Feature Selection: supporting the mining process on cyber-physical systems result datasets

Hebert Silva, Tania Basso, Regina Moraes

[1]University of Campinas - UNICAMP - Limeira, Brazil

hebert.oliveiras@gmail.com, {tbasso@cotil, regina@ft}.unicamp.br

Abstract. *Cyber physical systems (CPs) often generated large sets of data during monitoring or testing processes. Analyzing these results manually is not practical as it requires great human effort. Machine learning can be a valuable approach to support the analysis and can help the responsible professional to make urgent decisions. Moreover, most of time these datasets contain missing, extreme, duplicate or defective values that can bias the general classification methods, which can be worked around with feature selection techniques. However, identifying all the possible combinations of features and select the best set of them is not an easy task. In this work, we present a feature selection study to automate the analysis of CPs test result datasets by the support of machine learning. The idea is to automatically identify a set of attributes that optimize the accuracy of the chosen machine learning model. Three scenarios that use large amounts of data from cyber physical systems were used and the results of feature selection were surprising in some cases.*

Resumo. *Os sistemas físicos cibernéticos (CPs) geralmente geram grandes conjuntos de dados durante os processos de monitoramento ou teste. Analisar esses resultados manualmente não é viável, pois requer um grande esforço humano. O aprendizado de máquina pode ser uma abordagem valiosa para apoiar essa análise e pode ajudar o profissional responsável a tomar decisões urgentes. Além disso, na maioria das vezes, esses conjuntos de dados contêm valores ausentes, extremos, duplicados ou defeituosos que podem influenciar os métodos de classificação geral, podendo se contornar esse problema com técnicas de seleção de recursos. No entanto, identificar todas as combinações possíveis de recursos e selecionar o melhor conjunto deles não é uma tarefa fácil. Neste trabalho, apresentamos um estudo de seleção de recursos para automatizar a análise de resultados de testes de CPs com o suporte de aprendizado de máquina. A ideia é identificar automaticamente um conjunto de atributos que otimizam a precisão do modelo escolhido. Três cenários que usam grandes quantidades de dados de sistemas físicos cibernéticos foram usados e os resultados da seleção de recursos foram surpreendentes em alguns casos.*

1. Introduction

Cyber Physical systems (CPS) constitute automated systems for monitoring physical events and they are often considered a mission critical system, increasing the responsibility on decision making. Normally, these kind of systems are submitted to rigorous test process and need to be monitored along the time that are being used. Testing

and monitoring processes often generate large volumes of data (the well known Big Data), which often need to be analyzed to support important and urgent decisions. However, the analysis performed manually by humans is unfeasible as it would expend a great effort to allow timely actions. Therefore, the support of the data mining process can be valuable for this task.

As part of data mining process, machine learning implements computing models to learn the discovered patterns. Machine learning is a technology in which computers have the ability to learn through associations of different data, being advantageous that there is a large volume (that is, large number of instances, taking advantage of Big Data) to generate important results. As intelligence is extracted from the data, simply collecting the data is not enough, it is necessary that the data collected have the expected quality and that the analysis algorithms (or models) can make accurate predictions based on known data [Sraavnthi et al. 2019]. The automation of this modeling process, the training of the modeling and testing lead to accurate predictions to support needed changes. According to Rothermich [Rothermich J. 2021], 43% of the works researched in his study claim that data quality is the biggest barrier to machine learning. In the same direction, 38% of the same works state that the second biggest barrier is the lack of data availability. From this perspective, we can conclude that the machine learning process is still very dependent on the data quality and availability.

Dependent on physical devices, often the datasets resulting from test and monitoring processes in CPs contain missing, extreme, duplicate or defective values, configuring low quality data, that can bias the general classification methods, which is normally important part of machine learning. So, data quality assessment is vitally important to allow the application of data mining process on CPs test result datasets. Therefore, verifying and validating the quality of data that feed and are produced by CPs is an indispensable task, but quite complex due to specificities, such as limited hardware resources commonly found in CPs, which add a greater degree of difficulty to the data quality monitoring task.

