Attacks Detection in SCADA Systems using an Improved Non-Nested Generalized Exemplars Algorithm

Hisham A. Kholidy [1, 2]
hisham_dev@yahoo.com

Ali Tekeoglu [2]
ali.tekeoglu@sunyit.edu

Stefano Iannucci [3]
stefano@dasi.msstate.edu

Shamik Sengupta [1]
ssengupta@unr.edu

Qian Chen [4]
guenevereqian.chen@utsa.edu

Sherif Abdelwahed [3]
sherif@ece.msstate.edu

John Hamilton [3]
hamilton@cci.msstate.edu

[1] Department of Computer Science & Engineering, University of Nevada, Reno, NV, USA.
[2] SUNY Polytechnic Institute, College Of Engineering, Department of Network & Computer Security, Utica, NY
[3] Distributed Analytics and Security Institute, Mississippi State University, Starkville, MS, USA.
[4] Department of Electrical and Computer Engineering, The University of Texas at San Antonio, TX, USA.

Abstract— **Supervisory Control and Data Acquisition (SCADA) systems became vital targets for intruders because of the large volume of its sensitive data. The Cyber Physical Power Systems (CPPS) is an example of these systems in which the deregulation and multipoint communication between consumers and utilities involve large volume of high speed heterogeneous data. The Non-Nested 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 our new Feature Selection and Data Reduction Method (FSDRM) to improve the classification accuracy and speed of the NNGE algorithm and to reduce the computational resource consumption. FSDRM provides the following functionalities: (1) it reduces the dataset features by selecting the most significant ones, (2) it reduces the NNGE's hyperrectangles classifiers. The experiments show that the FSDRM reduces the NNGE hyperrectangles by 29.06%, 37.34%, and 26.76% and improves the classification accuracy of the NNGE by 8.57%, 4.19%, and 3.78% using the Multi, Binary, and Triple class datasets respectively.**

Keywords—Intrusion Detection, Power Systems, Data Reduction, Feature Selection, Pruning non-generalized exemplars, NNGE

I. INTRODUCTION

Modern SCADA systems′s operators typically require data to be transferred between industrial and external networks. This has created the potential for malware and hackers to gain access to and disrupt real time control systems and dependent infrastructure. The CPPS are one of these vital SCADA systems that require special cybersecurity efforts. The Wide Area Measurement Systems (WAMS) of the CPPS plays an important role in monitoring and controlling 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 cyber-attacks against the electric power grid. The CPPS need to be resilient to cyber-attacks through a precise and scalable attack classification technique that can deal with the large volume of high speed heterogeneous 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 methodology [1] monitoring devices 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 [3,4] that monitors system security states and sends the alerts when the system behavior approaching to an unsafe or disallowed state, the common path mining approach [5] that creates an IDS using heterogeneous data for detecting power system cyberattacks using the State Tracking and Extraction Method (STEM) algorithm [5] to pre-process 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] is used to create an offline and online event intrusion detection systems using STEM to process the CPPS security datasets. However, these approaches are still neither accurate nor scalable enough to process the high speed big data of the CPPS. The NNGE algorithm is among the most accurate classification technique that can work with heterogeneous datasets formats such as the WAMS data. NNGE is able to classify multiclass scenarios, sequential data, and handle heterogeneous datasets formats such as discrete, nominal, symbolic, continuous, and non-value features [7, 8]. In this work, we introduce a new data reduction method called FSDRM, Feature Selection and Data Reduction Method, which provides feature selection, exemplar pruning, feature reduction, and hyper rectangles reduction functionalities. FSDRM improves the classification accuracy, speed, and reduces the computational resource consumption of the NNGE algorithm through:

1) Selecting the most significant features in the dataset. To this target, we develope a new fitness function for the Particle Swarm Optimization (PSO) algorithm [9] that adopts

978-1-5386-1191-3/17/$31.00 ©2017 IEEE

the classification function of the NNGE algorithm by selecting the significant features that their values are closer to a margin of the covering hyper-rectangle.

2) Pruning of non-generalized exemplars using the highest ranked features of the PSO. FSDRM uses the Evolutionary Pruning Algorithms (EPA-NNGE) [10] to improve the classification accuracy of NNGE and to reduce the model size by reducing the hyperrectangles and ignoring the non-selected features among the selected significant ones defined by the new fitness function of the PSO.

