Hindawi
Security and Communication Networks
Volume 2019, Article ID 6816943, 15 pages
https://doi.org/10.1155/2019/6816943

Research Article

VHDRA: A Vertical and Horizontal Intelligent Dataset Reduction Approach for Cyber-Physical Power Aware Intrusion Detection Systems

Hisham A. Kholidy[1] **and Abdelkarim Erradi**[2]

[1]*Department of Network and Computer Security (NCS), College of Engineering, State University of New York (SUNY) Polytechnic Institute, 100 Seymour Rd., Utica, NY, USA*
[2]*Computer Science and Engineering Department, Qatar University, Doha, Qatar*

Correspondence should be addressed to Hisham A. Kholidy; hisham_dev@yahoo.com

Received 19 November 2018; Accepted 7 March 2019; Published 17 June 2019

Guest Editor: Rohit Ranchal

The Cypher Physical Power Systems (CPPS) became vital targets for intruders because of the large volume of high speed heterogeneous data provided from the Wide Area Measurement Systems (WAMS). The Nonnested Generalized Exemplars (NNGE) algorithm is one of the most accurate classification techniques that can work with such data of CPPS. However, NNGE algorithm tends to produce rules that test a large number of input features. This poses some problems for the large volume data and hinders the scalability of any detection system. In this paper, we introduce VHDRA, a Vertical and Horizontal Data Reduction Approach, to improve the classification accuracy and speed of the NNGE algorithm and reduce the computational resource consumption. VHDRA provides the following functionalities: (1) it vertically reduces the dataset features by selecting the most significant features and by reducing the NNGE's hyperrectangles. (2) It horizontally reduces the size of data while preserving original key events and patterns within the datasets using an approach called STEM, State Tracking and Extraction Method. The experiments show that the overall performance of VHDRA using both the vertical and the horizontal reduction reduces the NNGE hyperrectangles by 29.06%, 37.34%, and 26.76% and improves the accuracy of the NNGE by 8.57%, 4.19%, and 3.78% using the Multi-, Binary, and Triple class datasets, respectively.

1. Introduction

The CPPS are vital components that require special cybersecurity efforts. The deregulation and multipoint communication between consumers and utilities has introduced more complexities that make power system operation very difficult to manage. The WAMS plays an important role in monitoring and controlling of the CPPS since it provides large volume of information and an efficient communication infrastructure. However, this introduces cyber security vulnerabilities to these systems. Intruders may exploit such vulnerabilities to create cyberattacks against the electric power grid. The CPPS need to be resilient to cyberattacks through a precise and scalable attack classification technique that can deal with the large volume of high speed data provided by the WAMS and facilitate the autonomic control of the complex operation of the CPPS.

Several approaches have been proposed to secure the CPPS systems such as the behavior rule-based monitoring devices methodology [1] in the smart grid that is used to detect the insider threats, the anomaly detection techniques [2], which extract the normal behaviors from various communication protocols of Industrial Control Systems (ICSs) to create a full description of the communication pattern, The Specification-Based IDS [3] that monitors system security states and sends the alerts when the system behavior approaches an unsafe or disallowed state, the common path mining approach [4] that creates an IDS using heterogeneous data for detecting power system cyberattacks using the State Tracking and Extraction Method (STEM) algorithm [5] to

preprocess data and then uses frequent item set mining to extract common paths associated with specific system behaviors, and recently a NNGE with a Hoeffding Adaptive Trees approach [5, 6] that is used to create an offline and online Event Intrusion Detection Systems using STEM to process the power system security datasets. However, these approaches are still neither accurate nor scalable enough to process the high speed big data of the CPPS [5, 7–9]. To build an efficient security framework that well adapt the CPPS, the following security requirements are recommended by several well-established IT security organizations such as the Department of Homeland Security (DHS) [10], SANS [11], Check Point Software [12], and ENISA [13]: (1) the security framework should provide the required tools to perform the classification and detection of attacks, security assessment, and auditing processes in an accurate and fast way [11], (2) the security framework should be integrated with the current IT security solutions such as IDS to provide monitoring and log analysis capabilities, security for network communications, hosts, and control access to physical control systems [4], (3) the architecture of the security framework should be more scalable, robust, and flexible to deal with heterogeneous attacks and heterogeneous CPPS [5], and (4) the detection and response time of the security framework should be short enough to prevent attackers from completing their reconnaissance and/or installing any illicit monitoring or disrupting malware. According to these requirements, a scalable, accurate, and fast classification approach is required to work with the CPPS. The NNGE algorithm is among the most accurate classification techniques that can work with heterogeneous datasets formats such as the WAMS data [5, 7]. Although the accuracy of NNGE as a standalone classification method is lower [7, 14], it is still the most suitable classification method to develop a cypher physical power aware intrusion detection systems because of its ability to classify multiclass scenarios and sequential data and handle heterogeneous datasets formats such as discrete, nominal or symbolic, continuous, and nonvalue features [15–17]. In this work, we introduce a new data reduction approach called VHDRA, Vertical and Horizontal Data Reduction Approach, which includes feature selection, exemplar pruning, feature reduction, and system states compression methods. VHDRA improves the classification accuracy and speed and reduces the computational resource consumption of the NNGE algorithm. It provides the following functionalities.

(1) A vertical reduction for the dataset features by selecting the most significant features: to this target, we developed a new fitness function for the Particle Swarm Optimization (PSO) algorithm [18] that adopts the classification function of the NNGE algorithm through selecting the significant features whose values are closer to a margin of the covering hyper-rectangle.

(2) Pruning of nongeneralized exemplars using the highest ranked features of the PSO: VHDRA uses the Evolutionary Pruning Algorithms (EPA-NNGE) [19] to improve the classification accuracy of NNGE and to reduce the model size by reducing the hyperrectangles and ignoring the nonselected features among the significant ones defined by PSO.

(3) A horizontal reduction for the size of the dataset while preserving original key events and patterns within the datasets using an approach called STEM, State Tracking and Extraction Method [6], which is used to quantize and reduce the heterogeneous datasets to reduce STEM tracks system states from measurements and creates a compressed sequence of states for each observed scenario.

To evaluate the accuracy of the VHDRA, we compare the detection rates of the NNGE using VHDRA against current classification approaches including the NNGE with its best feature selection approach, namely, the Correlation based Feature Selection (CFS) [6]. The comparison uses an existing intrusion detection power grid dataset [15]. To evaluate the improvement in the detection speed and computational resource consumption, we compare the number of reduced exemplars using the NNGE with CFS, VHDRA with the horizontal reduction, VHDRA with the vertical reduction, and VHDRA with both the horizontal and the vertical reduction together. The real time implementation of VHDRA consists of the following three phases: (1) training and configuration phase. At this phase, the important features are extracted from the real time dataset using the modified PSO fitness function, and the k-fold cross-validation algorithm is applied to compute the dataset threshold. Furthermore, the quantization intervals of the STEM are created based on the dataset data boundaries. (2) Online detection phase: at this phase, the integrated PSO and Evolutionary Pruning NNGE (EPA-NNGE) approach is used to classify the online events and fires alerts when an attack is detected. (3) Update phase: at this phase, the quantization intervals and the STEM's states are updated and inserted into the quantized datasets and duplicated values are deleted. The threshold value is also updated and new behaviors and events are added to the dataset. This paper is organized as follows: after Section 1 introduces the NNGE algorithm, the CCPS testbed, and the test datasets, Section 2 surveys the state of the art of the previous attempts to improve the NGGE. After that, Section 3 introduces the improved NNGE algorithm and the VHDRA, then Section 4 discusses the experimental analysis of the results, and, finally, Section 5 concludes the paper and draws the furfure work.