Feature selection is a widely explored topic in machine learning and can help to choose the highest quality subset to be used, but only recently some approaches trying to automate this process have emerged. While it is advantageous to have large number of instances in the dataset to help the selection, this does not hold true for the number of features. For many reasons, feature selection is relevant to establishing good machine learning models and requires some level of dataset cardinality reduction, that is, the reduction of elements or attributes in the selection. It is not uncommon for dataset to have information that is unnecessary for creating and training models. For example, if a dataset has a large number of attributes (columns) to describe an object (instance) and that data is too sparse (unordered), keeping them in the training dataset may not add value to the model. In a worst-case scenario, it can impair or derail machine learning. Moreover, in case of duplicate attributes, there can be a negative impact on the model's performance, specifically on its execution time. This scenario is aggravated when big data is involved, precisely due to the number of elements, both attributes and number of instances. Feature selection related to data quality is not new, but even with the existing solutions for data analysis, it is still possible to observe a gap to be filled in relation to process automation and the ideal methodology for assessing the quality of data provided by CPs.

The goal of this work is to understand the impact of feature selection on the performance of supervised machine learning algorithms (e.g., Naive Bayes, Random Forest) during the classification task and to verify the effectiveness of the algorithm in classifying correctly even using a smaller number of features.

Through the experiments, it is possible to verify the accuracy of these algorithms regarding the feature selection in their pre-processing stage, with a view to increase their quality and the quality of data applied in machine learning when CPs data are involved.

For now, we checked the classic feature selection algorithms available in the Weka tool [1] (e.g., ClassifierAttributeEval, InfoGainAttributeEval). The results of the experiments show how feature selection can help more accurate machine learning process and to identify anomalous instances provided by CPs as well.

The remainder of this work is organized as follows: Section 2 presents related works to machine learning, feature selection and data quality. In Section 3 a design of a methodology for the feature selection is presented. Section 4 presents results acquired in the experiments to better understand the feature selection for data quality in CPs. Finally, Section 5 presents the conclusions of this work and some insights for future work.

2. Background and Related Work

Ensemble learning is based on combining multiple models instead of a single model to solve a particular problem. The idea of ensemble learning is not only applicable to classification, but it can be used to improve other machine learning disciplines such as feature selection. According to Bolan-Canedo and Alonso-Betanzos [Bolón-Canedo and A.Alonso-Betanzos 2019], there are three major feature selection approaches: filters (which rely on general characteristics of the data and are independent of the induction algorithm), wrappers (which use the prediction provided by a classifier to evaluate subsets of features) and embedded (which perform feature selection in the process of training and are specific to given learning machines). There are some works that address the use of these feature selection methods, which are described below.

The work by Pipino et al. [Pipino et al. 2019] describes the principles for developing metrics and how to measure data quality (DQ) in practice. DQ is divided into dimensions that indicate how to evaluate data from their different perspectives, such as relevance, temporality, security, among others and a generic model is used to assess data quality. The work by Tang et al. [Tang et al. 2014] defined a feature weighting as a generalization of feature selection, assigning a value, usually in the interval [0,1] or [-1,1] to each feature. The greater this value is, the more salient the feature will be. Both works [Pipino et al. 2019][Tang et al. 2014] are important to understand the options we have to deal with the large dimensionality of the dataset, as well as to guide the development of the methodology to obtain a DQ score.

Vidyavathi work [Vidyavathi 2019] proposes a two-phase feature selection approach using both filter and wrapper. Firstly, an artificial neural network weight analysis was used as a filter aiming to remove irrelevant features and then a genetic algorithm was used as a wrapper to remove redundant and useless features. The result is a significant reduction of the size of features without compromising the

[1] https://machinelearningmastery.com/perform-feature-selection-machine-learning-data-weka/

classification or the prediction performance. In the same direction, Husna and Adiwijaya [Husna and Adiwijaya 2018] used k-means algorithm as the clustering approach for feature selection, so that redundancy in microarray data is removed. The result of clustering is ranked using the Relief algorithm such that the best scoring element for each cluster is obtained. All best elements of each cluster are selected and used as features in the classification process. The accuracy of the proposed approach is therefore higher than the approach without clustering.