To evaluate the accuracy of the FSDRM, we compare the detection rates of the NNGE using FSDRM against current classification approaches including the NNGE with its best feature selection approach namely the Correlation based Feature Selection (CFS) [11]. The comparison uses an existing intrusion detection power grid dataset [12]. To evaluate the improvement of the detection speed and computational resource consumption, we compare the number of reduced hyperrectangles using the NNGE with CFS, FSDRM with the feature reduction only, FSDRM with both the feature reduction and exemplar pruning. This paper is organized as following, after section 1 introduces the NNGE algorithm, the CCPS testbed, and the test datasets, section 2 surveys the state of art of the attempts to improve the NNGE. After that, section 3 introduces the FSDRM, then section 4 discusses the experimental results, finally, section 5 concludes the paper and draws the furfure work.

A. The NNGE

NNGE [7] 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, they may represent more than one example of the same class from the training database. Hyperrectangles are generalized rules which represent a class and single examples are previous examples of a class which do not fit into a hyperrectangle. 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 is detailed in [7].

B. The CPPS Testbed and Datasets

The datasets that we used consists of synchrophasor measurements from Phasor Measurement Units (PMUs) of four substations. As shown in the testbed diagram of Figure 1, 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. Operators can also manually issue commands to the IEDs R1 through R4 to trip the breakers BR1 through BR4. To enhance the cyberattack detection rate, the security attributes such as relay control panel SNORT [13, 14] 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 cyber-attacks. Details of the attributes have been introduced in previous work [5, 6, 11].

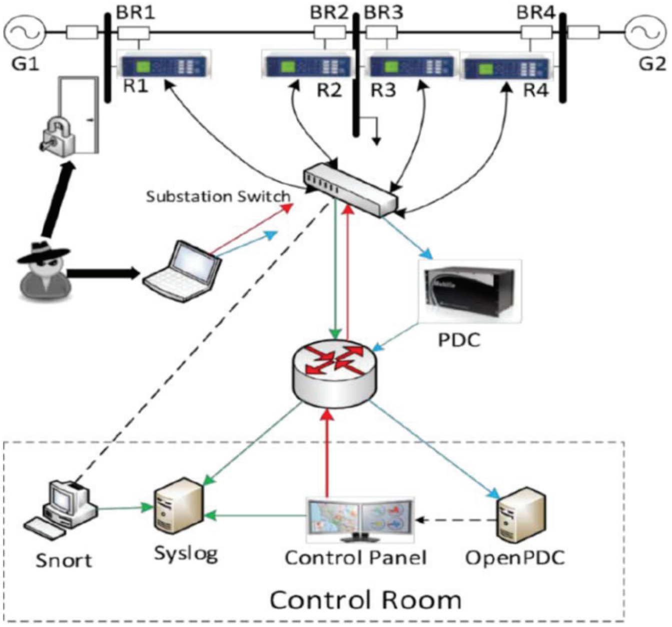

Fig. 1. Power System Framework for Generating Test Datasets [11]

II. State of the Art

There are several research works have been conducted to improve the classification accuracy of the NNGE algorithm, in this section, we briefly highlight them. Daniela et al [10] investigate the ability of an evolutionary pruning mechanism to improve the predictive accuracy of a classifier based on non-nested generalized exemplars. In [7], 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 non-generalized exemplars, and the presentation order of training instances. In [3] 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.

III. The Improved NNGE Algorithm

The improved NNGE algorithm uses our new FSDRM 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 two factors on the accuracy and computational performance of the NNGE. These factors are described later and summarized below in the following:

a) The reduction of the input features using a modified Particle Swarm Optimization (PSO) fitness function. The

PSO algorithm is used to compute the learning features weights and then rank the learning features according to their computed weights. The features with the highest weight 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 features weights of the weighted euclidean distance of the NNGE.

b) Pruning of non-generalized 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, but which are 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 hyprerectangle. The only drawback of the pruning algorithm is that the algorithm tends to produce rules that test a large number of input features that in turn hinders the scalability of the classification model. To this target, we use our FSDRM to reduce the input features and the NNGE hyperrectangles.