1.1. The NNGE. NNGE [4, 6, 20] is an instance based classifier in which the algorithm creates if then else like rules represented by generalized exemplars. Generalized exemplars may be singles in which case the exemplar represents exactly one example from the training database. Alternatively, generalized exemplars may be hyper rectangles which represent more than one example of the same class from the training database. Training the NNGE algorithm is an incremental process which includes steps named classification, generalization, and dynamic feedback. Each labeled example in the training database is first classified by comparing the new example to all known hyperrectangles and single examples. The classification step in training uses the same methodology

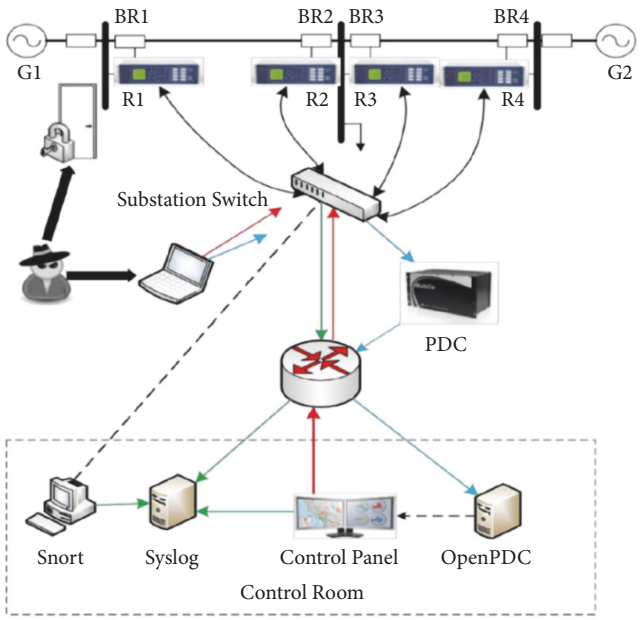

Figure 1: Power system framework for generating test datasets [15].

and distance metric as standalone NNGE classification. Hyperrectangles are generalized rules which represent a class and single examples are previous examples of a class which do not fit into a hyperrectangle. The dynamic feedback is used to adjust feature and exemplar weights used by the distance function.

After training, new examples are classified by calculating the Euclidean distance metric from the example to all exemplars. The new example is classified as the class of the nearest exemplar. The NNGE algorithm preforms generalization by merging exemplars, forming hyper rectangles in attribute space that represent conjunctive rules with internal disjunction. The algorithm forms a generalization each time a new example is added to the database, by joining it to its nearest neighbor of the same class.

1.2. The CPPS Testbed and Datasets. The datasets that we used to evaluate our approach are described in detail in [15]. It consists of synchrophasor measurements from Phasor Measurement Units (PMUs) of four substations. As shown in the testbed diagram of Figures 1 and 2, G1 and G2 are power generators. R1 through R4 are Intelligent Electronic Devices (IEDs) that can switch the breakers on or off. These breakers are labeled BR1 through BR4. Line one spans from breaker one (BR1) to breaker two (BR2) and line two spans from breaker three (BR3) to breaker four (BR4).

Each IED automatically controls one breaker; thus R1 controls BR1, R2 controls BR2, and so on accordingly. The IEDs use a distance protection scheme which trips the breaker on detected faults whether actually valid or faked since they have no internal validation to detect the difference. Operators can also manually issue commands to the IEDs R1 through R4 to trip the breakers BR1 through BR4. The manual override is used when performing maintenance on the lines or other system components. To enhance the cyberattack detection rate, the security attributes such as relay control panel SNORT [21] logs are included in the test datasets. The size of this heterogeneous dataset is approximately 38 Gigabytes and it includes 128 features (e.g., 29 attributes for a single PMU measurement, and four PMUs generate 116 features along with 12 log attributes), which includes nine power system events and 36 cyberattacks. Details of the attributes have been introduced in previous work [5, 6, 15]. The test datasets were generated under 41 scenarios when the power system operated normally (*No Event*) and is distributed by natural faults (*Natural Event*) and cyberattackers (*Attack*). Therefore, the 41 scenarios can be combined into three classes. To characterize and identify normal performance of the power systems, the test datasets were combined as *No Event* and *Natural Event* classes. The dataset is then randomly sampled at one percent and grouped into single Binary Class (i.e., *Normal* and *Attack*), Triple Class (*No Event*, *Natural Event* and *Attack*), and Multiclass(i.e., $1, 2 \ldots 41$). More details of the datasets are found in [15].

1.3. Threat Model and Attacks Scenarios in the Dataset. The dataset has five attacks categories described as follows [15].

(1) *Cyber Command Injection Attacks*. The CPPS Relays are tripped by remote command injection attack. In this attack scenario, a network packet capture tool was used to capture commands used to remotely trip the relay. These commands are replayed on the network from an attacker PC connected to the network switch. In another scenario of this attack in the dataset, insiders physically trip a relay from the face plate.

(2) *Man-In-The-Middle (MITM) Attacks.* These are used to emulate faults and contingencies. The MITM attacks alter power system measurements transmitted from field devices to control room systems. Implemented MITM attacks include a false data injection attack which alters current and voltage phasors and replay attacks which resend captured RTU frames from a previous period. Both of these attacks are used to confuse an operator or automate algorithm monitoring the systems. Faults, generator loss, load changes, and transmission line loss are simulated in the dataset. The MITM attack results in the testbed demonstrate that the attacker randomly alters current phasors in an RTU stream. The range of current for the normal operation in this case is between 200 and 550 Amps. This attack is carried out between the MTU and a control center computer.

(3) *Two Varieties of DOS Attacks Being Used*. First, scripts are available to send high volumes of network traffic (floods) to a network target in attempt to overwhelm network processors and memory. This causes a loss of communication which in turn leads to loss of system monitoring capability. Second, two Python based protocol mutation engines are available to send mutated packets against the IEEE C37.118 protocol [22] that is used for network communication between the CPPS devices. Attacks against this protocol lead to denial of service and a loss of visibility of the state of the power system. The first protocol mutation engine randomly flips bits in network packets. The second protocol mutation engine sends intelligently manipulated packets.

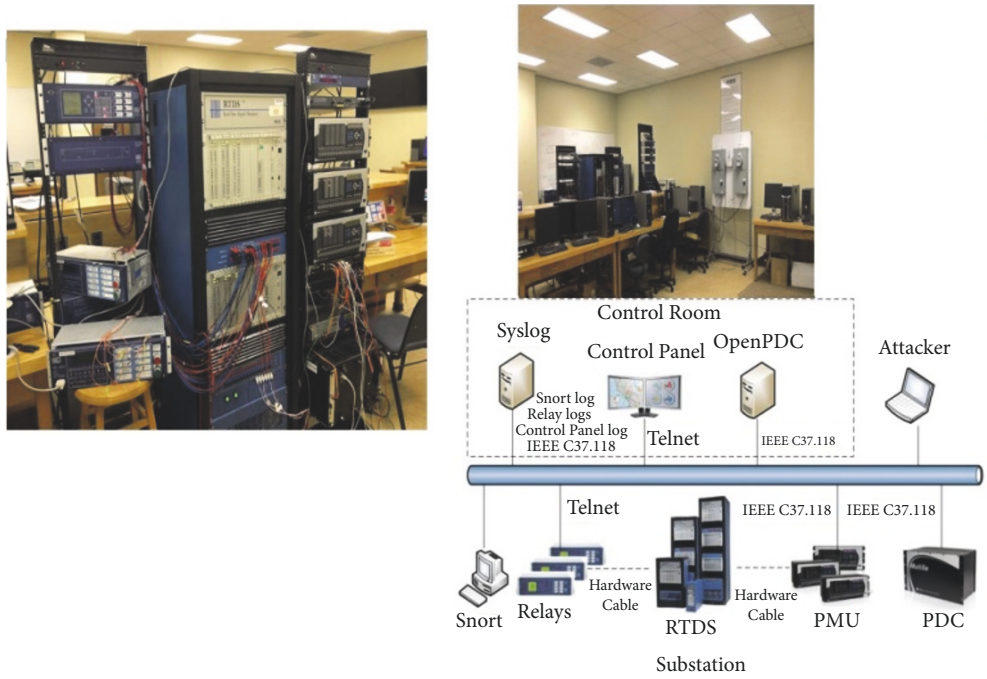

Figure 2: The real-time smart grid testbed.

