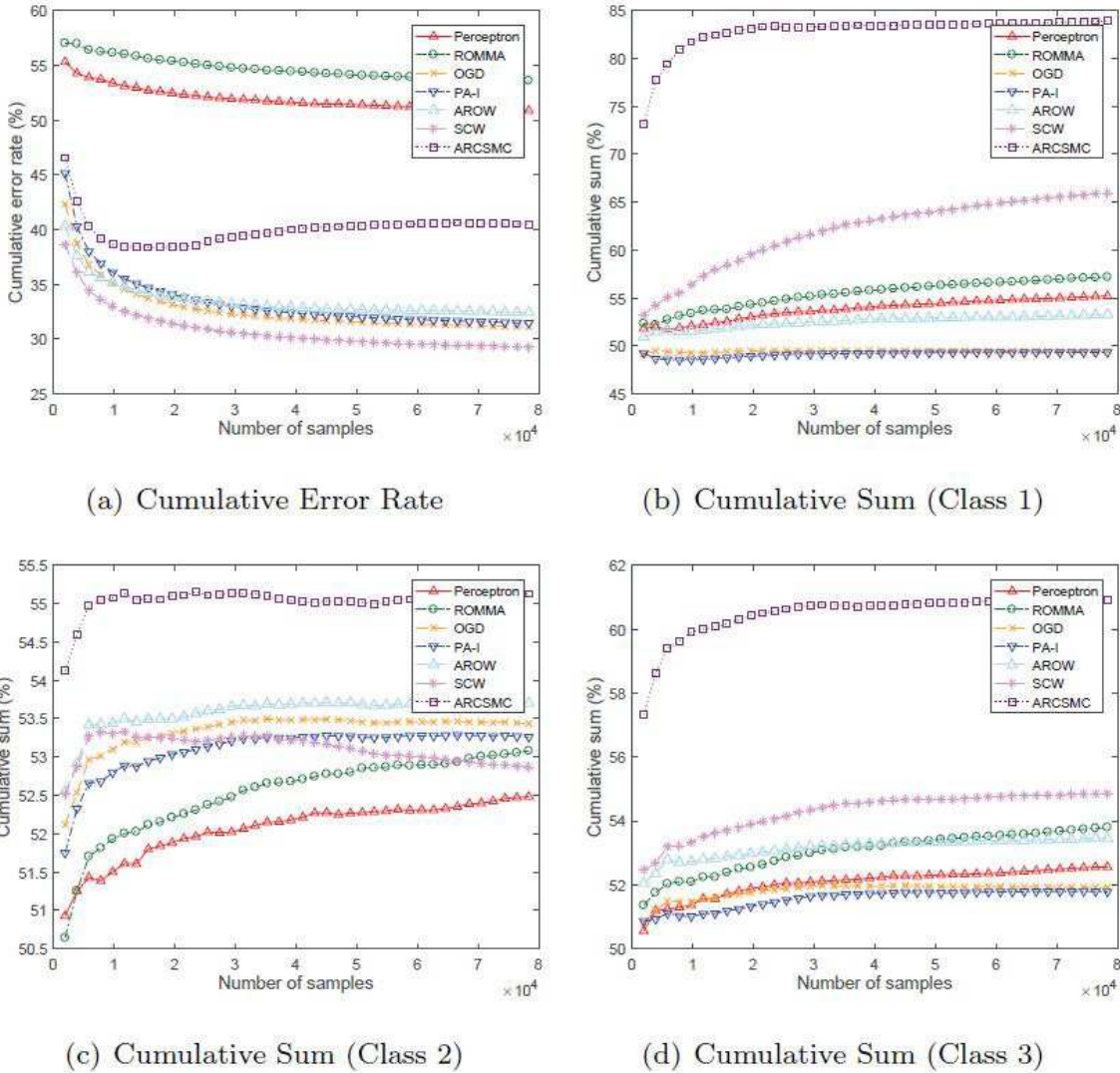


Fig. 3: The variation of evaluation metrics along the entire online learning process on the Power System multiclass datasets.

of the system, whereas the data related to attack events are difficult to obtain. In conclusion, our work serves as an initial attempt at applying online learning to detect cyberattacks in industrial control systems.