A different approach was used by Angelis et al. [Angelis et al. 2006]. They analyzed both filter and wrapper methods and modeled the problem as an optimization problem, defining an objective function and constraint that altogether express an integer programming problem. Then, they show how the feature selection problem can be formulated as a subgraph selection problem derived from the lightest k-subgraph problem. The results of some experiments show that the method can determine good subsets of features for data mining applications. Also, Mafarja and Mirjalili [Mafarja and Mirjalili 2018] proposed a wrapper feature selection approach based on Whale Optimization Algorithm (WOA). The experiments compared the approach results to three algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), the Ant Lion Optimizer (ALO), and five standard filter feature selection methods and conclude that the proposed approach perform better than other previous proposal. They also balance exploration and exploitation efficiently to first avoid a large number of local solutions in feature selection problems and it is able to find an accurate estimation of the best solution in searching the optimal feature subset.

Differently from these previous work, our study defines subsets of Cyber-Physical features based on ordered ranking provided by feature selecion algorithms from Weka. The goal is to evaluate the data quality of these subsets and their impact on classifiers as well as the capacity of the algorithm in choosing correctly the focused information (faults, attacks, among others) even using a smaller number of features.

3. Feature Selection: the proposed method

This section presents the method we used to define data subsets using feature selection in the context of CPs. Figure 1 presents the steps of this method.

A Cyber-Physical system (CPs) integrates sensor network with cyber resources. The CPs collects sensor data from physical world and links them to information sources for real-time analysis, often using middleware to support the applications. In Figure 1, this process is represented by the sequence *Cyber-physical systems under test, IoT/Sensor, Middleware*.

Since a typical CPs includes even thousands of sensors, which generate readings every few minutes and form a huge data stream, a *Big data Storage* is necessary. Relevant information can be extracted from this massive amount of data, however, the complexity of CPs (given from, for example, the lack of reliability in communications, the large-scale and the variability of the environments) makes data analysis a complex task. So, we used *Feature selection* and *Machine Learning - ML* processes to better perform the *Interpretation and Evaluation* of the results.

The focus of this work is the *Feature selection* process. It starts with a *Initial data set evaluation*, where the data analyst evaluates the data sets, its characteristics and

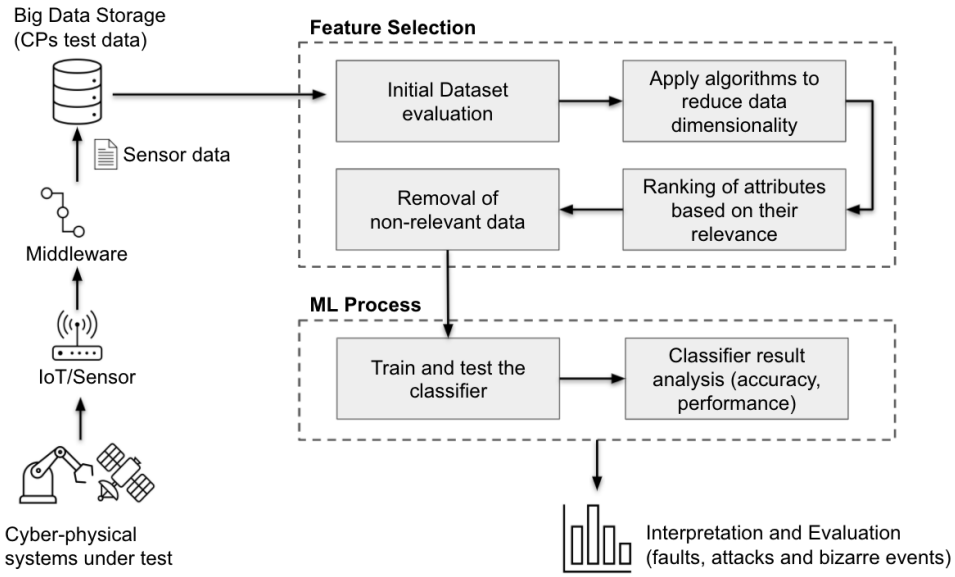

Figura 1. Feature selection method to improve data mining on CPS

necessary adaptations to be used in the feature selection algorithms. Then, the step *apply algorithms to reduce data dimensionality* is executed. The output of the algorithms are attribute rankings that indicates the attributes they consider more relevant for classification in the dataset under analysis. It is important to mention that, in this step, more than one algorithm should be used, because the more algorithms are used, the more accurate the ranking of attributes can be. In the next step, the *ranking of attributes based on their relevance*, the data analyst defines the selection ensemble. This is done through the aggregation of the better ranked attributes resulted by the algorithms, considering the frequency they were ranked (e.g., the attributes ranked as the most relevant by the majority of the algorithms is considered the most relevant to the ensemble). Finally, the data analyst must perform the *removal of non-relevant data*, i.e., based on the work of Kumar [Kumar 2021], the top 10 ranked attributes are selected and the subset to be used in the machine learning process is defined.