(4) *A Replay Attack.* This is used to replay an authentication session by an attacker to fool a computer into granting access. This attack appears in the dataset in a form or retransmission of a network data transmission and is usually used to gain authentication in a fraudulent manner.

(5) *Physical HMI and UI Manipulation Attacks.* These are provided to simulate invalid changes to relay settings. These relay settings changes include change threshold and timer values as well as disabling the relay completely. HMI and UI manipulation attacks are automated by an AutoIt script. Such attacks mimic effects of insiders taking illicit control actions and malware taking control of software systems to manipulate control devices. Usually the spoofing, MITM, sniffing, command injection, and DOS attacks lead to these categories of attacks.

2. State of the Art

The recent state of the art refers to several approaches that aim to select feature subsets with maximum discrimination capacity data reduction. We survey these approaches as follows. In [23], authors proposed a feature selection framework that is augmented with a learning method, termed generalized PDF [24] projection theorem (GPPT), to reconstruct the distribution in high-dimensional raw data space from the low-dimensional feature subspace information loss issue in feature reduction. In [25] authors proposed a Bayesian classification approach for automatic text categorization using class-specific features. The proposed approach follows Baggenstoss's PDF projection theorem [24, 26] to reconstruct PDFs in raw data space from the class-specific PDFs in low-dimensional feature space and build a Bayes classification rule. The same approach is extended in [27] using a new divergence measure to measure multidistribution divergence for multiclass classification. However, among the current states of the art, the CFS is the most accurate feature selection approach that works with NNGE [6, 17]. There are several research works that have been conducted to improve the classification accuracy of the NNGE algorithm; in this section, we briefly highlight them. Daniela et al. [19] investigate the ability of an evolutionary pruning mechanism to improve the predictive accuracy of a classifier based on nonnested generalized exemplars. In [28], authors proposed some NNGE variants based on the analyses of the impact of three elements of the NNGE classifier on the classification accuracy of the NNGE algorithm. These elements are the hyperrectangles splitting procedure, the pruning of nongeneralized exemplars, and the presentation order of training instances. In [19] authors used the NNGE to create rules for classifying the attacks by using the Ant-Miner Algorithm. First they created rules using NNGE. After that, they synthesized the rules by removing repetition rules by custom developed rule mining NNGE parser which removes repeated rules obtained. This parser is later pruned using the Ant-Miner algorithm. In [28], authors used the NNGE algorithm for the classification of the rice grains. The classification accuracy rate was high and NNGE was mixed with other image processing and machine learning approaches.

3. The Improved NNGE Algorithm

The improved NNGE algorithm uses our new VHDRA to provide a scalable and accurate classification solution that reduces the attack detection time and the computational

resources consumption. In our experiments, we evaluate the influence of the following three factors on the accuracy and computational performance of the NNGE. These factors are as follows.

(a) The reduction of the input features using a modified Particle Swarm Optimization (PSO) fitness function (Vertical Reduction): the PSO algorithm is used to compute the learning features weights and then ranking the learning features according to their computed weights. After that, the ones with the highest weight (i.e., which means that this feature enables the NNGE to accurately define the shortest Euclidean distance) are only selected to be used with the NNGE. To achieve this, we have developed a fitness function for the PSO algorithm to compute the learning feature weights of the weighted Euclidean distance of the NNGE between a new example and a set of exemplars. In this way, we efficiently integrate PSO with NNGE. We call this kind of reduction the vertical reduction of the input data.

(b) Pruning of nongeneralized exemplars using the highest ranked features of the PSO: the NNGE algorithm learns incrementally by first classifying, then generalizing each new example. When classifying an instance, one or more hyperrectangles may be found that the new instance is a member of wrong class. The algorithm prunes these so that the new example is no longer a member. Once classified, the new instance is generalized by merging it with the nearest exemplar of the same class, which may be a single instance or a hyperrectangle. Authors of [29] proved that generalizing exemplars result in improved classification performance over standard nearest neighbor. The only thing that may pose a problem is that the algorithm tends to produce rules that test a large number of input features that in turn hinder the scalability of the classification model. To this target, we use the reduced features produced by the PSO algorithm to solve this problem and furthermore we reduce the dataset input records using a STEM.

(c) Reducing the input dataset records using a STEM Algorithm (Horizontal Reduction): the STEM [5] is a new data processing and compression method to quantize and compress the heterogeneous datasets to reduce the size of data while preserving original key events and patterns within the datasets. STEM tracks system states from measurements and creates a compressed sequence of states for each observed scenario. It preprocesses the power system events to minimize the state space and number of rules generated by NNGE and results in high classification accuracy, short classification time, and small model building time.

The details of the previous three factors are described in the following three sections.

3.1. VHDRA Vertical Reduction Using a Modified PSO Fitness Function. The previous attempts of feature selection approaches that were tested with the NNGE algorithm such as CFS Expert Knowledge [6, 17], the Mutual Information based Feature Selection (MIFS) with the Joint Mutual Information (JMI) method [11], and the MISF with the Joint Mutual Information Maximisation (JMIM) method [11] have treated all features as equally important in computing the Euclidean distance to the nearest hyper rectangles and this makes them not accurate or suitable for the CPPS where each feature has a different weight. Furthermore, they give insignificant improvements in domains with relevant features such as the CPPS, where any of the features may influence the others. In this section, we introduce a new mechanism that ranks the input features based on their significance and considers the relevant features and their influence on the covering hyper-rectangle of the NGGE algorithm. A feature is considered more significant if its value is closer to a margin of the covering hyper-rectangle. The significant features enable the NNGE to accurately define the shortest Euclidean distance between a new example and a set of exemplars in memory to make a decision whether the new example belongs to a particular class. We implement our approach using the particle swarm optimization (PSO) algorithm that performs well in domains that have a large number of relevant and/or irrelevant features. PSO is one of stochastic optimization methods that are based on the swarming strategies in fish schooling and bird flocking [18]. It considers each solution to the problem in a D-dimensional space as a particle flying through the problem space with a certain position and velocity and finds the optimal solution in the complex search space through the interaction of particles in the population. The implementation of PSO requires few parameters to be adjusted and is able to escape from local optima. The velocity and position of the ith particle are denoted by the two vectors, respectively, $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ and $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. Each particle moves in the search space according to its previous computed best particle position (*pbest)* and the location of the best particle in the entire population (*gbest)*. The velocity and position of the particles are updated using the following [18]:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot rand1_t \cdot \left[pbest_i(t) - x_i(t)\right] + c_2 \cdot rand2_t \cdot \left[\text{gbest}(t) - x_i(t)\right] \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

where the velocity of the ith particle at iteration t *is given by* $v_i(t)$ and its position is given by $x_i(t)$ at the same iteration t, w is a weight factor to balance the global and local search function of particles, c_1 and c_2 are two learning factors which control the influence of the social and cognitive components and they are usually set to 2, *rand1* and *rand2* are two random numbers within the range of $[0, 1]$, $pbest_i(t)$ is the best previous position that corresponds to the best fitness value for ith particle at iteration t, and $gbest(t)$ is the global best particle by all particles at iteration t. The fitness value of the particle is evaluated after changing its position to $x_i(t+1)$. The *gbest and pbest* are updated according to the current position

```
Initialize population
While (number of generations, or the stopping criterion is not met)
  For i = 1 to number of particles
    If the fitness of X_i is greater than the fitness of pbest_i
    then Update pbest_i = X_i
    For k ∈ Neighborhood of X_i
      If the fitness of X_k is greater than that of gbest then
        Update gbest = X_k
    Next k
    For each dimension j
      v_ij(t + 1) = w × v_ij(t) + c_1 × rand_1 × (pbest_ij − x_ij(t))
       + c_2 × rand_2 × (gbest_j − x_ij(t))
      if v_ij(t + 1) ∉ (V_min, V_max) then
            v_ij(t + 1) = max(min(V_max, v_ij(t + 1)), V_min)
            x_ij(t + 1) = x_ij(t) + v_ij(t + 1)
    Next j
  Next i
Next generation until stopping criterion
```

Algorithm 1: The modified PSO Algorithm.

of the particles. The new particle velocity of each dimension $v_i(t+1)$ is tied to a maximum velocity V_{max} that is initialized by the user. As the PSO processes are repeated, all particles evolve toward the optimum solution. We give the pseudo code of the PSO algorithm at iteration t according to [18] with some modification as shown in Algorithm 1.