REFERENCES

- [1] D. Niyato, X. Lu, P. Wang, Machine-to-machine communications for home energy management system in smart grid.
- [2] G. Li, P. Zhao, X. Lu, J. Liu, Y. Shen, Data analytics for fog computing by distributed online learning with asynchronous update, in: ICC 2019-2019 IEEE International Conference on Communications (ICC), IEEE, 2019.
- [3] K. Stouffer, J. Falco, K. Scarfone, Guide to industrial control systems (ics) security, NIST special publication 800 (82) (2011).
- [4] D. Niyato, X. Lu, P. Wang, D. I. Kim, Z. Han, Economics of internet of things (iot): An information market approach, arXiv preprint arXiv:1510.06837.

- [5] D. Niyato, X. Lu, P. Wang, D. I. Kim, Z. Han, Distributed wireless energy scheduling for wireless powered sensor networks, in: 2016 IEEE International Conference on Communications (ICC), IEEE, 2016, pp. 1.
- [6] X. Lu, Sensor networks with wireless energy harvesting. (2016).
- [7] X. Lu, D. Niyato, P. Wang, 1 power management for wireless base station in smart grid environment: Modeling and optimization.
- [8] D. Niyato, X. Lu, P. Wang, Adaptive power management for wireless base stations in a smart grid environment, *IEEE Wireless Communications* 19 (6) (2012).
- [9] M. Korki, H. L. Vu, C. H. Foh, X. Lu, N. Hosseinzadeh, Mac performance evaluation in low voltage plc networks, *ENERGY* (2011) 135.
- [10] X. Lu, E. Hossain, T. Shafique, S. Feng, H. Jiang, D. Niyato, Intelligent reflecting surface (IRS)-enabled covert communications in wireless networks, arXiv preprint arXiv:1911.00986.
- [11] D. Niyato, P. Wang, D. I. Kim, Z. Han, L. Xiao, Game theoretic modeling of jamming attack in wireless powered communication networks, in: 2015 IEEE International Conference on Communications (ICC), IEEE, 2015.
- [12] X. Lu, E. Hossain, H. Jiang, G. Li, On coverage probability with type-ii harq in large-scale uplink cellular networks, *IEEE Wireless Communications Letters*.
- [13] X. Lu, D. Niyato, H. Jiang, D. I. Kim, Y. Xiao, Z. Han, Ambient backscatter assisted wireless powered communications, *IEEE Wireless Communications* 25 (2) (2018) pp. 170 - 177.
- [14] X. Lu, P. Wang, D. Niyato, Payoff allocation of service coalition in wireless mesh network: A cooperative game perspective, in: 2011 IEEE Global Telecommunications Conference-GLOBECOM 2011, IEEE, 2011.
- [15] X. Lu, P. Wang, D. Niyato, Hierarchical cooperation for operator-controlled device-to-device communications: A layered coalitional game approach, 545 in: 2015 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2015.
- [16] X. Lu, D. Niyato, H. Jiang, E. Hossain, P. Wang, Ambient backscatter-assisted wireless-powered relaying, *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 4, pp. 1087-1105, Dec. 2019.
- [17] X. Lu, G. Li, H. Jiang, D. Niyato, P. Wang, Performance analysis of wireless-powered relaying with ambient backscattering, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018.
- [18] Y. Zhang, D. Niyato, P. Wang, X. Lu, Optimizing content relay policy in publish-subscribe mobile social networks, in: 2015 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2015, pp. 2167-2172.