In the *ML Process*, the first step is to *train and test the classifier*. In this step, the data subset obtained from feature selection process is used in the classification algorithms. The results are analyzed in the *classifier result analysis* step, where the accuracy and performance of the classifier are evaluated. Then, the classifier with better results is selected and new *interpretation and evaluation* tasks must be performed. A classification model is generated and can be used to test new data sets such as attack or benign data, trying, for example, to identify faults, attacks and bizarre events.

4. Experiments and Results of Feature Selection

To better understand the impact of the features on the data quality of a dataset, we chose three different scenarios related to cyber-physical systems. The first scenario was borrowed from Hindy et al. [Hindy et al. 2018]. That work aim at improving Security Information and Event Management (SIEM) for critical infrastructure using machine learning to identify patterns in the data reported by PLCs (Programmable Logic Controllers) in a water system controlled by SCADA. The system is composed of two

tanks. Each tank can contain either fuel or water and can be set to two distinct modes — acting either as a distributor or as storage. The dataset[2] comprises actuator and sensor readings that the PLC recorded periodically at 0.1 second intervals. Each instance is composed only by three attributes (Date and Time, Register Number, and Register Value of the PLC), i.e. it is a dataset with small dimensionality. Being available through the internet, the system is a target of attacks, and the authors were able to identify 14 different classes of attacks. The first analysis performed on this dataset presented an accuracy that did not corroborate with the one published by Hindy et al. [Hindy et al. 2018] work. The classification was performed using the following classifiers: i) Support Vector Machines using Sequential Minimal Optimization – SMO; ii) Naive Bayes-NB; iii) Locally Weighted Learning - LWL; iv) Decision Table – DT; v) Random Forest - RF.

At the beginning of the tests, we observed that the timestamp attribute, containing the date and time when the attacks or benign traffic was collected, presents a recurrent value in the training dataset (once the attack is taking place and the data collection is done in 0.1s there is a large number of instances with the same timestamp), causing a possible overfitting. Thus, before performing the tests in the first scenario, we performed a selection of attributes using the resources available in the Weka tool (release 3.9, with Heap Memory size increased to 4096mb, to be able to process the high amount of data), in order to understand if the attributes selection could corroborate the initial conviction that the timestamp attribute could be removed without harming the classification models.

Register not separated - Similar Hindy et al. (2018)								
Algorithm	Classifier AttributeEval	Correlation AttributeEval	GainRatio AttributeEval	InfoGain AttributeEval	OneR AttributeEval	ReliefF AttributeEval	SymmetricalUncert AttributeEval	CfsSubsetEval
Ranked	attribute	attribute	attribute	attribute	attribute	attribute	attribute	attribute
1	TimeStamp	Value	Register	TimeStamp	Value	Register	TimeStamp	TimeStamp
2	Value	Register	TimeStamp	Register	Register	Value	Register	Register
3	Register	TimeStamp	Value	Value	TimeStamp	TimeStamp	Value	Value

Figura 2. Experiment Results - First Scenario

As we can observe in Figure 2, there is no consensus among the classification algorithms regarding the timestamp attribute. For the ClassifierAttributeEval, InfoGainAttributeEval, SymmetricalUncertAttributeEval and CfsSubsetEval algorithms, the timestamp attribute is the most important for sorting. The CorrelationAttributeEval, OneRAttributeEval and ReliefFAttributeEval algorithms the TimeStamp attribute is the least important, and for the GainRatioAttributeEval algorithm it is the second most important. A deep analysis shows that algorithms that work with data correlation realize that the date and time do not have a strong correlation with the rest of attributes in the dataset, whereas the algorithms that privilege the gain of information and the evaluation of the individual classification, judged the timestamp attribute as the most important.

