Annals of Telecommunications (2022) 77:457–471
https://doi.org/10.1007/s12243-022-00912-z

An extended assessment of metaheuristics-based feature selection for intrusion detection in CPS perception layer

Silvio E. Quincozes[1] · Diego Passos[2] · Célio Albuquerque[2] · Daniel Mossé[3] · Luiz Satoru Ochi[2]

Received: 17 March 2021 / Accepted: 28 March 2022 / Published online: 20 April 2022

Abstract

Cyber-physical systems (CPS) are multi-layer complex systems that form the basis for the world's critical infrastructure and, thus, have a significant impact on human lives. In recent years, the increasing demand for connectivity in CPS has brought attention to the issue of cyber security. Aside from traditional information systems threats, CPS faces new challenges due to the heterogeneity of devices and protocols. In this paper, we assess how feature selection may improve different machine learning training approaches for intrusion detection and identify the best features for each intrusion detection system (IDS) setup. In particular, we propose using F1-Score as a criteria for the adapted greedy randomized adaptive search procedure (GRASP) metaheuristic to improve the intrusion detection performance through binary, multi-class, and expert classifiers. Our numerical results reveal that there are different feature subsets that are more suitable for each combination of IDS approach, classifier algorithm, and attack class. The GRASP metaheuristic found features that detect accurately four DoS (denial of service) attack classes and several variations of injection attacks in cyber physical systems.

Keywords Cyber-physical systems · Intrusion detection · Feature selection · Metaheuristics

1 Introduction

Cyber-physical systems (CPS) are complex systems that integrate cyber components to physical processes through communication and computing technologies. Similarly to the Internet of Things (IoT) paradigm, tiny devices with sensing and actuation capabilities are attached to the daily objects to connect them to the internet and enable remote monitoring and control. These devices are placed at the *perception layer* and are commonly connected through wireless sensor networks (WSN) technologies. Although WSNs have been around for several years, they have recently been incorporated into CPSs. Through the *transmission layer* that involves other communication technologies such as Wi-Fi, the information collected from the physical world is sent to the *application layer*. This layer is responsible for running the cyber systems that typically use the acquired data for making decisions and sending commands back to the perception layer — often without human intervention [4].

As an integral part of so many critical infrastructure systems, it is very important to consider security aspects of CPS. Several Industrial IoT systems form the scope of CPS, including healthcare, manufacturing, transportation, chemical engineering, automotive systems, power, water, and the petroleum industry [12, 15, 22, 24]. Although security must be a concern at design time, real-world experiences demonstrate that it is infeasible to predict — and prevent — all security issues. In the last decade, several harmful attacks were reported. In 2010, 900 MW were lost during 7 s of an attack targeting a power plant [17]. Late, the Stuxnet worm attacks caused a serious deterioration in Iran's nuclear program. In the worst cases, even human life may be at risk.

✉ Silvio E. Quincozes
sequincozes@ufu.br

Diego Passos
dpassos@ic.uff.br

Célio Albuquerque
celio@ic.uff.br

Daniel Mossé
mosse@pitt.edu

Luiz Satoru Ochi
satoru@ic.uff.br

[1] Universidade Federal de Uberlândia, Monte Carmelo, Brazil

[2] Computer Science Department, Universidade Federal Fluminense, Niterói, Brazil

[3] Computer Science, University of Pittsburgh, Pittsburgh, USA

Malicious remote control of cars and airplanes has already been successfully demonstrated [17]. Currently, similar vulnerabilities are still being reported: In 2020, TrapX Security reported a malware found on several automatically guided vehicles [21].

The increased connectivity between the physical and cyber domains is at the core of the unique security challenges in CPS [4]. Whereas the increasing attacks and vulnerabilities targeting CPSs can have catastrophic consequences [14], the perception layer is the most critical point of a CPS application. Computational devices in this layer have direct access to and control of physical objects. Besides, these devices are often the most vulnerable targets due to their resource-constrained characteristics [4].

Intrusion detection systems (IDS) can be a crucial component to detect malicious activity in CPSs. Nevertheless, their effectiveness depends upon the features being analyzed: a useless feature subset may compromise the IDS results. Feature selection (FS) methods aim at reducing the number of features analyzed by an IDS, discarding irrelevant or redundant ones, thus improving the IDSs' detection performance. As an added benefit, FS reduces the feature space, making the data processing for online intrusion detection faster. Whereas intrusion detection is a time-sensitive procedure, the FS may occur off-line (at design or deployment time). Once the proper features to detect attacks for a specific scenario are defined, they can be used by an IDS to perform real-time analysis. Moreover, the selected features may be updated at any time (i.e., the FS process can be executed in parallel with intrusion detection itself to enable further improvements). Thus, while certain FS methods are less time-consuming than others, for this particular application, their effectiveness should be the main concern.

In our previous work [19], we demonstrated that our adapted version of the greedy randomized adaptive search procedure (GRASP) for FS, namely GRASP-FS, focused on the perception layer, outperforms traditional filter-based approaches to detect three different attacks: *flooding*, *blackhole*, and *grayhole*. For each attack, a different specialized feature subset was found by GRASP-FS. In this work, we study the feasibility of building generic classifiers through two approaches: *binary* and *multi-class* classifications. We compare these approaches to expert classifiers using four different performance metrics. Additionally, we propose using F1-Score as an alternative metric for GRASP-FS instead of the traditional *accuracy* metric. We show that this enables a better choice of an optimized feature subset, especially when using an unbalanced dataset such as the WSN-DS [3]. Finally, we consider two attacks have not yet been assessed in published literature, namely (i) the time division multiple access (TDMA) scheduling attack (or simply TDMA attack) and (ii) the injection attack. Our main contributions can be summarized as follows:

- The first evaluation of injection attacks with GRASP-FS, in an empirical analysis based on data from actuators of a water treatment CPS [10];
- A novel use of F1-score metric for feature selection criteria, making it more robust and improving the GRASP-FS algorithm;
- The first evaluation of the TDMA attack with GRASP-FS, in an empirical analysis based on data collected from a WSN [3];
- The identification of features for blackhole, grayhole, flooding, TDMA, and injection attacks with a discussion of the domain-related meaning of the features as well as their potential to improve intrusion detection;
- An analysis of how an IDS is affected by binary and multi-class classification, compared to classifiers specialized for individual attacks; and
- The insight that expert and multi-class IDSs are good alternatives to replace the binary classification and reach better detection performance.

In the remaining of this work, we provide an overview of the perception layer on CPS in Section 2, discuss the related work in Section 3, present the GRASP-FS metaheuristic in Section 4, describe our evaluation methodology and the GRASP-FS parameters and metrics in Section 5, analyze the experimental results in Section 6, and, finally, present our conclusions in Section 7.

2 Cyber-physical systems perception layer

The traditional three-layer CPS architecture is presented in Fig. 1 [19]. The *perception layer* of a CPS is composed of physical objects equipped with tiny devices that have sensing, actuation, computation, and networking capabilities. The main novelty of CPSs, compared to traditional information systems, is the addition of perception layer devices that collect physical world information (e.g., sensors and RFID tags); the main novelty in CPSs is the ability of these devices to change the physical environment they control (e.g., actuators). Many of the devices are characterized by their resource-constrained nature, with limited power supply. Therefore, they typically transmit the collected data through the *transmission layer* to control centers responsible for handling physical processes. These control centers, placed at the *application layer*, may respond with commands to be executed by the perception layer to perform changes into the physical processes [4, 24].

The physical and cyber worlds may connect through both infrastructure and infrastructure-less network technologies, such as Wi-Fi, 3G/4G/5G, Bluetooth, and Zigbee. Therefore, the transmission layer may also involve heterogeneous technologies and communication protocols.