Our modification focuses on adapting the PSO to work with the NNGE algorithm by developing a new fitness function $x(t)$ as shown in (3), (4), and (5). The PSO fitness function defines the correct classification rate using the features picked by each particle. Figure 3 shows the flowchart of the PSO algorithm.

$$x_i(t) = \emptyset.\Omega_t(A_i) + \theta.(n - |(A_i)|) \quad (3)$$

$$\Omega_t(A_i) = \min_{f \in A_i}(\gamma_i(f)) \quad (4)$$

$$\gamma_i(f) = \min\left\{E_i(f) - H_i^{min}(f), H_i^{max}(f) - E_i(f)\right\} \quad (5)$$

where

(i) $x_i(t)$ is the fitness of particle i (one record of the dataset) at iteration t and it denotes how much a particle i features values are closer to a margin of the covering hyper-rectangle H which is going to be split through the NNGE classifier using the selected subset of features A of particle i. In other words, the main target of the fitness function is to choose the feature which ensures the most "balanced split" and in case there is a tie (two or more features have the same distance to a margin of H), the attribute leading to the largest number of training examples included in one of the splitting hyper-rectangles will be chosen.

(ii) A_i is the feature subset of particle i at iteration t; that is, $A = \{f1, f2, \ldots..f_{|A|}\}$.

(iii) $|(A_i)|$ is the length of the feature subset without the nonvalue features

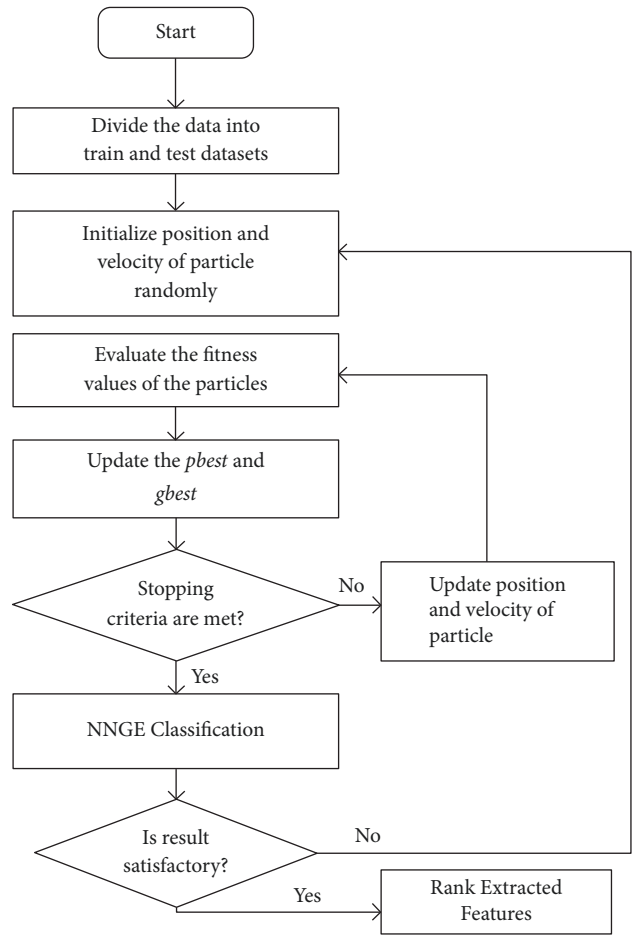

Figure 3: Feature ranking using PSO and the proposed fitness.

(iv) n is the total length of the feature subset including the nonvalue features.

(v) Ω_t is a measure of the classifier performance. It returns, for the whole subset of features A of particle i at iteration t, the shorts distance that any of these features can achieve to a margin of the covering hyper-rectangle H.

(vi) $\emptyset, \theta$ are two parameters that control the relative weight of classifier performance and feature subset length, $\emptyset \in [0, 1]$ and $\theta = 1 - \emptyset$. This formula denotes that the classifier performance and feature subset length have different effect on fitness function. In our experiments, we consider that classifier performance is more important than subset length because most of the power grid dataset records are of similar size and they have very few nonvalue features, so we set them to $\emptyset = 0.9, \theta = 0.1$.

(vii) γ_i denotes how much a certain feature f value *is* closer to a margin of the covering hyper-rectangle H.

(viii) E_i is the conflicting example of particle i; it represents an example record of dataset that needs to be classified.

(ix) H_i^{min}, H_i^{max} are the minimum and maximum margin values, respectively, of the covering hyper-rectangle H.

The iteration of the PSO will continue and stop when either one of the stopping criteria is met; (i) maximum number of iterations is defined to PSO or (ii) the fitness of the proposed feature subset has exceeded the maximum fitness value being set. We will use the fitness function given in (3) to compute the fitness of each particle in the dataset. For each feature f, the parameter γ_i will be computed; then at each point a stopping criterion is met; (ω) the minimum value of γ_i parameters corresponding to each feature f is selected. After that, all features are sorted according to their significance to the NNGE classification from the smallest to the largest one. Only few numbers of good features that exceed a particular threshold T that is computed during the training phase are selected. The threshold identified to select the most significant features is computed in the training phase for the Binary, Triple, and Multiclass datasets. To compute these three thresholds, the fitness of the PSO is defined by using the k-fold cross-validation, where k = 128, number of features in the dataset. The training dataset is divided into k subsets of the same size. One subset is used for validation. Using the remaining k–1 subsets, we build the new PSO fitness function based on (3), (4), and (5) that compute the nearest hyper rectangles of the NNGE. Each subset is used once for validation using one feature, and the process is repeated k times. We compute the threshold T using (6) by computing the average of the measure of fitness (γ_i), defined at (5), for all features from the 128 trials.

$$T = \frac{\left(\sum_{i=1}^{n} \gamma_i\right)}{n} \tag{6}$$

where n denotes number of validation data.

In the second phase, NNGE is used for classification using the top significant features that have fitness values ω lower than T. The classification step is based on the computation of the distance $D(E,H)$ between an example $E = (E1, E2, \ldots, En)$ and a hyper-rectangle H as given in (7) [7]. Figure 4 illustrates an example of splitting variants for numerical attributes.

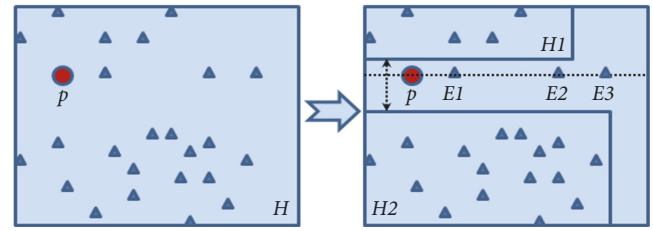

Figure 4: An example of NNGE splitting variants based on the extracted input features.

$$D(E, H) = \sqrt{\sum_{i=1}^{n} \left(w_i \frac{d(E_i, H_i)}{E_i^{max} - E_i^{min}} \right)} \tag{7}$$

where

(i) N is number of features in the current Example E.

(ii) E_j^{max}, E_j^{min} define the range of values over the training set which correspond to attribute i.

(iii) H_i is the interval $[H_i^{min}, H_i^{max}]$.