We performed the classification by alternating on each classifier the use of timestamp attribute (including or excluding it). Accuracy results, error analysis and run time are shown in Figure 3 parts A, B and C respectively. The best accuracy rate is 71% for the SMO classifier without removing the timestamp; the corresponding absolute mean error is 0.11% and the run time is 645 seconds. When we removed the timestamp attribute, a reduction in the accuracy is observed in all classifiers, reaching close to 9% in the worst

[2]Dataset available in https://www.sciencedirect.com/science/article/pii/S2352340917303402

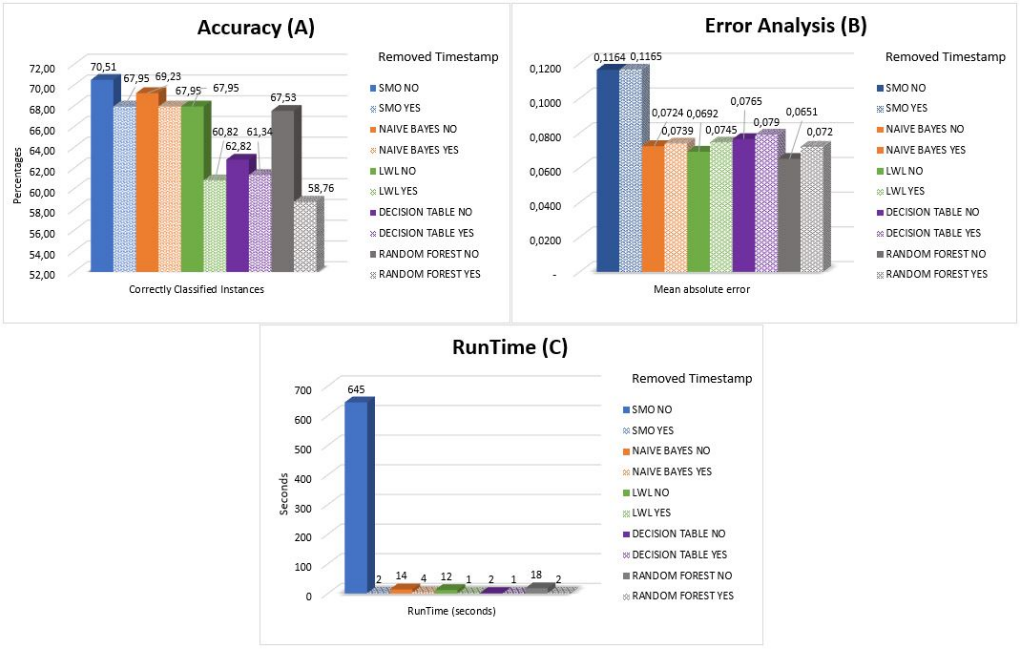

Figura 3. Experiment Results - First Scenario

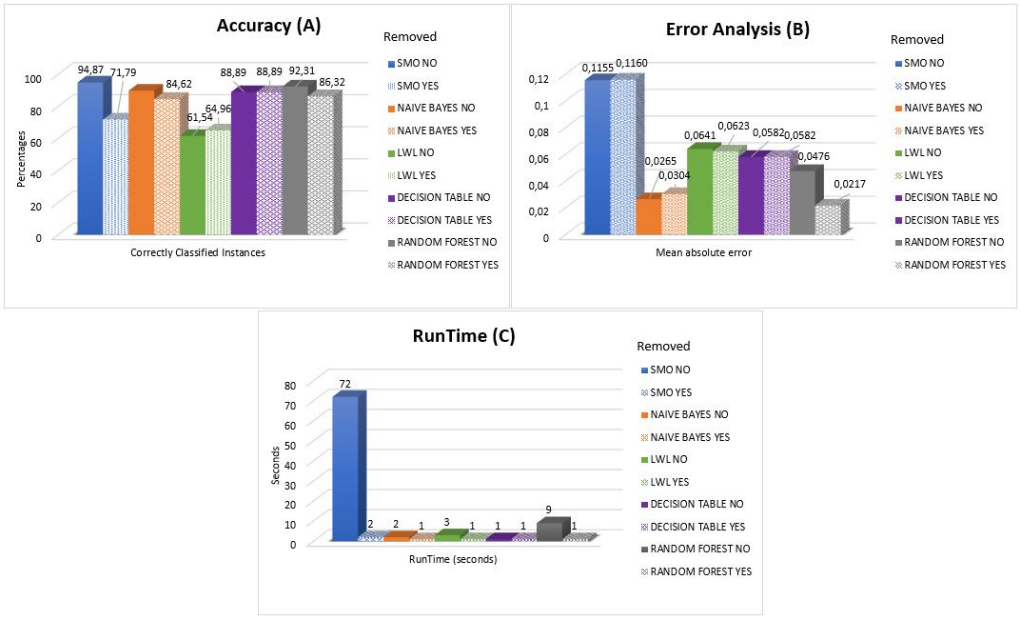

Figura 4. Experiment Results - First Scenario Without Timestamp

case (Random Forest algorithm). Surprisingly, despite the worsening of the accuracy, it is possible to verify that the mean absolute error suffered little change, indicating that despite the accuracy having worsened, the difference in the classification error is negligible. Regarding run time, there was a considerable reduction for all classifiers, which means an improvement in their performance. For the SMO it was reduced from 645 to 2 seconds.

Once verified the relevance of the timestamp attribute and to better understand

the correlation of the data at the same instant, we reconstruct the dataset, grouping the values collected by all sensors in the same timestamp and class of data (i.e., normal or each 14 types of attacks). In the new dataset, the dimensionality was increased (from 3 to 12 attributes) and the number of instances of the dataset decreased. Even increasing the dimensionality we follow the recomendation of Kumar [Kumar 2021] that points that around 10-features dataset is the most appropriate to achieve a good accuracy. When executing the attribute selection again (see Figure 5), it is verified that the timestamp attribute is not the main attribute for any of the classifiers.