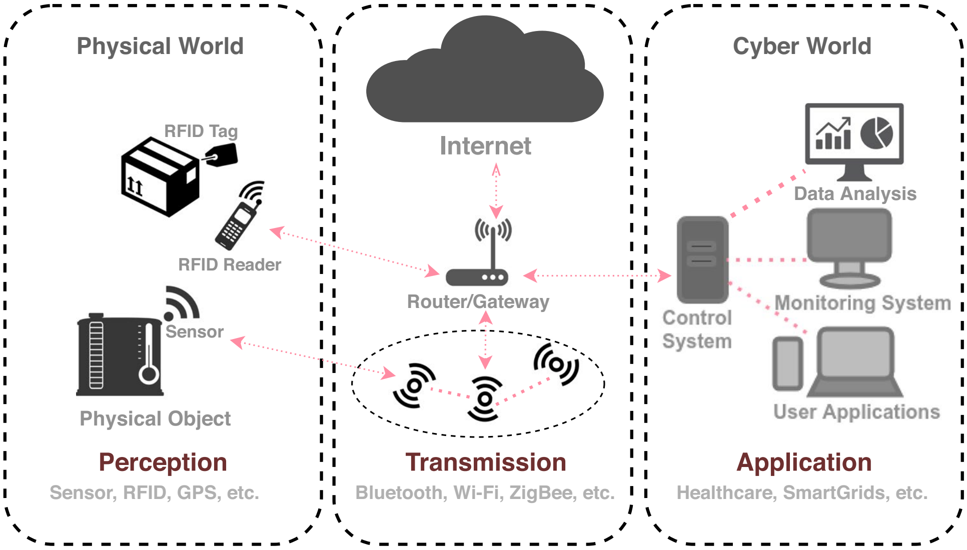


Fig. 1 A typical three-layer CPS architecture (extracted from [19])

2.1 Security issues and threats on the perception layer

The interaction between cyber and physical domains implies that the system is vulnerable to different attacks at each layer. Applications must be protected and monitored to avoid attackers gathering sensitive data, as well as controlling the physical-world behavior. Similarly, the data transferred in the transmission layer must be protected to avoid message manipulation and unauthorized access. However, the most critical point is the perception layer because the devices are often attached to the physical objects: once they are compromised, attackers may interact directly with the physical world. Thus, security measures in the upper layers may not be effective if the attackers have direct access to the perception layer.

Perception layer attacks include unauthorized access, information manipulation, information disclosure, tracking, tampering, and availability compromising. In many cases, the resource-constrained devices that compose the infrastructure at the CPS perception layer are vulnerable to injection attacks. In [10], 36 injection attacks are launched targeting different sensors and actuators of a CPS water treatment system. Moreover, even if lightweight security solutions are used, availability may still be compromised by denial-of-service (DoS) attacks [3].

In addition to these attacks, while confidentiality is considered the most relevant security property for traditional information systems, in CPS, *availability* is often considered the most important requirement (due to the need for constant monitoring and actuation), followed by integrity, confidentiality, and authenticity [4, 12].

If the perception layer is unable to reliably collect information from the physical world (e.g., if a sensor's energy is exhausted during a flooding attack), the entire CPS may be compromised. This means applications will not be aware of the systems' states. Consequently, they will not be able to make the proper response actions to match changes in the state of physical processes (e.g., sounding a fire alarm, administering insulin to patients with low glucose levels, reducing the load on an overheated machine). Indeed, there are current efforts in the literature that focus on providing situational awareness to control the infrastructure being monitored by Industrial WSNs [1, 2].

Common DoS attacks to the perception layer include grayhole, blackhole, flooding, and TDMA. We illustrate these attacks following in [3, 18], where attacks are explored through a simulated WSN with the Low Energy Aware Cluster Hierarchy (LEACH) routing protocol. This protocol assumes a fixed base station (BS) and clusters sensor nodes (see Fig. 2). Each sensor cluster has a special node called cluster head (CH), responsible for aggregating the data received from the other sensor nodes within its cluster and for transmitting them to the BS. Four important

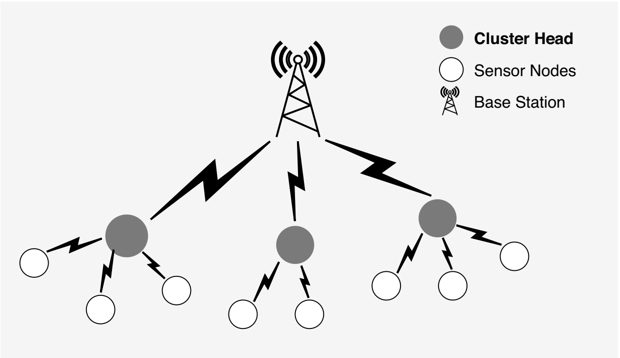


Fig. 2 A typical WSN topology (extracted from [19], and based on [3])

aforementioned perception layer attacks are described below (in addition to the injection attack).

Flooding Flooding attacks targeting the LEACH protocol consist of random nodes sending a large number of *advertising cluster head* messages with high transmission power. This causes sensors to spend more time and energy trying to decide which CH to join. Another consequence is that the victim may choose the malicious node as CH, enabling the attacker to perform further attacks.

Grayhole Grayhole attacks are also know as *selective forwarding attacks*. To perpetrate them, the attacker performs selective or random packet discard. This action aims at dropping legitimate packets, so that they are not delivered to the BS. To do so, attackers must be selected as CH by the legitimate nodes — or they must compromise a legitimate CH. As a CH, the attacker receives data packets from other nodes and can drop them. Selective discard may be performed according to the sensitivity of data carried by the packet, leading to amplification of the damage to the victim and helping hide the attack.

Blackhole Blackhole attacks are similar to grayhole, except that packet discarding is not selective. Instead, blackhole attacks drop every packet.

TDMA Scheduling TDMA attacks occur during the setup phase of LEACH. The main goal of the attacker is to cause packet collisions that leads to data loss. To do that, an attacker should act as a CH and then assign all nodes the same time slot to send data. Therefore, such attacks occur when CHs set up TDMA schedules for the data transmission time slots in a malicious way.

3 Related work

The greedy randomized adaptive search procedure (GRASP) [20] is an iterative multi-start metaheuristic proposed originally to achieve an approximate solution for graph planarization problems. It has since been generalized to solve other combinatorial optimization problems in different domains [23]. The general GRASP strategy can be carried in two iterative phases: *construction* and *local search*. These steps are repeated until a stop criterion is reached (e.g., the maximum number of total iterations). For each iteration, a different random seed solution is generated. The construction phase relies on this seed to generate a restricted candidate list (RCL) and choose from it a randomized greedy solution. The RCL consists of a predefined number of candidates to compose a feasible solution. Typically, this initial greedy solution can be improved by a local search phase, that is, the greedy solution is the starting point for a neighborhood exploration.

Yusta [23] demonstrates that GRASP can outperform Sequential Forward Floating Selection (SFFS), Tabu Search, and Genetic and Memetics algorithms in generating the best feature subset to classify instances from different databases (Spambase, Waveform, Ionosphera, Vehicle, Wincosin, and German). Esseghir and Amir [9] also apply GRASP for FS considering some of those datasets (Ionosphere and SpamBase) and others (Sonar, Audiology, and Arrhythmia). Whereas the former work used the K-Nearest Neighbors (KNN) classifier for wrapping evaluations, the latter explored artificial neural networks — neither compares the used classifier with alternative algorithms.

Bermejo et al. [5] use GRASP to deal with large and high-dimensional datasets with focus on reducing the number of wrapping evaluations (i.e., those that rely on machine learning algorithms) by using the Incremental Wrapper Subset Selection (IWSS) algorithm in the construction phase. The main drawback of this approach is the possibility of excluding important features due to a premature stop caused by the current solution exceeding the solution cardinality threshold.