- [19] X. Lu, K. Zhang, C. P. Fu, C. H. Foh, A sender-side tcp enhancement for startup performance in high-speed long-delay networks, in: 2010 IEEE Wireless Communication and Networking Conference, IEEE, 2010, pp. 1-5.
- [20] X. Lu, K. Zhang, C. H. Foh, C. P. Fu, Ssthresholdless start: A sender-side tcp intelligence for long fat network, arXiv preprint arXiv: 1401.7146.
- [21] X. Lu, D. Niyato, H. Jiang, P. Wang, H. V. Poor, Cyber insurance for heterogeneous wireless networks, *IEEE Communications Magazine* 56 (6) (2018), pp. 21-27.
- [22] X. Lu, D. Niyato, N. Privault, H. Jiang, P. Wang, Managing physical layer security in wireless cellular networks: A cyber insurance approach, *IEEE Journal on Selected Areas in Communications* 36 (7) (2018) pp. 1648-1661.
- [23] X. Lu, D. Niyato, N. Privault, H. Jiang, S. S. Wang, A cyber insurance approach to manage physical layer secrecy for massive mimo cellular networks, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018.
- [24] D. Niyato, P. Wang, D. I. Kim, Z. Han, L. Xiao, Performance analysis of delay-constrained wireless energy harvesting communication networks under jamming attacks, in: 2015 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2015, pp. 1823-1828.
- [25] N. Falliere, L. O. Murchu, E. Chien, W32.stuxnet dossier, White paper, Symantec Corp., Security Response 5 (6).
- [26] S. C. H. Hoi, D. Sahoo, J. Lu, P. Zhao, Online learning: A comprehensive survey, *CoRR* abs/1802.02871. arXiv:1802.02871. URL <http://arxiv.org/abs/1802.02871>.
- [27] A. A. Cardenas, S. Amin, S. Sastry, Research challenges for the security of control systems, in: 3rd USENIX Workshop on Hot Topics in Security, HotSec'08, San Jose, CA, USA, July 29, 2008, Proceedings, 2008.
- [28] W. Knowles, D. Prince, D. Hutchison, J. F. P. Disso, K. Jones, A survey of cyber security management in industrial control systems, *IJCIP* 9 (2015) pp. 52-80.
- [29] D. Ding, Q. Han, Y. Xiang, X. Ge, X. Zhang, A survey on security control and attack detection for industrial cyber-physical systems, *Neurocomputing*, 275 (2018) pp. 1674-1683.
- [30] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, A. Trombetta, A multidimensional critical state analysis for detecting intrusions in SCADA systems, *IEEE Trans. Industrial Informatics* 7 (2) (2011) pp. 179-186.
- [31] S. Pan, T. H. Morris, U. Adhikari, Classification of disturbances and cyberattacks in power systems using heterogeneous time-synchronized data, *IEEE Trans. Industrial Informatics* 11 (3) (2015) pp. 650-662.
- [32] S. Pan, T. H. Morris, U. Adhikari, Developing a hybrid intrusion detection system using data mining for power systems, *IEEE Trans. Smart Grid* 6 (6) (2015) pp. 3104-3113.
- [33] W. Gao, T. H. Morris, On cyber attacks and signature based intrusion detection for MODBUS based industrial control systems, *JDFSL* 9 (1) (2014) pp. 37-56.
- [34] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, A. Valdes, Using model-based intrusion detection for scada networks, in: Proceedings of the SCADA security scientific symposium, Vol. 46, 2007.