A. FSDRM Features 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 [11], 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 are 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 method that is based on the swarming strategies in fish schooling and bird flocking [9]. 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 *i*th 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 Eq. 1 and 2 [9]:

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

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

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

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

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

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

Where,

- $x_i(t)$ 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.
- A_i : is the feature subset of particle *i* at iteration *t*. i.e., $A=\{f1, f2, \ldots. f_{|A|}\}$,
- $|(A_i)|$: is the length of the feature subset without the non-value features,
- *n:* is the total length of the feature subset including the non-value features,
- 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.*

- $\emptyset, \theta$: 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 non value features, so we set them to $\emptyset = 0.9$, $\theta = 0.1$.
- γ_i: it denotes how much a certain feature *f* value *is* closer to a margin of the covering hyper-rectangle *H.*
- E_i is the conflicting example of particle *i*, it represents an example record of dataset that needs to be classified.
- H_i^{min}, H_i^{max}: 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 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 Eq. 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 criteria 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* is computed during the training phase are selected. 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, ..., En)* and a hyper-rectangle *H* as given in Eq. 6 [7, 10].

$$D(E,H) = \sqrt{\sum_{i=1}^{n} \left(w_i \frac{d(E_i, H_i)}{E_i^{max} - E_i^{min}} \right)} \qquad (6)$$

Where,

- N is number of features in the current Example *E.*
- E_j^{max}, E_j^{min}: They define the range of values over the training set which correspond to attribute *i*.
- H_i: is the interval $[H_i^{min}, H_i^{max}]$,
- d: is the distance between the features values and the corresponding hyper-rectangle "side" and it is computed according to Eq. 7.

$$d(E_i, H_i) = \begin{cases} 0, & H_i^{min} \le E_i \le H_i^{max} \\ H_i^{min} - E_i, & E_i < H_i^{min} \\ E_i - H_i^{max}, & E_i > H_i^{max} \end{cases} \qquad (7)$$

B. FSDRM's Hyperrectangles Reduction Using the Evolutionary Pruning Approach.

There are two main approaches to reduce the size of classifiers: pre-pruning and post-pruning. The pre-pruning approach aims to select the good training instances or prototypes and those aiming to select the relevant attributes. This is achieved by FSDRM through the feature reduction using the PSO algorithm. The post-pruning 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 to reduce its size and to improve 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 [10], 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 [10] 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*, e.g. 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 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 Eq. (8).

$$f(x) = \lambda Acc(\mathcal{H}(x)) + (1-\lambda)\frac{|\mathcal{H}| - |\mathcal{H}(x)|}{|\mathcal{H}|} \qquad (8)$$

Where,

- *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.
- |*H*| denotes the number of hyperrectangles.
- $\lambda \in (0, 1)$ is a parameter controlling the compromise between the two quality measures.

The population elements of the EPA-NNGE algorithm are evaluated using Eq. (8). 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 that instead of using Eq. (6), we will use Eq. 9.

$$D(E,H) = \sqrt{\sum_{i=1}^{n} \left(X_s w_i \frac{d(E_i, H_i)}{E_i^{max} - E_i^{min}} \right)} \qquad (9)$$

IV. Experimental Analysis and Results.

In our experiments, we evaluate the following:

1. The effectiveness of 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.
2. The impact of the NNGE's exemplar pruning 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. The experiments evaluate the reduction of the computational resources consumption in terms of: (a) number of reduced hyperrectangles and (b) number of ignored features that are defined by the pruning process in the training phase of the NNGE algorithm.

A. *Evaluate the Impact of The FSDRM Feature Reduction Using the New PSO Fitness function*

In these experiments, we evaluate the impact of the FSDRM feature reduction using the new PSO euclidean fitness function on the NNGE classification accuracy and the model size. We 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 Multi class 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 features for the three class datasets. Table 1 shows the top 10 features of each dataset.

TABLE 1: THE MOST SIGNIFICANT SELECTED FEATURES FITNESS VALUES IN THE BINARY, TRIPLE, AND MULTI CLASS DATASETS.