Moshki et al. [16] also use GRASP to deal with high-dimensional and large datasets. They combined GRASP with an extended version of simulated annealing as a local search procedure to introduce new parameters that allow the implicit weighting between accuracy and execution time. However, this method is limited to the implicit trade-off control between these parameters. It is still unable to control it explicitly. Besides, using accuracy metric is not suitable for unbalanced datasets because it gives the same weight for all classes (e.g., attacks and benign).

Diez et al. [8] propose GRASP Forest (G-Forest) for constructing ensembles of decision trees. This method uses concepts of GRASP for both feature selection and choosing splitting points at each tree node. For selecting features for each level of the tree, G-Forest assembles an RCL composed of all features with an *information gain* above a certain threshold. It then proceeds to choose features randomly from this RCL. Once a feature is chosen, G-Forest computes the IG of each possible split point for that feature and forms a new RCL following the same strategy (i.e., using those features with an IG above a certain threshold). The split point is then chosen randomly from this list. This procedure is repeated until the desired number of trees is created. Diez et al. [8] use only the construction procedure of GRASP.

In a later work, Diez et al. [7] extend their proposed G-Forest to create an Annealed Randomness Forest (GAR-Forest). The key idea is to introduce a parameter that controls the randomness during the solution construction phase, which ranges from an entirely random procedure to a totally greedy one. The authors consider 62 datasets (including Waveform, Ionosphere, and Vehicle). However, none of those is related to the cyber security domain. Kanakarajan and Muniasamy [13] apply GAR-Forest to detect DoS, Probe, R2L, and U2R attacks in traditional networks. The reported results reveal that F1-Score and accuracy are both slightly over 85%, which means that performance can (and should) be improved.

In our previous work [19], we experimented with an adapted version of GRASP for FS, namely GRASP-FS. We used GRASP-FS to select three feature subsets for building expert IDSs for the flooding, blackhole, and grayhole attacks (each IDS was specialized in one attack class). The results reveal that the selected features outperform those selected by traditional filter-based approaches to detect attacks targeting the CPS perception layer. In this work, we cover experimentation involving binary or multi-class classification, and we extend the attack coverage by including another attack from WSN-DS dataset (i.e., TDMA attacks) and injection attacks from the SWaT CPS dataset not previously studied with GRASP-FS.

Table 1 shows a summary of the discussed work. Clearly, the use of the GRASP metaheuristic has been frequently considered in the literature for FS. However, this approach is still seldom studied in the domain of intrusion detection. Thus, before our work, it is still unclear how much can GRASP contribute to select features for the state-of-the-art machine learning–based IDSs.

4 Proposed GRASP feature selection algorithm

This section describes the proposed specialization of the GRASP algorithm [19] to the feature selection problem. In particular, we adapted the generic GRASP approach to select features in the CPS perception layer. The adaptations of GRASP presented herein include the solution composition (features subset in our case), fitness function choice, neighborhood structure, and RCL generation. The implemented GRASP-based FS is shown in Algorithm 1.

Algorithm 1 GRASP Algorithm Adapted to FS.

```
   input : fSet // full features set
           maxIt // number of iterations
           rclSize // RCL number of features
           numF // feature subset size
   output: bestFS // best feature subset
 1 begin
 2     rcl ← ∅ // restricted candidate list
 3     bestFS ← ∅
 4     bestFS.F1Score ← 0
 5     while numIterations++ < maxIt do
 6         greedyFS ← construct(numF, rcl, rclSize, fSet)
 7         greedyFS.F1Score ← evaluate(greedyFS)
 8         bestLocalFS ← localSearch(greedyFS, rcl)
 9         if bestLocalFS.F1Score > bestFS.F1Score then
10             bestFS ← bestLocalFS
11         end
12     end
13 end
14 return bestFS
```

The input parameters are the full feature list (`fSet`), the maximum number of GRASP iterations (`maxIt`), the number of features to compose the RCL (`rclSize`), and the desired feature subset size (`numF`). For each GRASP iteration, a greedy and randomized feature subset (`greedyFS`) is generated from the RCL features. This feature subset is further used as a seed to the local search procedure. At the end of the iteration, the best solution found so far among all construction and local searches iterations is saved (`bestFS`). The construction (Line 5) and local search (Line 7) phases are detailed in Algorithms 2 and 3, respectively.

4.1 Construction phase

In the construction phase, we employ the Gain Ratio (GR) [11] algorithm for feature ranking and RCL creation (Lines 2–8 of Algorithm 2). Note that once the feature list is ranked (Line 2), this step is skipped in the next iterations. The `greedyFS` is constructed incrementally by choosing random features from the RCL until the desired number of features is achieved (Lines 9–11).

Table 1 Comparison of related works based on GRASP

Ref.	Attacks	Obj. function	Classifier	Approaches	Datasets
[5]	None	Accuracy	Naive Bayes	Multi-class	Multiple
[7]	None	Accuracy	J48	Multi-class	Multiple
[8]	None	Info. gain	None	Multi-class	Multiple
[9]	None	Error rate	KNN, ANN, Naive Bayes	Multi-class	Multiple
[13]	DoS, Probe, R2L, U2R	Accuracy	GAR-forest	Binary, multi-class	NSL-KDD
[16]	None	Accuracy	MATLAB Functions	Multi-class	Multiple
[19]	Blackhole, grayhole, flooding	Accuracy	J48, Naive Bayes, REP Tree, R. Forest, R. Tree	Expert	WSN
[23]	None	Accuracy	None	Multi-class	Multiple
This work	Blackhole, grayhole, flooding, TDMA injection	F1-Score	J48, Naive Bayes, REP Tree, R. Forest, R. Tree	Binary, multi-class, expert	WSN and SWaT

Algorithm 2 CONSTRUCTION PHASE.

```
   input : fSet // full features set
           rcl // restricted candidate list
           numFeatures // number of features to select
           rclSize // RCL number of features
   output: greedyFS // Greedy Random Feature Set
 1 begin
 2     if rcl = ∅ then
 3         foreach feature ∈ fSet do
 4             feature.GR ← calcGR(feature)
 5         end
 6         rcl ← selectTopRanked(fSet, rclSize)
 7     end
 8     greedyFS ← ∅
 9     while (|greedyFS| < numFeatures) do
10         greedyFS ← ∪ selectRandom(feature ∈ rcl)
11     end
12 end
13 return greedyFS
```

4.2 Local search phase

In the local search phase (Algorithm 3), we employ the Bit-Flip [6] neighborhood structure to perform iterative simple movements from the seed feature subset. This is repeated for a fixed number of iterations (maxIt), determined by the user. Each generated neighbor feature subset (`nbFS`) is evaluated through the selected classifier algorithm (line 6). In our previous work [19], the feature subsets' quality is assessed by the classifiers' accuracy (`acc`). However, we found that it may be an improper metric specially considering unbalanced datasets. Therefore, to overcome this weakness, in this work, we propose the use of the F1-Score metric to replace the accuracy. The local search procedure is also repeated at most `maxIt` iterations.

The GRASP-FS algorithm relies on a decision criterion to compare the found solutions and select the best ones. In contrast to the previous GRASP implementations [19], we choose the F1-Score as a metric to base the feature selection. This metric is defined as:

$$F1Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

F1-Score is a commonly used metric in machine learning. It is computed using true positives (TP), false positives (FP), and false negatives (FP) and can be derived from precision (i.e., $\frac{TP}{TP+FP}$) and recall (i.e., $\frac{TP}{TP+FN}$). As true negatives (TN) are not a factor in F1-Score, this metric is immune to biases introduced by a large imbalance toward normal instances (something common in security datasets), contrary to what happens with different metrics such as accuracy, which is defined as:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{2}$$

Thus, in the context of IDS, F1-Score focuses on the classifiers' ability to detect the specific attack classes (e.g., injection, TDMA, grayhole, and flooding attacks). For this reason, we selected F1-Score to replace accuracy.