(iv) d is the distance between the features values and the corresponding hyper-rectangle "side" and it is computed according to

$$d(E_i, H_i) = \begin{cases} 0, & H_i^{min} \le E_i \le H_i^{max} \\ H_i^{min} - E_i, & E_i < H_i^{min} \\ E_i - H_i^{max}, & E_i > H_i^{max} \end{cases} \tag{8}$$

As shown in Figure 4, the left picture shows that the initial instances (filled triangle) belong to one hyper-rectangle H. After splitting, right picture shows that each instance is assigned to a particular hyper-rectangle *H1 or H2 according to* the closest margin strategies described before; some example instances *p1, E1, E2,* and *E3* represent a tie case described before; then, in such case, one of the features of these examples which leads to the largest number of training examples included in one of the splitting hyper-rectangles *H1* or *H2* will be chosen.

3.2. VHDRA Vertical Reduction Using the Evolutionary Pruning. There are two main approaches to reduce the size of classifiers: prepruning and postpruning. The prepruning approach aims to select the good training instances or prototypes and those are aiming at selecting the relevant attributes. This is achieved by VHDRA through the vertical data reduction using the PSO algorithm. The postpruning approach is applied to a set $H = \{H_1, H_2, \ldots, H_K\}$ of NNGE hyperrectangles once it has been generated with the aim of reducing its size and improving its classification accuracy.

In this paper, the selection of the hyperrectangles is based on the evolution of a population of binary encoded elements corresponding to various subsets of the initial set of hyperrectangles. In [19], two evolution pruning algorithms are introduced, the first version of the algorithm called EP-NNGE (Evolutionary Pruning in NNGE) and the EPA-NNGE algorithm. Authors of [19] proved that EPA-NNGE achieves high accurate classification results. In this paper, we use the EPA-NNGE to prune the hyperrectangles of the NNGE. EPA-NNGE is based on the idea of evolving a population of M binary strings containing K components. Each element x of the population corresponds to a subset of H; for example, if a component x_k has the value 1, it means that H_k is selected into the model, while if it is 0, it means that H_k is not selected. The quality of an element x is quantified using two measures: one is related to the accuracy of the classifier based on the selected hyperrectangles $H(x)$ and the other is related to the reduction of the model size. Thus the fitness is given by

$$f(x) = \lambda Acc(\mathscr{H}(x)) + (1-\lambda)\frac{|\mathscr{H}| - |\mathscr{H}(x)|}{|\mathscr{H}|} \quad (9)$$

where

(i) Acc denotes the accuracy that is computed by counting the correctly classified instances covered by the hyperrectangle that also means the total number of instances covered by the hyperrectangle after excluding the conflicting examples.

(ii) $|H|$ denotes the number of hyperrectangles.

(iii) $\lambda \in (0, 1)$ is a parameter controlling the compromise between the two quality measures.

The population elements of the EPA-NNGE algorithm are evaluated using (9). The computation of the classification accuracy of the EPA-NNGE algorithm is based on the computation of the distance between a test instance and a hyperrectangle and only the selected attributes (as are they specified by the corresponding part Xs of the population elements) are only considered. This means instead of using (7), we will use

$$D(E, H) = \sqrt{\sum_{i=1}^{n}\left(X_s w_i \frac{d(E_i, H_i)}{E_i^{max} - E_i^{min}}\right)} \quad (10)$$

3.3. VHDRA Horizontal Data Reduction Using STEM. The horizontal reduction approach uses the STEM. The STEM was used to preprocess power system data from a diverse set of sensors. Sensors include PMU, relay logs, control panel logs, and a network monitor called SNORT. The PMU provides mostly continuous data types and the other sensors provide discrete inputs. Section 4.3 gives a practical example of the STEM, and its detailed steps are discussed in [6] and are summarized in the following.

(1) Collect raw data: a PMU provides measurements of different electrical quantities from PMU, relay logs, control panel logs, and SNORT.

(2) Merge raw data: the previous measurements are upsampled prior to merging.

(3) Quantization: measurements with large state space should be quantized in such a way that the quantization interval maintains important patterns in the data.

(4) State mapping and compression: each row is mapped to a state ID. Unique states are provided an ID and repeated states use the same ID as previous instances of that state. After mapping, each row will have an assigned state ID. Compression removes repeated states in a sequence. After compression, each event is represented by a temporally ordered list of states and a label. After that, STEM sequentially captures only distinct states, compresses states, and stores it in the database. In this way the data size is significantly reduced by intelligently pruning the repetitive states.

4. Experimental Analysis and Results

We have conducted three types of experiments to evaluate the following.

(1) The effectiveness of the VHDRA vertical data reduction is by using the new fitness function of the PSO in selecting the significant features that their values are closer to a margin of the covering hyper-rectangle of the NNGE and the impact of this reduction on the classification accuracy of the NNGE algorithm

(2) The impact of the vertical data reduction is by using the EPA-NNGE pruning algorithm on the classification accuracy of the NNGE and on reducing the model size which in turns reduces the computational resources consumption. These experiments evaluate the reduction of the computational resources consumption in terms of the number of reduced hyperrectangles and ignored features that are defined by the pruning process in the training phase of the NNGE.

(3) There is the impact of the horizontal reduction of the STEM on the NNGE classification accuracy and the reduction of the model size in terms of the number of reduced hyperrectangles.

4.1. Evaluate the Impact of the VHDRA Vertical Reduction Using the New PSO Fitness Function. In these experiments, we evaluate the impact of the VHDRA vertical data reduction using the new PSO Euclidean distance fitness function on the NNGE classification accuracy and the model size. The experiments use the power grid dataset described in Section 1.2. In the training phase, we compute a particular threshold to extract the most significant features. The threshold values in the Binary, Triple, and Multiclass datasets, respectively, are 44.38, 61.23, and 81.78. Any feature with a fitness value larger than these thresholds is ignored. The algorithm defines the most significant 17 features for the Binary class dataset, 19 features for the Triple class dataset, and 22 features for the Multiclass datasets as shown in Table 1.

Table 1: The most significant selected features fitness values in the binary, triple, and multiclass datasets.

Binary class dataset			Triple class dataset			Multiclass		
Order	Feature name	Fitness value	Order	Feature name	Fitness value	Order	Feature name	Fitness value
1	R2-CPA1	9.2	1	R1-VPA1	12.5	1	relay1_log	22.2
2	R3-CPA1	11.1	2	relay1_log	13.2	2	relay3_log	25.3
3	relay1_log	12.5	3	R1-CPM1	16.0	3	R4-VPA2	29.1
4	relay4_log	13.8	4	R2-VPA1	18.8	4	R4-VPM2	33.6
5	relay2_log	17.1	5	relay4_log	18.9	5	R4-VPA3	37.1
6	relay3_log	19.7	6	R2-CPM1	22.0	6	relay4_log	42.5
7	R1-VPA1	20.1	7	R3-VPA1	26.3	7	relay2_log	49.8
8	R4-CPA1	21.0	8	R3-CPA1	29.4	8	snort_log1	51.0
9	snort_log3	28.5	9	R3-CPM1	33.0	9	snort_log2	55.3
10	R1-CPA1	31.0	10	relay2_log	38.2	10	R4-VPM1	59.3
11	R2-VPA1	34.8	11	R4-VPM1	39.0	11	R2-CPA1	62.5
12	R3-VPA1	35.2	12	R4-CPA1	39.7	12	R3-CPA1	64.0
13	R4-CPM1	39.4	13	R4-CPM1	41.3	13	R2-VPA1	68.0
14	R4-Power	40.3	14	R3-Power	47.0	14	R3-VPA1	69.4
15	R3-Power	41.0	15	R2-Power	50.4	15	snort_log3	72.9
16	R3-CPM1	42.9	16	R1-CPA1	53.2	16	R4-CPM1	73.3
17	R2-Power	43.8	17	R4-VPA1	54.3	17	R3-Power	73.5
			18	R2-CPA1	57.0	18	R1-VPA1	75.4
			19	relay3_log	59.4	19	R2-Power	77.5
						20	R2-CPM1	79.5
						21	R4-Power	80.0
						22	R3-CPM1	81.0

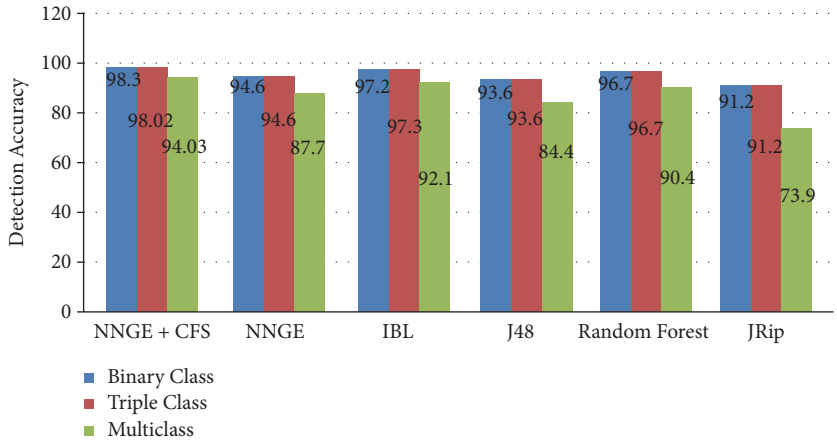

Figure 5: A comparison between the VHDRA using the PSO and the other existing approaches using the Binary, Triple, and Multi-class Datasets.