Binary Class			**Triple**			**MULTI**		
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

To test the accuracy of the selected features, we apply the NNGE classifier using Eq. 6 to compute the classification rates using the three datasets, see Table 2. In the following, we compare the output of our FSDRM 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 three datasets to compare the classification accuracy of these approaches against our approach.

TABLE 2: NNGE CLASSIFICATION RATE USING THE BINARY, TRIPLE, AND MULTI CLASS DATASETS

Feature Selection	**Binary Class**	**Triple CLASS**	**MULTI CLASS**
With (%)	98.38	98.01	94.03
Without (%)	65.42	66.41	23.66

Table 3 shows that FSDRM with the new PSO fitness function uses less number of features and outperforms the classification accuracy of the current classification algorithms.

TABLE 3: A COMPARISON BETWEEN THE FSDRM USING THE PSO ALGORITHM AND THE OTHER EXISTING APPROACHES

Approach	**Binary Class**		**Triple CLASS**		**MULTI CLASS**	
	No. of Features	Detection Rate (%)	No. of Features	Detection Rate (%)	No. of Features	Detection Rate (%)
FSDRM (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

B. *Evaluate the Impact of the FSDRM's Hyperrectangles Reduction using the EPA-NNGE Pruning Algorithm*

In these experiments, we evaluate the impact of the FSDRM hyperrectangles 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 as following, 17 features from the Binary dataset, 19 features from the Triple dataset, and 22 from the Multi class datasets. Since our main goal of our approach is to improve both 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 Eq. 8. The influence of the parameter λ on the accuracy and on the reduction of the model size is evaluated for the Binary class dataset as shown in Figures 2. According to our experments, the best values of λ that corresponds to the equilibrium point in the three datasets are 0.74, 0.53, and 0.44 respectively. To evaluate the impact of the pruning algorithm on the reduction of the model size, we consider both the hyperrectangles reduction ratio described before and the reduced number of features that are given in table 4.

An overall view of the accuracy gain ratio, hyperrectangles reduction ratio, and attributes reduction ratio for the Binary, Triple, and Multi Class Dataset are shown in Table 5 for the

FSDRM Hyperrectangles reduction using the EPA-NNGE and PSO vs. the traditional NNGE with CFS without the pruning capabilities. The accuracy gain is computed as *(Acc(FSDRM) −Acc(NNGE))/Acc(NNGE) * 100)*. The ratio of the reduced hyperrectangles is computed as $|H_{FSDRM}|/|H_{NNGE}|$ * *100*). The ratio of the reduced features is computed as (N_{FSDRM} / 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 Multi class dataset. This can be explained by the fact that this dataset has the hightest 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 Tripl dataset because it has the lowest accuracy gain (4.29%).

TABLE 4: A COMPARISON BETWEEN EACH FSDRM FEATURE AND THE TRADITIONAL NNGE WITH CFS

	Binary Class		**Triple CLASS**		**MULTI CLASS Dataset**	
Approach	No. of Features	Detection Rate (%)	No. of Features	Detection Rate (%)	No. of Features	Detection Rate (%)
FSDRM (PSO+EPA)	**15**	**99.21**	**14**	**98.91**	**18**	**97.03**
FSDRM (PSO)	17	98.38	19	98.01	22	94.03
NNGE (CFS)	28	94.68	28	94.62	28	87.77

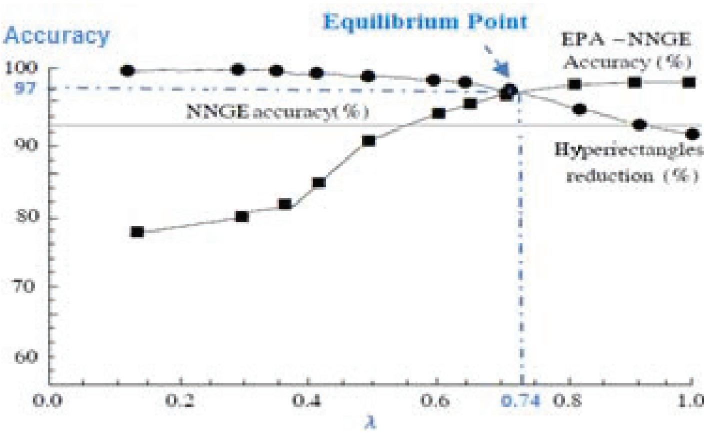

Fig. 2: Influence of λ on the EPA-NNGE accuracy gain and hyperrectangles reduction using the Binary class dataset.