We run again the classification, using now the dataset with 12 attributes, including the timestamp and with 11 attributes (without the timeStamp). We use the same classifiers as in the first experiment. It was possible to observe an accuracy of 94.87% for the data set with the timestamp and an absolute mean error of 0.11% for the best case (SMO algorithm). When we removed the timestamp attribute, the accuracy was again reduced for all models, except for the LWL classifier, which had an increase from 61.54% to 64.96% in the accuracy rate. The mean absolute error suffered small fluctuations, with a sharper drop only for the Random Forest algorithm, being reduced from 0.0476% to 0.0217%.

Register separated (New Approach)								
Algorithm	Classifier AttributeEval	Correlation AttributeEval	GainRatio AttributeEval	InfoGain AttributeEval	OneR AttributeEval	ReliefF AttributeEval	SymmetricalUncert AttributeEval	CfsSubsetEval
Ranked	attribute	attribute	attribute	attribute	attribute	attribute	attribute	attribute
1	Register 10	Register 5	Register 6	Register 10	Register 6	Register 5	Register 6	Register 6
2	Register 2	Register 4	Register 7	Register 2	Register 5	Register 9	Register 5	Register 5
3	Register 1	Register 9	Register 8	Register 1	Register 7	Register 7	Register 7	TimeStamp
4	Register 4	Register 7	Register 9	Register 4	Register 8	Register 8	Register 8	Register 7
5	Register 3	Register 6	Register 5	Register 3	Register 4	Register 4	Register 4	Register 2
6	Register 5	Register 8	Register 10	Register 5	Register 2	Register 6	Register 2	Register 8
7	Register 9	Register 2	Register 1	Register 9	Register 9	TimeStamp	TimeStamp	Register 4
8	Register 8	Register 3	Register 2	Register 8	Register 3	Register 3	Register 9	Register 3
9	Register 7	Register 10	Register 4	Register 7	Register 10	Register 2	Register 3	Register 9
10	Register 6	Register 1	Register 3	Register 6	Register 1	Register 1	Register 10	Register 10
11	TimeStamp	TimeStamp	TimeStamp	TimeStamp	TimeStamp	Register 10	Register 1	Register 1