Algorithm 3: LOCAL SEARCH.

```
   input : rcl // restricted candidate list
           greedyFS // Seed feature subset
           maxIt // Number of iterations
   output: bestLocalFS // Best feature set found
 1 begin
 2     bestLocalFS ← greedyFS
 3     numLSIterations ← 0
 4     while numLSIterations++ < maxIt do
 5         nbFS ← bitFlipt (greedyFS, rcl)
 6         nbFS.F1Score ← evaluate(nbFS)
 7         if nbFS.F1Score > bestLocal.F1Score then
 8             bestLocalFS ← nbFS.F1Score
 9         end
10     end
11 end
   return : bestLocalFS
```

5 Methodology

Some IDSs typically have a two-step process to support the detection of attacker activities. First, known attacks and benign traces should be used to *train* the IDS. Then, unknown instances are analyzed by classifier algorithms which attempt to *classify* (i.e., to set a class label) them into the same class as the similar training instances. Each classifier algorithm may use its own particular methodology and metrics. Therefore, to assess the best manner to use GRASP for FS, we explore its impact on selecting features for different training strategies. The configurations for our experiments are shown in Table 2.

The literature has demonstrated that GRASP usually selects different feature subsets for each classifier algorithm [9, 19]. This is because, according to the classification method employed and the attacks being analyzed, particular features are more important than others. We further assume that the training process and the set of target classes may affect the intrusion detection performance. Therefore, our methodology consists of comparing three different approaches for training classifier algorithms on datasets that include normal and attack traffic:

- *Binary classification:* all attack instances are simply labeled as `Attack`, regardless of the type of attack. Thus, the classifier algorithms are trained and tested by their ability to determine whether each instance belongs to the `Normal` class or (some) `Attack` class.
- *Multi-class classification*: attack instances are labeled with specific labels for each attack. Thus, the classifier algorithms are assessed for their ability to determine whether or not a instance is `Normal` and, if it is not, which particular attack it represents.
- *Expert classification*: each classifier is trained and tested with instances from the normal class and from a single attack class. Thus, five different datasets are built and assessed giving origin to five expert IDSs, one for each kind of attack.

Table 2 The chosen parameters for our methodology

Parameter	Value
Number of folds for cross-fold validation	5
Number of features to select (*N*)	5
Number of features to compose the RCL	20
Algorithm for RCL generation	Gain Ratio (GR)
Objective function (decision criteria)	F1-Score
Datasets	WSN-DS dataset, SWaT dataset
Classifier algorithms	J48, NaiveBayes, Random Forest, Random Tree, REP Tree
Training approaches	Expert, binary, multi-class (5 experts)
Experimentation scenarios	Expert: 25 scenarios, binary: 5 scenarios, multi-class: 5 scenarios
Attacks	Blackhole (WSN-DS), grayhole (WSN-DS), flooding (WSN-DS), TDMA (WSN-DS), injection (SWaT)

We use two datasets that can represent the perception layer from a CPS: (i) Secure Water Treatment (SWaT [10]) is a water treatment dataset for cybersecurity research; and (ii) WSN-DS dataset [3], which is based on a wireless sensor network. The SWaT dataset contains 51 features, but the RCL generation in the construction phase of GRASP-FS selects the top 20 features based on Gain Ratio method (see Section 4). The top 20 SWaT features are listed in Table 3. The 18 features that compose the WSN-DS dataset are listed in Table 4. Note that as the RCL length is 20 and since WSN-DS has only 18 features, all WSN-DS features are included in the RCL.

We set up GRASP to select $N = 5$ features to compose a reduced subset that is used to feed multiple algorithms. We use J48, Naive Bayes, Random Forest, Random Tree, and REP Tree as classification algorithms. For each training approach (i.e., binary, multi-class, flooding expert, grayhole expert, blackhole expert, scheduling expert, and injection expert), each classifier is assessed individually. Thus, the total number of experimentation scenarios is 35 (i.e., 5 classification algorithms × 7 classifier training approaches). Each scenario is evaluated by means of a 5-fold cross-validation. This means that each full dataset is segmented into 5 parts: each of them is used as testing dataset while the other 4 parts are used for training. This procedure is repeated for each part until all of them are used as testing dataset once. At the end of these 5 runs, we compute the average of the results.

In addition to using F1-Score as decision criteria, we also examine whether other metrics are improved by using GRASP as an FS method. Given that F1-Score is highly dependent on TP, we note that the TP is also highly dependent on the way that multi-class results are computed. In particular, there are 2 ways to extract multi-class results:

- *Multi-class as binary*: this method classifies as TPs all attack instances classified into any attack class, whereas normal instances classified as attack are considered FNs;
- *Multi-class specific*: this method classifies as TPs only the attack instances classified in the correct class that represents its specific type of attack; any other attack instance classified in a different class than expected (including being classified as normal) is considered a FN.

We assume the second approach to analyze the multi-class results as presented in Section 6.5. Nevertheless, we consider the confusion matrix to discuss the classifiers' ability to distinguish among all the experimented classes. Lastly, in Section 6.6, we compare and discuss the results for both *multi-class as binary* and *multi-class specific* methods.

6 Experiments

To assess the hypotheses that each classifier training approach and algorithm has a different optimized feature subset, we experiment the GRASP-FS algorithm with different training approaches (i.e., binary, multi-class, and expert classifiers).

These approaches are assessed through five classifier algorithms, as discussed in Section 5. The number of times that each feature is selected for each approach, considering each of the 5 folds used for testing each classifier, is shown in Section 6.1 for the SWaT dataset and Section 6.2 for the WSN dataset.

We chose `F1-Score` as the main metric to assess the classifiers' performance. We also computed `accuracy`, `precision`, and `recall`. The average from these metrics is computed through 5-fold cross-validation. Each of the 5 folds/segments in each dataset may have a different number of attacks (see Table 5) due to the random choice of instances to compose each fold.

6.1 Results for SWaT dataset

We first analyze the SWaT CPS dataset. This dataset has a total of 36 variations of injection attacks. All these attacks

Table 3 Top 20 SWaT perception layer features (based on SWaT [10])

Index	Id	Device	Description
34	UV-401	Actuator	Dechlorinator
43	P-501	Actuator	Pump
31	P-402	Actuator	Pump
5	P-102	Actuator	Pump
14	P-204	Actuator	Dosing pump
16	P-206	Actuator	Dosing pump
23	MV-304	Actuator	Motorized valve
25	P-302	Actuator	UF feed pump
21	MV-302	Actuator	Motorized valve
40	FIT-502	Sensor	Flow meter
29	LIT-401	Actuator	Level transmitter
13	P-203	Actuator	Dosing pump
4	P-101	Actuator	Pump
10	MV-201	Actuator	Motorized valve
41	FIT-503	Sensor	Flow meter
6	AIT-201	Sensor	Conductivity analyzer
42	FIT-504	Sensor	Flow meter
46	PIT-502	Sensor	Pressure meter
39	FIT-501	Sensor	Flow meter
19	LIT-301	Sensor	Level transmitter

can be classified into the same class due to the similar behavior they have (i.e., to inject malicious data or commands). As this dataset contains a single attack class, it is not possible to perform a multi-class analysis. Therefore, we present only an expert IDS specialized to injection attacks — which can also be considered a binary analysis in this case.

6.1.1 Injection results

For the SWaT injection attacks, GRASP-FS selected the following features most often: sensors `FIT-501` (100%) and `LIT-301` (80%), and actuator `P-101` (60%). Besides, 8 other features were selected less frequently. Thus, among the 51 features, 20 were chosen to limit the RCL and, among these, only 11 were ever used to compose the best 5-length solution for at least one classifier. Figure 3 shows these results.