Table 2: NNGE classification rate using the Binary, Triple, and Multiclass datasets.

Feature selection	Binary class	Triple class	Multiclass
With (%)	98.38	98.01	94.03
Without (%)	65.42	66.41	23.66

Table 3: A comparison between the VHDRA using the PSO algorithm and the other existing approaches using the binary, triple, and multiclass datasets.

Approach	Binary class		Triple class		Multiclass	
	No. of features	Detection rate (%)	No. of features	Detection rate (%)	No. of features	Detection rate (%)
VHDRA (PSO)	*17*	*98.38*	*19*	*98.01*	*22*	*94.03*
NNGE (CFS)	28	94.68	28	94.62	28	87.77
IBL	28	97.20	17	97.38	28	92.10
J48	28	93.69	28	93.69	28	84.45
Random Forest	28	96.77	28	96.77	28	90.41
JRip	28	91.20	28	91.22	129	73.94

To test the accuracy of the selected features, we apply the NNGE classifier using (6) to compute the classification rates using the three datasets; see Table 2. In the following, we compare the output of our VHDRA with the new PSO fitness function against the most accurate five classification algorithms that we have tested before [11], namely, the traditional NNGE, Instance-based Learning (IBL), J48 tree, Random Forest, and JRip. According to our previous experiments [11], the best feature selection approach among the existing ones is the CFS. We used the CFS with the previous mentioned five classification algorithms using the Binary, Triple, and Multiclass datasets to compare the classification accuracy of these approaches against our approach. Table 3 and Figure 5 show that VHDRA with the new PSO fitness function uses less number of features and outperforms the classification accuracy of the current classification algorithms.

4.2. Evaluate the Impact of the VHDRA's Vertical Reduction Using the EPA-NNGE Pruning Algorithm. In these experiments, we evaluate the impact of the VHDRA vertical data reduction using the EPA-NNGE pruning algorithm on the classification accuracy and the model size. In these experiments, we use the significant features selected in the previous experiments of Section 4.1 using the modified PSO fitness function as follows: 17 features from the Binary dataset, 19 features from the Triple dataset, and 22 from the Multiclass datasets. Since our main goal of our approach is both to improve the classification accuracy and to reduce the model size, we edit the evolutionary process by a fitness function based on a value of λ that corresponds to an equilibrium point at which the EPA-NNGE accuracy rate and the hyperrectangles reduction rate are equal. The EPA-NNGE accuracy rate is computed for each dataset as a ratio between the numbers of correctly classified records/instances to the total number of records/instances in the dataset. The hyperrectangles reduction ratio is defined as $(|H| - |H(xbest)|)/|H|)$ where xbest is the instance with the corresponding best $f(x)$ value which is computed using (8). The influence of the parameter λ on the accuracy and on the reduction of the model size is evaluated for the Binary, Triple, and Multiclass datasets as shown in Figures 6, 7, and 8, respectively. The best values of λ that corresponds to the equilibrium point in the three datasets

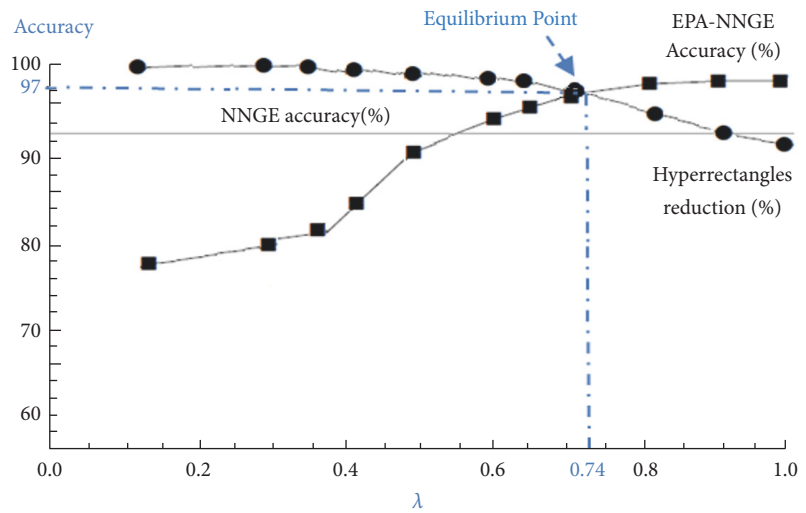

Figure 6: Influence of λ on the EPA-NNGE accuracy gain and hyperrectangles reduction using the Binary class dataset.

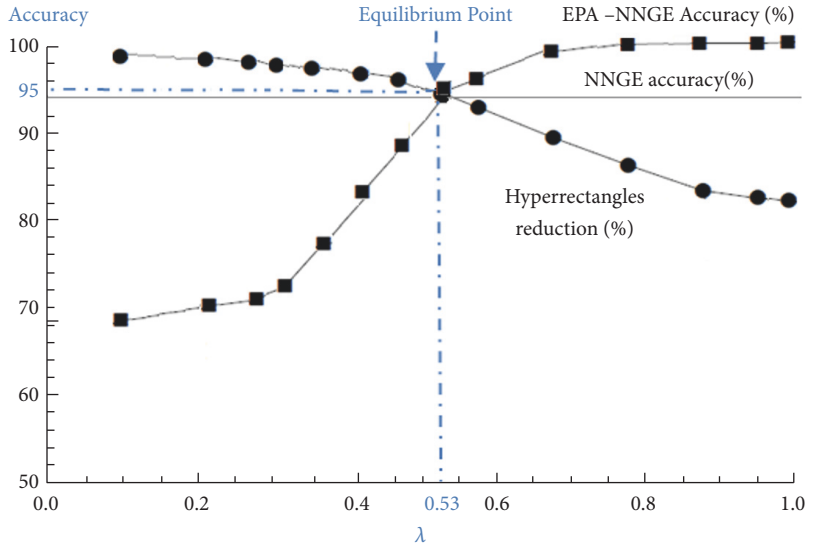

Figure 7: Influence of λ on the EPA-NNGE accuracy gain and hyperrectangles reduction using the Triple class dataset.

are 0.74, 0.53, and 0.44, respectively. To evaluate the impact of the pruning algorithm on the reduction of the model size and accuracy gain, we consider both the hyperrectangles reduction ratio described before and the reduced number of features as shown in Table 4.