Figura 5. Values Obtained - First Scenario Experiments

Selection Features For Second Scenario								
Algorithm	Classifier AttributeEval	Correlation AttributeEval	GainRatio AttributeEval	InfoGain AttributeEval	OneR AttributeEval	ReliefF AttributeEval	SymmetricalUncert AttributeEval	CfsSubsetEval
Ranked	Attributes	Attributes	Attributes	Attributes	Attributes	Attributes	Attributes	Attributes
1	HpHp_L0,01_pcc	H_L0,1_mean	HpHp_L5_mean	HH_jit_L5_mean	HH_jit_L5_mean	HH_L5_magnitude	HH_jit_L5_mean	HH_L5_magnitude
2	HH_L5_magnitude	MI_dir_L0,1_mean	HpHp_L3_mean	HH_jit_L3_mean	HH_jit_L3_mean	HH_L1_mean	HH_jit_L3_mean	HH_L0,1_radius
3	HH_L5_pcc	MI_dir_L1_mean	HpHp_L1_mean	HH_jit_L1_mean	HH_jit_L1_mean	H_L1_mean	HpHp_L5_mean	HH_jit_L5_mean
4	HH_L3_weight	H_L1_mean	HH_L0,1_radius	HH_jit_L0,01_mean	HH_jit_L0,01_mean	MI_dir_L1_mean	HpHp_L3_mean	HH_jit_L3_mean
5	HH_L3_mean	H_L3_mean	HpHp_L0,01_mean	HH_jit_L0,1_mean	HH_jit_L0,1_mean	HH_L1_magnitude	HpHp_L1_mean	HH_jit_L1_mean
6	HH_L5_radius	MI_dir_L3_mean	HpHp_L0,1_mean	MI_dir_L0,01_mean	HH_jit_L0,01_variance	HH_L5_mean	HpHp_L0,01_mean	HH_jit_L0,1_mean
7	HH_L5_std	H_L0,01_mean	HpHp_L1_magnitude	H_L0,01_mean	H_L0,01_mean	HH_L0,1_magnitude	HH_jit_L1_mean	HH_jit_L0,01_variance
8	HH_L3_magnitude	MI_dir_L0,01_mean	HpHp_L5_magnitude	HpHp_L5_mean	MI_dir_L0,01_mean	HH_L3_magnitude	HpHp_L0,1_mean	HpHp_L3_mean
9	HH_L5_mean	H_L5_mean	HH_jit_L0,01_variance	HpHp_L3_mean	HH_L0,1_std	HH_L0,1_mean	HpHp_L5_magnitude	HpHp_L1_mean
10	H_L0,01_mean	MI_dir_L5_mean	HpHp_L0,1_magnitude	HpHp_L1_mean	HH_L0,01_std	HH_L3_mean	HpHp_L1_magnitude	--

Figura 6. Experiment Results - Features Selection For Second Scenario

In the second scenario the dataset[3] was extracted from UCI Machine Learning Repository [4], it is a multivariate large dataset (more than 7,000,000 instances) and large dimensionality as well (115 attributes per instance). The dataset refers to the IoT (Internet

[3]Dataset available in https://archive.ics.uci.edu/ml/machine-learning-databases/00442/Ecobee_Thermostat/
[4]https://archive.ics.uci.edu/

of Things) context including malicious data that can be divided into 10 attacks classes carried by 2 botnets and can also be used for multi-class classification (10 classes of attacks, plus 1 class of benign data).

The feature selection algorithms used in the first scenario were the ones used in this scenario as well. Similar to the work of [Bolón-Canedo and A.Alonso-Betanzos 2019], we selected the best features to train the model based on the frequency and ranking that the attributes received from the set of algorithms (8 algorithms in total). To decide the top-10 most important attributes a score was set for the top-10 attributes of each algorithm, when the first receives 10 points and follows in a decreased punctuation till the tenth that receives 1 point (other attributes has no punctuation). It means that if an attribute appears as the most important in all algorithms its punctuation will be set to 80 (8 times 10 points) and so on so forth.

We decided to work with the top-10 most important attributes based on the literature [Kumar 2021]. In case the last of the ten attributes presents a tie with the attribute that immediately follows it, the tiebreak will be given to the one with the highest total of times that it was ranked as the top-10 in the set of algorithms (considering the 8 algorithms being used). Figure 6 presents the top-10 attributes select by each algorithm. In Figure 7, the selected 10 attributes better classified among all algorithm are presented. They are obtained by selecting the ten highest weight, which is calculated by the punctuation of each one of them (see explanation above) divided by the maximum punctuation, i.e., divided by 80.