The Random Tree classifier reached the highest F1-Score (96.97%) for injection attacks from the SWaT dataset. It used the feature subset composed of `P-101`, `AIT-201`, `LIT-301`, `P-302`, and `FIT-501`. It correctly detected 3,153.4 attack instances with only 92.2 false positives on average for a 5-fold cross-validation. Random Tree was a close second, and has precision of 97.16%, recall of 96.78%, and accuracy of 99.65%. When considering the 95% confidence interval (shown as error bars in the figure), all classifiers' F1-Scores are not statistically equivalent.

6.2 WSN-DS dataset

In general, the results confirm the initial hypotheses: each experimentation run selected a particular feature subset. This

Table 4 WSN perception layer features (based on WSN-DS [3])

Index	Acronym	Description
1	`Id`	Unique sensor node identification
2	`Time`	Current simulation time of the node
3	`IsCH`	Flag to distinguish CH and normal sensor
4	`WhoCH`	The identification of the current CH node
5	`DistToCH`	Distance between the node and its CH
6	`AdvS`	Number of advertising broadcast `sent` (S) by CH
7	`AdvR`	Number of advertising CH messages `received` (R) from CHs
8	`JoinS`	Number of join request sent to CH
9	`JoinR`	Number of join request received by the CH
10	`SchS`	Number of advertising TDMA messages sent
11	`SchR`	Number of TDMA messages received from CH
12	`Rank`	Order of the node within the TDMA schedule
13	`DataS`	Number of data packets sent to CH
14	`DataR`	Number of data packets from CH
15	`DataStoBS`	Number of data packets sent to the BS
16	`DistCHtoBS`	Distance between the CH and the BS
17	`SendCode`	Cluster sending code
18	`Energy`	Energy consumption in the previous round

Table 5 The average number of instances per fold for each class in WSN-DS and SWaT datasets (each fold represents 1/5 of the total instances)

Dataset	Class	Average number of instances
WSN-DS	Grayhole	2919.2
	Blackhole	2009.8
	Flooding	662.4
	TDMA	1327.6
	Normal	68013.2
SWaT	Injection	3258.4
	Normal	53544.8

happens because each attack may affect specific features and each classifier uses these features in a different manner. Nevertheless, there are common features that enable the most classifiers to detect multiple classes of attacks, such as the number of advertising broadcast sent by CH (`AdvS`) — selected by GRASP-FS for 76.67% of the 30 experimentation runs related to WSN-DS [3] dataset, as described in Section 5. In the following subsections, we further detail those results.

Table 6 shows the number of times each feature was selected by each classifier (therefore maximum of 5) from WSN-DS dataset [3]. The last column shows the frequency that each feature is selected in the solution over the 30 experimentation runs (i.e., 5 classifiers × 6 training approaches—recall that injection expert applies only for the SWaT dataset).

According to Table 6, the number of advertising TDMA messages sent (`SchS`) is a particular feature selected by GRASP-FS for all expert classifiers specialized to detect flooding attacks. On the other hand, when trained to detect other specific attacks, this feature is not among the top five selected features for these classifiers. This shows that some features are important to represent a single attack class whereas they are less relevant (or not relevant) to generalize the attacker behavior considering other classes. Therefore, our analysis reveals that feature selection methods such as GRASP-FS should consider attack classes separately to improve performance.

6.3 Results for WSN-DS with expert IDSs

6.3.1 Grayhole results

Among the expert IDSs for detecting the grayhole attack, the most selected features were `AdvS` (100%), `DataStoBS` (80%), `DistCHtoBS` (60%), and `Id` (60%). Among these features, `AdvS` and `DataStoBS` features are indicators of the data volume sent to other sensor nodes and to BS, respectively.

In Fig. 4, we show performance metrics. Naive Bayes reached 100% of recall to detect grayhole attacks, correctly classifying an average of 2,919.2 attack instances across the 5-fold cross-validation. However, because Naive Bayes presented 1,832.8 false positives, on average, its precision was severely affected (61.43%), resulting in an F1-Score of 76.11%. On the other hand, the Random Forest classifier detected fewer attacks (an average of 2,894) but presented an average of only 45.4 false positives, resulting in an average precision of 98.46%.

The top-5 features that compose the subset selected by GRASP-FS for the Random Forest specialized classifier were `WhoCH`, `AdvS`, `DataR`, `DataStoBS`, and `DistCHtoBS`. These features enabled Random Forest to reach the best result in terms of F1-Score (98.80%). Among them, `DataR` and `DataStoBS` can be used as indicators to detect the amount of data received but not sent by an attacker, thus, characterizing the grayhole results.

6.3.2 Blackhole results

When used with expert classifiers for the blackhole attack, GRASP-FS selected the following features for most classifiers: `Time` (80%), `DistCHtoBS` (80%), `IsCH` (60%), and `DataR` (60%). The distance to the BS may help to identify a malicious node running blackhole attacks, similarly as described for grayhole.

Blackhole attacks are easily detected by all classifiers. As shown in Fig. 5, all metrics are above the 99.5% for all classifiers. Considering the 95% confidence intervals, Random Forest and Random Tree obtained statistically equivalent F1-Scores. All classifiers obtained equivalent

Fig. 3 Results from the SWaT expert scenario (5-fold average)

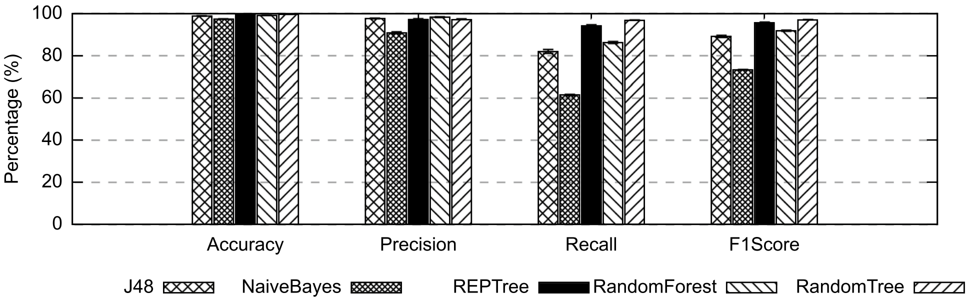


Table 6 Features ranked by the frequency in which they are selected

Feature	Expert classifiers				Generic		Freq.
	G. hole	B. hole	Flooding	TDMA	Bin.	Mult.	
AdvS	5	2	5	3	3	5	76.67%
Time	2	4	5	5	2	3	70.00%
Id	3	2	4	2	3	2	53.33%
DistCHtoBS	3	4	0	4	3	2	53.33%
DataStoBS	4	1	2	0	4	4	50.00%
WhoCH	2	2	3	1	2	3	43.33%
AdvR	1	2	3	1	2	3	40.00%
IsCH	1	3	0	0	2	1	23.33%
Energy	2	0	0	0	3	2	23.33%
SchS	0	0	0	5	1	0	20.00%
DataR	1	3	0	1	0	0	16.67%
JoinR	1	0	0	3	0	0	13.33%
SchR	0	0	2	0	0	0	6.67%
JoinS	0	2	0	0	0	0	6.67%
DataS	0	0	1	0	0	0	3.33%
Rank	0	0	0	0	0	0	0.00%
DistToCH	0	0	0	0	0	0	0.00%
SendCode	0	0	0	0	0	0	0.00%

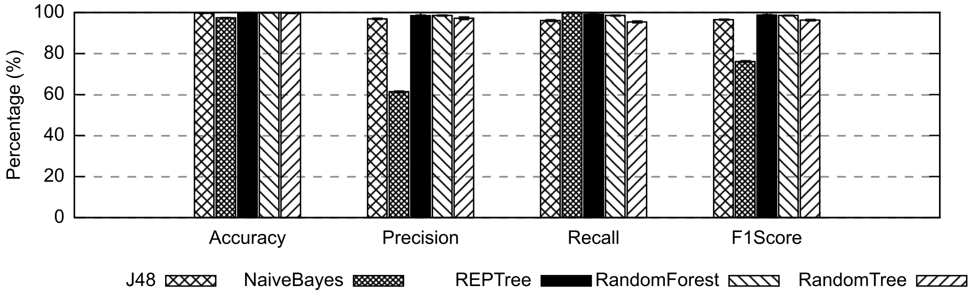


Fig. 4 Results from the gray-hole expert scenario (5-fold average)

accuracy. J48, Random Forest, and Random Tree obtained equivalent precision. Regarding recall, all classifiers except Random Forest are statistically equivalent (Random Forest is slightly worse, but still good).