TABLE 5: A COMPARISON BETWEEN THE FSDRM AND THE TRADITIONAL NNGE WITH CFS USING THE THREE CLASSES DATASETS

	Binary	**Triple**	**MULTI**
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**

V. CONCLUSION AND FUTURE WORK:

In this paper, we introduced the FSDRM, a feature selection and data reduction method to improve the detection accuracy, speed, and to reduce the computational resource consumption of the NNGE algorithm. FSDRM reduces the NNGE's hyperrectangles by pruning the non-generalized exemplars using the highest ranked features selected by the PSO algorithm with a new feature selection function. The experiments show that the NNGE using FSDRM outperforms the current classification techniques for the Multi, Binary, and Triple class datasets respectively as following; it outperforms 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 pruning the traditional NNGE examplers by 13.07%, 9.68%, and 7.41%. From the computational performance prospective, FSDRM's feature reduction and exemplar pruning reduce the hyperrectangles by 29.06%, 37.34%, and 26.76%.

For future work, we will evaluate the scalability, computational resource consumption, and the speed of the FSDRM. Furthermore, we will also study the influence of quantizing and clustering the input data of the STEM using a neural network model instead of using domain expert input data on the accuracy and computational performance of our approach.

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. PhD thesis, Mississippi State University, 2014.

[3] Nimmy C., Dhanya K.A, "Rule Induction using Ant-Miner Algorithm", Int. Journal of Scientific & Engineering Research, Volume 5, Issue 2, February-2014.

[4] Hisham A. Kholidy, Abdelkarim Erradi, Sherif Abdelwahed, Fabrizio Baiardi, "A risk mitigation approach for autonomous cloud intrusion response system", in Journal of Computing, Springer, DOI: 10.1007/s00607-016-0495-8, June 2016.

[5] U. Adhikari, T. H. Morris, and S. Pan, "Applying non-nested generalized exemplars classification for cyber-power event and intrusion detection," IEEE Transactions on Smart Grid, vol. PP, no. 99, pp. 1–1, 2016.

[6] U. Adhikari, Event and intrusion detection systems for cyber-physical power systems. PhD thesis, Mississippi State University, 2015.

[7] D Zaharie, L Perian, V Negru, A view inside the classification with non-nested generalized exemplars, IADIS European Conference on Data Mining, 2011

[8] Brent M, "Instance Based Learning: Nearest Neighbor with Generalization," University of Waikato, Hamilton, 1995

[9] Wang H, Sun H, Li C, et al. Diversity enhanced particle swarm optimization with neighborhood search.Information Sciences. 2013;223:119–135.

[10] Daniela Z., Lavinia P., Viorel N. and Flavia Z., "Evolutionary Pruning of Non-Nested Generalized Exemplars", 6th IEEE Int. Symposium on Applied Computational Intelligence and Informatics, May 19–21, 2011, Timişoara, Romania.

[11] Qian Chen, Hisham A. Kholidy, Sherif Abdelwahed, John Hamilton, "Towards Realizing a Distributed Event and Intrusion Detection System", the International Conference on Future Network Systems and Security (FNSS 2017), Gainesville, Florida, USA, 31 August 2017. Conference publisher: Springer.

[12] "Industrial control system (ics) cyber attack datasets, http://www.ece.uah.edu/~thm0009/icsdatasets/ PowerSystem_Dataset_README.pdf

[13] Hisham A. Kholidy, Fabrizio Baiardi, "CIDD: A Cloud Intrusion Detection Dataset for Cloud Computing and Masquerade Attacks", the 9th Int. Conf. on Information Technology: New Generations (ITNG), Las Vegas, Nevada, USA, 2012.

[14] Hisham A. Kholidy, Abdelkarim Erradi, Sherif Abdelwahed, Abdulrahman Azab, "A Finite State Hidden Markov Model for Predicting Multistage Attacks in Cloud Systems", in the 12th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC), Dalian, China, August 2014.