An overall view of the accuracy gain ratio, hyperrectangles reduction ratio, and attributes reduction ratio for the Binary, Triple, and Multiclass Dataset is shown in Figure 9 for the VHDRA reduction using the EPA-NNGE and PSO versus the traditional NNGE with CFS without the pruning capabilities. The accuracy gain is computed as $(Acc(VHDRA) - Acc(NNGE))/Acc(NNGE) * 100)$. The ratio of the reduced hyperrectangles is computed as $|H_{VHDRA}|/|H_{NNGE}| * 100)$. The ratio of the reduced features is computed as $(N_{VHDRA}/N_{NNGE} * 100)$. Table 5 shows the accuracy gain, features reduction ratio, and hyperrectangles reduction ratio. The largest gain in accuracy (9.25%) was obtained using the Multiclass dataset. This can be explained by the fact that this dataset has the highest hyperrectangles reduction ratio (13.07%) and the lowest feature reduction ratio (35.71%). The smallest reduction in the model size (including the feature reduction and hyperrectangles reduction ratios) occurs using the Triple class dataset. This is why this dataset has the lowest accuracy gain (4.29%).

4.3. Evaluation of the Horizontal Data Reduction Using STEM. In this section, we evaluate the impact of the horizontal data reduction using STEM on the NNGE classification accuracy and the Model Size. The following STEM steps are applied.

(1) After collecting and merging raw data, we use the significant features selected in the previous experiments using the PSO and EPA approaches as follows: 15 features from the Binary dataset, 14 features from the

Table 4: A comparison between the VHDRA vertical reduction and the other existing approaches.

	Binary class *Size: 81,664 MB, 128 features*		Triple class *Size: 81,669 MB, 128 features*		Multiclass *Size: 78,963 MB, 128 features*	
Approach	No. of features	Detection rate (%)	No. of features	Detection rate (%)	No. of features	Detection rate (%)
VHDRA (PSO+ EPA)	*15* *(9.57MB)*	*99.21* *EPA Accu. gain:0.83%*	*14* *(8.93MB)*	*98.91* *EPA Accu. gain:0.90%*	*18* *(11.10MB)*	*97.03* *EPA Accu. gain: 3.0%*
VHDRA (PSO)	17 (10.85MB)	98.38	19 (12.12MB)	98.01	22 (13.57MB)	94.03
NNGE (CFS)	28 (17.86MB)	94.68	28 (17.91MB)	94.62	28 (17.77MB)	87.77
IBL	28 (17.97MB)	97.20	17 (10.87MB)	97.38	28 (17.62MB)	92.10
J48	28 (18.18MB)	93.69	28 (17.96MB)	93.69	28 (18.92MB)	84.45
Random Forest	28 (18.29MB)	96.77	28 (18.20MB)	96.77	28 (17.54MB)	90.41
JRip	28	91.20	28	91.22	129	73.94

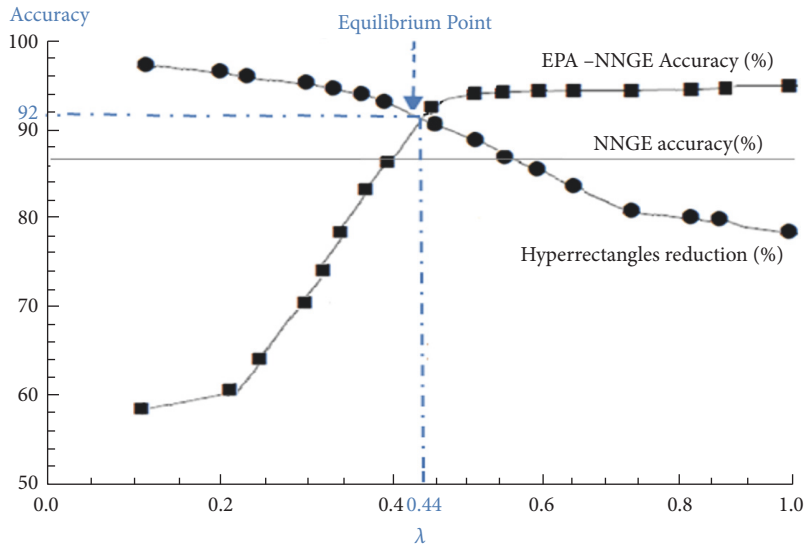

Figure 8: Influence of λ on the EPA-NNGE accuracy gain and hyperrectangles reduction using the Multi–class dataset.

Table 5: A comparison between the VHDRA and the traditional NNGE with CFS.

	Binary class	Triple class	Multiclass
Accuracy gain (%)	4.53	4.29	*9.25*
Feature reduction ratio (%)	46.43	*50*	35.71
Hyperrectangles reduction ratio (%)	9.68	7.41	*13.07*

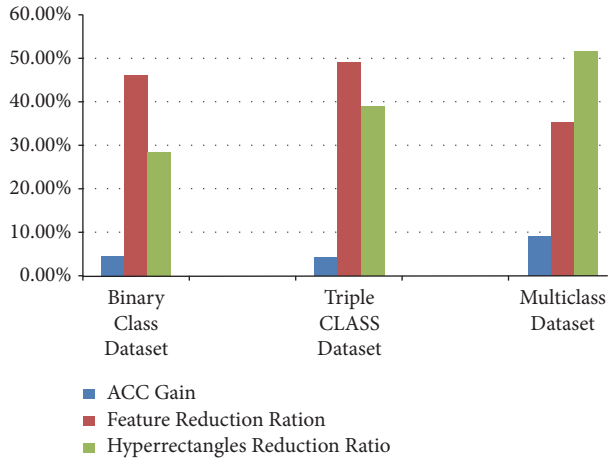

Figure 9: VHDRA with EPA-NNGE and PSO vs. NNGE-CFS: accuracy gain ratio, hyperrectangles reduction ratio, and attributes reduction ratio for the Binary, Triple, and Multi-class Dataset.

Triple dataset, and 18 from the Multiclass datasets; see Table 6. The numeric values of each feature are then quantized using domain expert inputs. The quantization intervals are shown in Table 6. To compare VHDRA to the NNGE with CFS, we repeated this step with the 28 features of the NNGE with CFS.

(2) Assigning states to each sample is as follows.

(a) Any row of the quantized data that contains the same quantization values for each feature is assigned the same state such as {S0, S1 . . . S14}

(b) The states of each data are inserted into the Quantized Datasets using the selected features.

(3) The duplicated values (i.e., the records in the dataset have the same state and marker values) are deleted. Only unique records are saved. Using this step, the size of the dataset was reduced from 5MB raw datasets to 6KB. This reduction can significantly enhance the detection speed.

The horizontal reduction improves the performance of the NNGE with CFS by 29.15%, 21.42%, and 17.81% using the Binary, Triple, and Multiclass datasets, respectively. The overall performance of VHDRA using both the vertical and the horizontal reduction improves the accuracy of the NNGE with CFS by 1.25%, 1.18%, and 5.23% using the Binary, Triple, and Multiclass datasets, respectively. STEM works also for online detection. The dataset is used to train the system and the quantization intervals are defined based on the dataset data boundaries. For online detection, the dataset is periodically updated and quantization intervals are modified. Furthermore, the states of each data are also updated and inserted into the Quantized Datasets and duplicated values are deleted.

Table 7 compares the detection accuracy gain ratio and the hyperrectangle reduction ratio of all VHDRA elements against the NNGE with CFS. From this table, we can notice

Table 6: Measurement quantization.

Feature	Quantization interval name	Quantization interval enumeration	Range
Voltage	Low, Normal, High	{0, 1, 2}	{(0, 130MV], (120MV, 146MV], (146MV, inf)}
Current	Low, Normal, Warming, High	{0, 1, 2, 3}	{(0, 100A], (100A, 700A], (700A, 12000A], (1200A, inf)}
Frequency	Low, Normal, High	{0, 1, 2}	{(0, 59.8Hz], (59.8Hz, 60.2Hz], (60.2Hz, inf)}
Impedance	Zone1, Zone2, Normal	{0, 1, 2}	{(0, 7.9], (7.9, 9.875], [9.875, inf)}
Control, relay, and SNORT log	Yes, No	{0, 1}	{0, 1}

Table 7: A comparison between VHDRA elements against NNGE with CFS.