6.3.3 Flooding results

For the expert classifiers targeting the flooding attack, the following features were selected most frequently: Time (100%), AdvS (100%), Id (80%), and WhoCH (80%). The timestamp and the number of advertising messages sent by the sensors were selected by GRASP-FS to train the expert classifiers for this attack. These features are representative of this attack class due to the high number of messages sent in the same LEACH round. The sensor node identifier and the CH identifier are also important features to detect the origin of the flow of attack messages. The former can reveal which is the malicious/infected node. The latter can reveal which CH is the potential malicious/infected node.

When analyzing performance metrics, the feature subset composed of Time, WhoCH, AdvS, AdvR, and SchR enabled the J48 classifier to reach the best F1-Score (96.17%) when trained to detect flooding attacks. On average, for a 5-fold cross-validation, 647.4 attack instances were detected correctly whereas 36.4 false positives were found. Thus, J48 presented a precision of 94.68% and a recall of 97.72%. The results from the perspective of these metrics are shown in Fig. 6. Although the error bars are not easily seen, it is important to note that, considering a confidence interval of 95%, J48, REP Tree and Random Tree obtained statistically equivalent F1-Scores.

6.3.4 TDMA results

For the TDMA expert classifiers, GRASP-FS selected the following features most often: Time (100%), SchS (100%), DistCHtoBS (80%) and AdvS (60%). Among these features, the number of advertising TDMA scheduling (SchS) messages sent was not used for the other types of expert classifiers–in contrast to DistToCH and Time that are recurrent features for multiple attacks. This is an important feature to model TDMA attacks since it explores vulnerabilities of this particular protocol. Figure 7 shows these results.

The J48 classifier reached the highest F1-Score (91.94%) for this kind of classifier using the feature subset composed of Id, Time, SchS, DataR, and DistCHtoBS. It correctly detected 1,162.8 instances with only 38.8 false positives on average for a 5-fold cross-validation. As a result, J48 presented a precision of 96.78% and a recall of 87.58%. When considering a 95% confidence interval, REP Tree and Random Tree obtained statistically equivalent F1-Scores.

6.4 Results for WSN-DS with the binary classifiers

After analyzing the individual attacks, we experimented with combining grayhole, blackhole, flooding, and TDMA attack instances into a single attack class. In this case, the most selected features were the DataStoBS (80%), Id (60%), AdvS (60%), DistCHtoBS (60%), and Energy (60%). Note that in contrast to the features selected for the expert classifiers, in the combined dataset, the amount of energy consumed in the previous round is among the most selected features. A potential explanation is that nodes under attack tend to consume more energy. Although this is not a specific feature related to a particular attack pattern, it can be used as an indicator to detect a behavior shift that signals an attack. The results from the binary expert scenario are shown in Fig. 8.

Once again, the J48 classifier reached the best performance results. The feature subset composed of DataStoBS, DistCHtoBS, Id, Energy, and AdvS enabled J48 to reach the highest F1-Score (96.93%), with a precision of 97.64% and a recall of 96.22%. On average for the 5-fold cross-validation, 6,657.6 attack instances were correctly detected by J48, with 160.8 false positives. Considering the 95% confidence intervals, Random Tree has an equivalent F1-Score to J48.

Fig. 5 Results from the blackhole expert scenario (5-fold average)

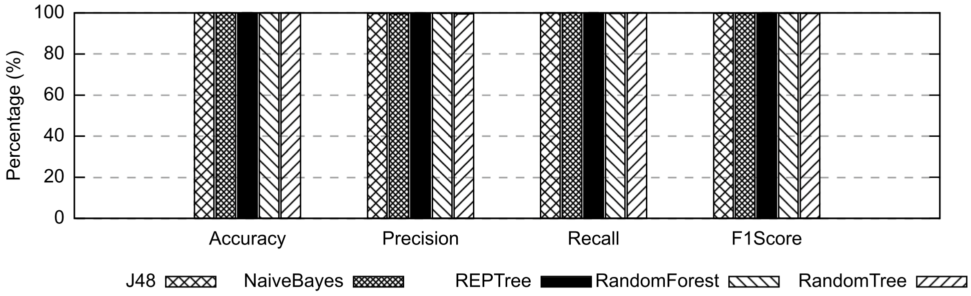


Fig. 6 Results from the flooding expert scenario (5-fold average)

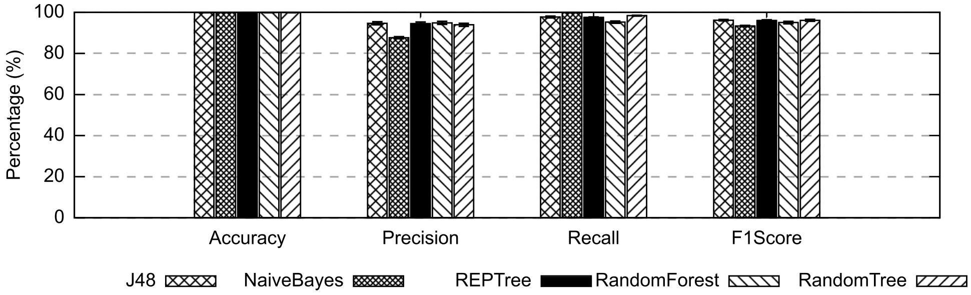
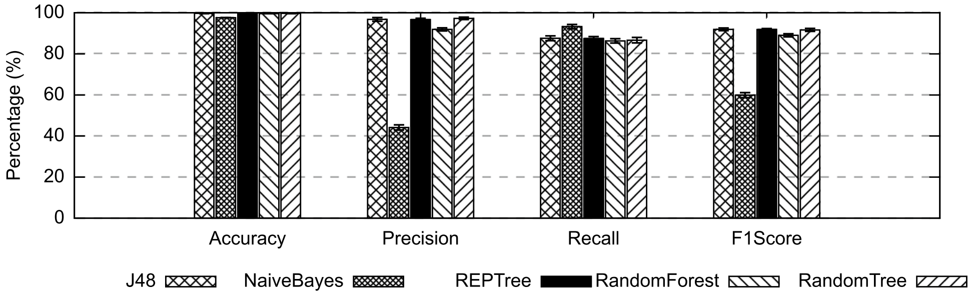


Fig. 7 Results from the TDMA expert scenario (5-fold average)



To better understand the consequences of performing a binary classification, Table 7 shows a confusion matrix expanding the `attack` class into the four specific types of attack. The underlined items are those classified in their correct class. It can be observed that the blackhole attacks did not generate false positives for this binary classifier. Moreover, on average, 74,510 instances were classified correctly, these being 67,852.4 `normal` instances and 6,657.6 `attack` instances. In comparison, the expert classifiers classified 74,596.6 instances correctly, these being 67,971.2 `normal` and 6625.4 `attack`. With a 95% confidence interval, the error margin is less than 36.5 instances.

Table 7 Confusion matrix for J48 classifier in the binary classification scenario (5-fold average)

		Classification results	
		Normal	Attack
Ground truth	Normal	67,852.4 (99.76%)	160.8 (0.24%)
	TDMA	141.6 (10.67%)	1,186.0 (89.33%)
	Blackhole	0 (0%)	2,009.8 (100%)
	Grayhole	110.0 (3.77%)	2,809.2 (96.23%)
	Flooding	9.8 (1.48%)	652.6 (98.52%)

6.5 Results for WSN-DS with multi-class classifiers

For the multi-class classifiers, the most selected features were the `AdvS` (100%), `DataStoBS` (80%), `Time` (60%), `WhoCH` (60%), and `AdvR` (60%) to represent all attacks combined into the same experiment. From those results, we can observe that the most recurrent features for the individual attacks are also selected to the multi-class classification (e.g., `AdvS` and `Time`).