- [35] R. Mitchell, I. Chen, Behavior-rule based intrusion detection systems for safety critical smart grid applications, *IEEE Trans. Smart Grid* 4 (3) (2013) pp. 1254-1263.
- [36] J. M. Beaver, R. C. Borges-Hink, M. A. Buckner, An evaluation of machine learning methods to detect malicious SCADA communications, in: 12th 610 International Conference on Machine Learning and Applications, ICMLA 2013, Miami, FL, USA, December 4-7, 2013, Volume 2, 2013, pp. 54-59.
- [37] R. C. B. Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, S. Pan, Machine learning for power system disturbance and cyber-attack discrimination, in: Resilient Control Systems (ISRCs), 2014 7th International Symposium on, IEEE, 2014.
- [38] A. Terai, S. Abe, S. Kojima, Y. Takano, I. Koshijima, Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile, in: 2017 IEEE European Symposium on Security and Privacy Workshops, EuroS & PWorkshops 2017, Paris, France, April 26-28, 2017.
- [39] F. Schuster, A. Paul, R. Rietz, H. Konig, Potentials of using one-class SVM for detecting protocol-specific anomalies in industrial networks, in: IEEE Symposium Series on Computational Intelligence, SSCI 2015, Cape Town, South Africa, December 7-10, 2015, 2015, pp. 83-90.
- [40] U. Adhikari, T. H. Morris, S. Pan, Applying non-nested generalized exemplars classification for cyber-power event and intrusion detection, *IEEE Trans. Smart Grid* 9 (5) (2018) pp. 3928-3941.
- [41] Y. Liao, V. R. Vemuri, Use of k-nearest neighbor classifier for intrusion detection, *Computers & Security* 21 (5) (2002) pp. 439-448.
- [42] K. Wang, S. J. Stolfo, Anomalous payload-based network intrusion detection, in: Recent Advances in Intrusion Detection: 7th International Symposium, RAID 2004, Sophia Antipolis, France, September 15-17, 2004. Proceedings, 2004, pp. 203-222.
- [43] S. Mukkamala, G. Janoski, A. Sung, Intrusion detection using neural networks and support vector machines, in: Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on, Vol. 2, IEEE, 2002, pp. 1702-1707.
- [44] W. Chen, S. Hsu, H. Shen, Application of SVM and ANN for intrusion detection, *Computers & OR* 32 (2005) pp. 2617-2634.
- [45] S. Mukkamala, A. H. Sung, A. Abraham, Intrusion detection using an ensemble of intelligent paradigms, *J. Network and Computer Applications* 28 (2) (2005) pp. 167-182.
- [46] L. Koc, T. A. Mazzuchi, S. Sarkani, A network intrusion detection system based on a hidden naive bayes multiclass classifier, *Expert Syst. Appl.* 645 39 (18) (2012) pp. 13492-13500.
- [47] R. Sommer, V. Paxson, Outside the closed world: On using machine learning for network intrusion detection, in: 31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berkeley/Oakland, California, USA, 2010, pp. 305-316.
- [48] M. Mantere, I. Uusitalo, M. Sailio, S. Noponen, Challenges of machine learning based monitoring for industrial control system networks, in: 26th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2012, Fukuoka, Japan, March 26-29, 2012, 2012, pp. 968-972.
- [49] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological review* 65 (6) (1958) pp. 386-408.
- [50] C. Gentile, A new approximate maximal margin classification algorithm, *Journal of Machine Learning Research* 2 (2001) pp. 213-242.
- [51] Y. Li, P. M. Long, The relaxed online maximum margin algorithm, *Machine Learning* 46 (1-3) (2002) pp. 361-387.
- [52] M. Zinkevich, Online convex programming and generalized infinitesimal gradient ascent, in: Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, 665 USA, 2003, pp. 928-936.
- [53] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer, Online passive-aggressive algorithms, *Journal of Machine Learning Research* 7 (2006) pp. 551-585.
- [54] M. Dredze, K. Crammer, F. Pereira, Confidence-weighted linear classification, in: Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008, 2008, pp. 264-271.
- [55] K. Crammer, M. Dredze, F. Pereira, Exact convex confidence-weighted learning, in: NIPS, 2008, pp. 345-352.
- [56] K. Crammer, A. Kulesza, M. Dredze, Adaptive regularization of weight vectors, in: Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada., 2009, pp. 414-422.
- [57] K. Crammer, Y. Singer, Ultraconservative online algorithms for multiclass problems, *Journal of Machine Learning Research* 3 (2003) pp. 951-991.
- [58] M. Fink, S. Shalev-Shwartz, Y. Singer, S. Ullman, Online multiclass learning by interclass hypothesis sharing, in: Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006, pp. 313-320.
- [59] C. Elkan, The foundations of cost-sensitive learning, in: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001, 2001, pp. 973-978.

- [60] P. Zhao, F. Zhuang, M. Wu, X. Li, S. C. H. Hoi, Cost-sensitive online classification with adaptive regularization and its applications, in: 2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015, 2015, pp. 649-658.
- [61] T. H. Morris, A. K. Srivastava, B. Reaves, W. Gao, K. Pavurapu, R. Reddi, A control system testbed to validate critical infrastructure protection concepts, *IJCIP* 4 (2) (2011) pp. 88-103.
- [62] T. H. Morris, Z. Thornton, I. Turnipseed, Industrial control system simulation and data logging for intrusion detection system research, 7th Annual Southeastern Cyber Security Summit.
- [63] Y. Wu, S. C. H. Hoi, C. Liu, J. Lu, D. Sahoo, N. Yu, SOL: A library for scalable online learning algorithms, *Neurocomputing* 260 (2017) pp. 9-12.
- [64] S. C. H. Hoi, J. Wang, P. Zhao, Exact soft confidence-weighted learning, in: Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012, 2012.