	Binary class		Triple class		Multiclass	
	Accuracy gain (%)	Hyperrect. Reduction (%)	Accuracy gain (%)	Hyperrect. Reduction (%)	Accuracy gain (%)	Hyperrect. Reduction (%)
VHDRA (vertical and horizontal)	4.19	*37.34*	3.78	*26.76*	8.57	*29.06*
VHDRA (vertical)	*4.53*	9.68	*4.29*	7.41	*9.25*	13.07
VHDRA (horizontal)	1.25	29.15	1.18	21.42	5.23	17.81

that VHDRA with its vertical and horizontal reduction features achieves the best performance by reducing the NNGE hyperrectangles while keeping good detection accuracy.

5. Conclusion and Future Work

The large volume of high speed heterogeneous data of the CPPS makes their network targets for intrusions. In this paper, we introduced the VHDRA, a Vertical and Horizontal Data Reduction Approach, to improve the detection accuracy and speed and reduce the computational resource consumption of the NNGE algorithm. VHDRA reduces the NNGE's hyperrectangles by pruning the nongeneralized exemplars using the highest ranked features of the PSO.

It also reduces the size of dataset while preserving original key events and patterns within the datasets using the State Tracking and Extraction Method. The experiments show that the NNGE using VHDRA outperforms the current classification techniques for the Multi-, Binary, and Triple class datasets, respectively, as follows: the vertical reduction improves the accuracy of the NNGE with CFS by 9.25%, 4.53%, and 4.29% and reduces the features by 35.71%, 46.43%, and 50%. It also reduces the hyperrectangles by 13.07%, 9.68%, and 7.41%. On the other hand, the horizontal reduction improves the accuracy of the NNGE with CFS by 5.23%, 1.25%, and 1.18%.

The overall performance of VHDRA using both the vertical and the horizontal reduction reduces the hyperrectangles by 29.06%, 37.34%, and 26.76% and improves the accuracy of the NNGE with CFS by 8.57%, 4.19%, and 3.78%.

For future work, we will evaluate the scalability, actual computational resource consumption, and the speed of the VHDRA. Furthermore, we will also study the influence of quantizing and clustering the input data of the STEM using an intelligent model that integrates an Expert System with a Neural Network one instead of using domain expert input data on the accuracy and computational performance of our approach. Furthermore, to detect zero-day attacks, we will integrate our finite state Hidden Markov Model [30] to predict multistage and zero-day attacks in CPPS.

Data Availability

Reference [15] refers to the dataset used in this research. Furthermore, the tools used to implement the framework are given in the references.

Disclosure

The statements made herein are solely the responsibility of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was made possible by NPRP Grant # NPRP9-005-1-002 from the Qatar National Research Fund (a member of Qatar Foundation).

References

[1] H. Bao, R. Lu, B. Li, and R. Deng, "BLITHE: behavior rule-based insider threat detection for smart grid," *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 190–205, 2016.

[2] S. Pan, *Cybersecurity testing and intrusion detection for cyber-physical power systems [Ph.D. thesis]*, Mississippi State University, 2014.

[3] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences*, vol. 223, pp. 119–135, 2013.

[4] D. Zaharie, L. Perian, and V. Negru, "A view inside the classification with non-nested generalized exemplars," in *Proceedings of the IADIS European Conference on Data Mining*, 2011.

[5] U. Adhikari, *Event and intrusion detection systems for cyber-physical power systems [Ph.D. thesis]*, Mississippi State University, 2015.

[6] D. Zaharie, L. Perian, V. Negru, and F. Zamfirache, "Evolutionary pruning of non-nested generalized exemplars," in *Proceedings of the 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2011*, Timişoara, Romania, May 2011.

[7] U. Adhikari, M. Thomas, and P. Shengyi, "Applying non-nested generalized exemplars classification for cyber-power event and intrusion detection," *IEEE Transactions on Smart Grid*, vol. 9, pp. 3928–3941, 2018.

[8] T. B. Rasmussen, G. Yang, A. H. Nielsen, and Z. Dong, "A review of cyber-physical energy system security assessment," in *Proceedings of the IEEE Power and Energy Society PowerTech Conference*, pp. 1–6, Manchester, United Kingdom, June 2017.

[9] J. Valenzuela, J. Wang, and N. Bissinger, "Real-time intrusion detection in power system operations," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1052–1062, 2013.

[10] "National Infrastructure Protection Plan, DHS," 2018, https://www.dhs.gov/xlibrary/assets/NIPP_RiskMgmt.pdf.

[11] D. Harp and B. Gregory-Brown, *The State of Security in Control Systems Today*, SANS Institute, June 2015.

[12] Protecting Industrial Control Systems and SCADA Networks, Check Point Software Technologies Ltd., May 2015.

[13] *Protecting Industrial Control Systems, Recommendations for Europe and Member States*, ENISA, 2011.

[14] K. Grudziński, M. Grochowski, and W. Duch, "Pruning classification rules with reference vector selection methods," *ICAISC*, 2010.

[15] U. Adhikari, T. H. Morris, and S. Pan, "A cyber-physical power system test bed for intrusion detection systems," in *Proceedings of the 2014 IEEE Power & Energy Society General Meeting*, pp. 1–5, National Harbor, MD, USA, July 2014.

[16] H. A. Kholidy and F. Baiardi, "CIDD: A cloud intrusion detection dataset for cloud computing and masquerade attacks," in *Proceedings of the 9th International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, Nev, USA, April 2012.

[17] Q. Chen, H. A. Kholidy, S. Abdelwahed, and J. Hamilton, "Towards realizing a distributed event and intrusion detection system," in *Proceedings of the International Conference on Future Network Systems and Security (FNSS 2017)*, Springer, Gainesville, Fla, USA, August 2017.

[18] M. Brent, *Instance Based Learning: Nearest Neighbor with Generalization*, University of Waikato, Hamilton, New Zealand, 1995.

[19] NimmyCleetus and K. A. Dhanya, "Rule induction using ant-miner algorithm," *International Journal of Scientific & Engineering Research*, vol. 5, no. 2, 2014.

[20] B. Martin, "Instance-Based Learning: Nearest Neighbour with Generalisation," Working Paper Series 95/18 Computer Science, University of Waikato, Hamilton, New Zealand, 1995.

[21] H. A. Kholidy, A. Erradi, S. Abdelwahed, and F. Baiardi, "A risk mitigation approach for autonomous cloud intrusion response system," *Computing: Archives for Scientific Computing*, vol. 98, no. 11, pp. 1111–1135, 2016.

[22] K. E. Martin, D. Hamai, M. G. Adamiak et al., "Exploring the IEEE standard C37.118-2005 synchrophasors for power systems," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 1805–1811, 2008.

[23] V. K. Ojha, K. Jackowski, A. Abraham, and V. Snasel, "Dimensionality reduction, and function approximation of poly(lactic-co-glycolic acid) micro- and nanoparticle dissolution rate," *International Journal of Nanomedicine*, 2015.

[24] P. M. Baggenstoss, "Class-specific feature sets in classification," *IEEE Transactions on Signal Processing*, vol. 47, no. 12, pp. 3428–3432, 1999.

[25] B. Tang, H. He, P. M. Baggenstoss, and S. Kay, "A bayesian classification approach using class-specific features for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, 2016.

[26] P. M. Baggenstoss, "The PDF projection theorem and the class-specific method," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 672–685, 2003.

[27] B. Tang, S. Kay, and H. He, "Toward optimal feature selection in naive bayes for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2508–2521, 2016.

[28] O. Aki, A. Güllü, and E. Uçar, "Classification of rice grains using image processing and machine learning techniques," in *Proceedings of the International Scientific Conference*, Gabrovo, Bulgaria, 2015.

[29] H. A. Kholidy, A. Tekeoglu, S. Iannucci et al., "Attacks detection in SCADA systems using an improved non-nested generalized exemplars algorithm," in *Proceedings of the 12th IEEE International Conference on Computer Engineering and Systems (ICCES 2017)*, December 2017.

[30] H. A. Kholidy, A. Erradi, S. Abdelwahed, and A. Azab, "A finite state hidden markov model for predicting multistage attacks in cloud systems," in *Proceedings of the 12th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC)*, Dalian, China, August 2014.