According to Fig. 9, the highest F1-Score (96.89%) was obtained by the J48 classifier; J48 uses `DataStoBS`, `DistCHtoBS`, `WhoCH`, `Energy`, and `AdvS` to reach a precision of 97.63% and a recall of 96.16%. Considering the 95% confidence interval, Random Forest had an equivalent F1-Score to J48.

The detailed results of the multi-class J48 classifier are shown in Table 8. The underlined items are those classified in their correct class. J48 detected 6,653.2 attacks correctly—713 of them were classified as a different attack class than the expected, but still as attack—and presented 161.4 false positives.

In general, the expert classifiers are statistically a better option than binary and multi-class to compose classification-based IDSs. Considering only the attack coverage, the J48 classifier reached more true positives using the binary classification (i.e., 6,656 TPs) than the multi-class approach (i.e., 5,939 TPs). Therefore, simply considering the TP as a metric, the binary approach can give better results. Otherwise, if the IDS response action depends on the identification of the exact attack class, the multi-class setup should be adopted.

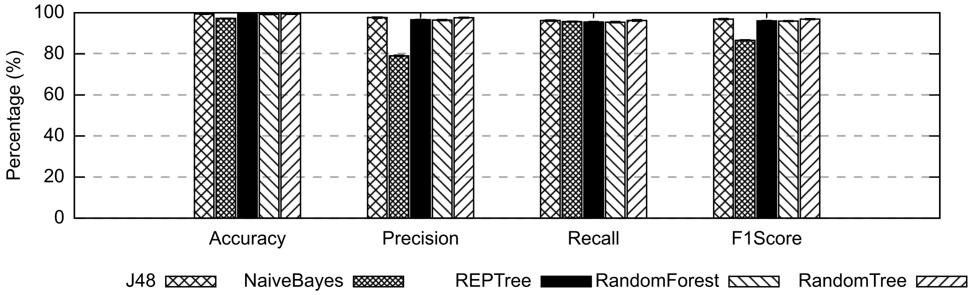
6.6 Discussion

GRASP-FS selected features that improved the classifiers' F1-Score for sensors and actuators from both WSN-DS [3] and SWaT [10] datasets. However, the SWaT dataset is composed of only one attack class. Thus, it was not possible to compare binary and multi-class classifiers to the expert classifier specialized for injection attacks. On the other hand, WSN-DS has four attack classes which enabled us to compare these approaches.

In most cases, all approaches have very similar performance for WSN-DS scenarios, often within the uncertainty of the confidence intervals for all metrics, as shown in the previous sections. That suggests that all three training approaches can achieve good results combined with a proper feature subset is selected, like our GRASP-FS. Nevertheless, the best option can vary according to the training approach and the considered metrics.

In Table 9, we present a compilation of results from expert, binary, and multi-class (both as binary and as specific, see Section 5) approaches considering accuracy, F1-Score, recall, and precision. The first column of the table specifies dataset used to test the IDSs for each experiment. In particular, we assess all training approaches by considering each specific attack and normal instances separately (i.e., grayhole and normal, blackhole and normal, flooding

Fig. 8 Results from the binary expert scenario (5-fold average)



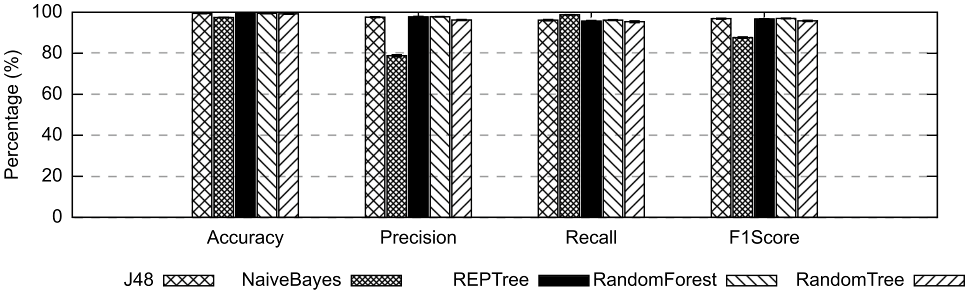


Fig. 9 Results from the multi-class scenario (5-fold average)

Table 8 Confusion matrix for J48 classifier in the multi-class scenario (5-fold average)

		Classification results				
		Normal	TDMA	Blackhole	Grayhole	Flooding
Ground truth	Normal	67,851.8	29.4	1	79.2	51.8
	TDMA	146.6	988.8	123	69.2	0
	Blackhole	0	415	1,580	14.8	0
	Grayhole	108.2	35.2	54.6	2,721.2	0
	Flooding	11	0.6	0.4	0.2	650.2

and normal, and TDMA and normal). Thus, for each expert classifier, TPs are only those attacks correctly classified in that specific class (e.g., a TP is when Expert-G classifies an instance as a grayhole attack). Moreover, we also carry out experiments considering all attacks simultaneously in the testing dataset (i.e., all attacks and normal). For this latter analysis, we consider as TPs every attack class classified as an attack, like we do for the multi-class binary analysis.

Considering hypothetical scenarios in which an attacker executes each specific attack class separately, the expert IDSs have the highest accuracy and F1-Score for all cases — see the bold items in the 3rd and 4th columns of Table 9. In particular, blackhole and flooding are detected with more than 99.9% accuracy. Considering the F1-Score metric, Grayhole attacks are similarly detected with more than 93%. These results show that GRASP-FS can provide proper features to enable intrusion detection with overall high performance. However, detection of TDMA attacks requires further improvement.

The multi-class IDSs performed generally better than the binary IDS, except for TDMA attacks. However, when computed as the `Multi-class Specific` method, recall is affected by the high number of false negatives (it is easier to detect an attack than to correctly classify it). Note also that `Multi-class as Binary` has lower precision due to the number of false positive generated when normal instances are classified into any attack class.

The more realistic scenario, when multiple types of attacks occur simultaneously, is presented in the last lines of Table 9 (test dataset includes all attacks and normal instances). In particular, we see that standalone expert IDSs may not be the most appropriate: the average of results of each expert IDS, when classifying all instances (i.e., including instances that the experts were not trained for), presents very low recall (42.95%) and F1-Score (52.57%). To illustrate the magnitude of IDS effectiveness, the number of false positives is lower (i.e., 791.18, on average) and the number of false negatives is very high (i.e., 3,947.31, on average).

The results reveal that expert and multi-class IDSs are a good alternative to replace the binary classification, especially in terms of precision.

When we choose 5 as the number of features for the GRASP-FS, the results for all approaches and metrics are typically very high (some above 99%) using both the expert and the multi-class approaches, even though the specific features varied for each classifier. This, again, shows the importance of a proper feature selection process, as the one presented in this paper. In the future, we will carry out a sensitivity analysis on the number of features required for these IDSs.

We emphasize that our scope is the proposal of the GRASP-FS to select features for IDSs deployed into the CPS scenarios, in particular, analyzing data collected from the perception layer. Thus, the deployment aspects are out of our scope. Nevertheless, as feature selection can be an off-line process, note that GRASP-FS can run on devices without energy or processing power constraints, such as application layer or cloud servers. If the CPS security manager chooses to deploy it on perception-layer devices, we recommend using tree-based classifiers, such as J48 due to the intrinsic functions that may demand fewer resources without impacting the performance results.

Table 9 A detailed comparison between expert, binary, and multi-class classifiers for WSN dataset, considering accuracy, F1-Score, recall, and precision

Testing dataset	Approach	Accuracy	F1-Score	Recall	Precision
Grayhole and normal	**Expert-G**	**99.32%**	**93.25%**	**97.84%**	**90.51%**
	Binary	99.13%	90.13%	96.40%	84.63%
	Multi-class spec.	98.99%	87.41%	84.60%	90.41%
	Multi-class as bin.	99.15%	90.38%	96.93%	84.66%
Blackhole and normal	**Expert-B**	**99.99%**	**99.83%**	99.95%	99.72%
	Binary	99.25%	88.38%	99.31%	79.61%
	Multi-class spec.	99.59%	92.42%	86.04%	**99.83%**
	Multi-class as bin.	99.27%	88.68%	**99.97%**	79.67%
Flooding and normal	**Expert-F**	**99.91%**	**95.34%**	**97.75%**	**93.14%**
	Binary	99.20%	69.57%	94.49%	55.05%
	Multi-class spec.	99.64%	84.07%	76.80%	92.87%
	Multi-class as bin.	99.23%	71.03%	97.69%	55.80%
TDMA and normal	**Expert-T**	**99.26%**	**84.88%**	88.26%	**85.36%**
	Binary	99.07%	78.72%	**89.90%**	70.01%
	Multi-class spec.	98.91%	67.40%	58.42%	79.64%
	Multi-class as bin.	99.06%	78.40%	89.36%	69.83%
All attacks and normal	**Experts (average)**	94.50%	52.57%	42.95%	**97.37%**
	Binary	98.93%	94.48%	95.82%	93.47%
	Multi-class spec.	97.58%	86.12%	81.23%	91.64%
	Multi-class as bin.	**98.99%**	**94.62%**	**96.44%**	92.87%

Bold items are the best results for each training dataset

7 Conclusion

The increased connectivity between the physical and cyber domains is at the core of the unique security challenges in c*yber-physical systems* (CPS). As an integral part of so many critical infrastructure systems, it is very important to consider security aspects of CPS. In this context, intrusion detection systems (IDSs) should analyze the proper features to enable the best performance be it measured by precision, recall, accuracy, or F1-Score.

We demonstrate in this work, through two datasets — WSN-DS (i.e., with four attacks) and SWaT (i.e.,with one attack) — and five classifier algorithms, that choosing the appropriate features is very important and should consider several factors: the classifier algorithms employed, the attack class analyzed, and the IDS training approach. In particular, expert classifiers can use the best features to represent a specific attack pattern, whereas binary and multi-class approaches should select features that can generalize multiple attacks.

From the 51 available features on the SWaT dataset, GRASP-FS selected a reduced subset of 5 features and enabled Random Tree to reach 96.97% F1-Score and 99.65% accuracy. It shows that using GRASP-FS for the perception layer feature selection is feasible and provides good results.

The highest average results for WSN-FS dataset were given by the expert IDSs: blackhole attacks were detected with up to 99.83% F1-Score and 99.99% accuracy, whereas the average for all attacks is 93.32% F1-Score and 99.62% accuracy. The detailed analysis of individual attack scenarios reveals multi-class as a prominent approach, when using the appropriate features. This approach is especially recommended to avoid false negatives for scenarios in which it is not feasible to ensure that expert IDSs can process only their respective known attack patterns. Finally, we showed that all IDSs had their poorest performance in trying to detect TDMA attacks.

As future work, we plan to extend our analysis to cover other CPS layers and particular CPS domains. In addition, we plan to explore the impact of periodic updates on the features being analyzed by real-time IDSs. Finally, we plan to study a method to combine multiple expert IDSs into a single IDS with improved performance. Using clustering algorithms to group different attack patterns may be a potential future direction.

Acknowledgements This work is supported in part by CAPES, CNPq, FAPERJ, and CGI/FAPESP.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Alcaraz C, Lopez J (2014) Diagnosis mechanism for accurate monitoring in critical infrastructure protection. Comput Stand Interfaces 36(3):501–512
2. Alcaraz C, Lopez J (2014) WASAM: a dynamic wide-area situational awareness model for critical domains in smart grids. Future Gener Comput Syst 30:146–154
3. Almomani I, Al-Kasasbeh B, Al-Akhras M (2016) WSN-DS: a dataset for intrusion detection systems in wireless sensor networks. Journal of Sensors 2016
4. Ashibani Y, Mahmoud QH (2017) Cyber physical systems security: analysis, challenges and solutions. Comput Secur 68:81–97
5. Bermejo P, Gámez JA, Puerta JM (2011) A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. Pattern Recogn Lett 32(5):701–711
6. Dahi ZA, Alba E (2020) The grid-to-neighbourhood relationship in cellular GAs: from design to solving complex problems. Soft Comput 24(5):3569–3589
7. Díez-Pastor JF, García-Osorio C, Rodríguez JJ (2014) Tree ensemble construction using a grasp-based heuristic and annealed randomness. Inf Fusion 20:189–202
8. Diez-Pastor JF, García-Osorio C, Rodríguez JJ, Bustillo A (2011) GRASP forest: a new ensemble method for trees. In: International workshop on multiple classifier systems, pp 66–75. Springer
9. Esseghir MA (2010) Effective wrapper-filter hybridization through GRASP schemata. In: Feature selection in data mining, pp 45–54
10. Goh J, Adepu S, Junejo KN, Mathur A (2016) A dataset to support research in the design of secure water treatment systems. In: International conference on critical information infrastructures security, pp 88–99. Springer
11. Harris E (2002) Information gain versus gain ratio: a study of split method biases. In: ISAIM
12. Jia D, Lu K, Wang J, Zhang X, Shen X (2016) A survey on platoon-based vehicular cyber-physical systems. IEEE Commun Surv Tutor 18(1):263–284
13. Kanakarajan NK, Muniasamy K (2016) Improving the accuracy of intrusion detection using gar-forest with feature selection. In: Proceedings of the 4th international conference on frontiers in intelligent computing: theory and applications (FICTA), pp 539–547. Springer
14. Kholidy HA (2021) Autonomous mitigation of cyber risks in the Cyber-Physical Systems. Futur Gener Comput Syst 115:171–187
15. Leitao P, Colombo AW, Karnouskos S (2016) Industrial automation based on cyber-physical systems technologies: prototype implementations and challenges. Comput Ind 81:11–25
16. Moshki M, Kabiri P, Mohebalhojeh A (2015) Scalable feature selection in high-dimensional data based on GRASP. Appl Artif Intell 29(3):283–296
17. Nourian A, Madnick S (2015) A systems theoretic approach to the security threats in cyber physical systems applied to stuxnet. Trans Dependable Secure Comput 15(1):2–13
18. Quincozes SE, Kazienko JF (2020) Machine learning methods assessment for denial of service detection in wireless sensor networks. In: 2020 IEEE 6th world forum on internet of things (WF-IoT), pp 1–6. IEEE
19. Quincozes SE, Passos D, Albuquerque C, Ochi LS, Mossé D (2020) GRASP-based feature selection for intrusion detection in CPS perception layer. In: 2020 4th Conference on cloud and internet of things (CIoT), pp 41–48. IEEE
20. Ribeiro CC, Resende MG (1999) Algorithm 797: Fortran subroutines for approximate solution of graph planarization problems using GRASP. ACM Trans Math Softw (TOMS) 25(3):341–352
21. Week S (2020) IoT devices infected via supply chain attack. https://www.securityweek.com/
22. Yu X, Xue Y (2016) Smart grids: a cyber-physical systems perspective. Proc IEEE 104(5):1058–1070
23. Yusta SC (2009) Different metaheuristic strategies to solve the feature selection problem. Pattern Recognition Letters 30(5)
24. Zhang Y, Qiu M, Tsai C-W, Hassan MM, Alamri A (2017) Health-CPS: healthcare cyber-physical system assisted by cloud and big data. IEEE Syst J 11(1):88–95