Selected Features For Second Scenario	
Attribute	**Weight**
HH_jit_L5_mean	0,4750
HH_jit_L3_mean	0,4250
HH_L5_magnitude	0,3625
HH_jit_L1_mean	0,3250
HpHp_L5_mean	0,2625
HpHp_L3_mean	0,2625
HpHp_L1_mean	0,2125
HH_jit_L0,1_mean	0,2125
HH_L0,1_radius	0,2000
H_L1_mean	0,1875

Figura 7. Experiment Results - Ensemble Features Selection For Second Scenario

For the classification of the second scenario we used the Random Forest algorithm, as it was the algorithm that kept a better performance after the selection of attributes in the first scenario, presenting a small reduction in the accuracy, but also a reducing the mean absolute error. In Figure 8, we can see that after the selection of attributes (see the chosen set in Figure 7), the accuracy had a negligible reduction (from 99.98% to 99.95%) when compared to the accuracy obtained using all the attributes of the dataset. The increase in the mean absolute error is not significant, rising from 0.0003 to 0.0004. One can observe the relevant contribution of attribute selection, when we analyzed the classification execution time. When the complete dataset (with all attributes) was used, the time taken to resume the classification was around ten minutes (563 seconds) against practically 3 minutes (169 seconds) when the reduced dataset was the target.

The third and last scenario is a public historical data from the satellite context

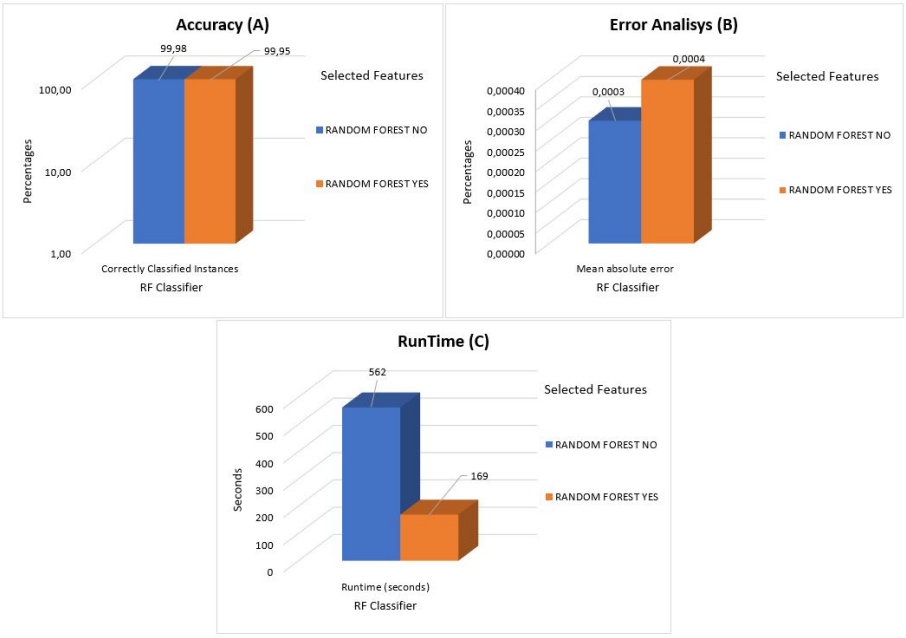

Figura 8. Experiment Results - Accuracy Second Scenario

obtained in the INPE (*Instituto Nacional de Pesquisas Espaciais*) website [5], which maintains monitoring systems such as meteorological information and large hydroelectric plants. The integrated environmental data system (called SINDA) stores all these data, which are collect through PCDs (Data Collection Platforms) spread all around the country and then transmitted to the database through the satellite support. Besides the real failure data (some instances lack partial data due to sensors malfunctioning or failures in data transmission) we deliberated injected faults (outlier values) in a copy of the dataset [6]. At first, we analyzed the original data using the classifiers, aiming to understand the dataset and the limits of each attribute. Next, we injected outlier values into some attributes of 9% of the instances (the ones that lack partial data) to emulate malfunctioning sensors and apply again the classifiers to test their capacity to deal with multi-columns failures.

Similar to the experiments performed, particularly in the second scenario, in this third scenario the classifiers were used to evaluate the original dataset and the dataset with the results obtained using the emulated attack scenario. Figure 9 presents the top-10 attributes selected by each one of the classifiers and the same is presented in Figure 10 regards the emulated attack scenario. The same calculus used in the second scenario were done in this scenario to obtain the weights of each one of the top-10 attributes and the ones that hold the highest values is selected and presented in the first two tables in Figure 11, being the (A) for the original dataset and the (B) for the emulated attack dataset.

Observing both tables (A) and (B) we realised that 5 attributes are common among them. They are presented in the last table (C), in Figure 11. The weight of attributes in the (C) table was obtained by calculating the average score of the first two tables. We performed the classification task using the Random Forest algorithm considering the original dataset with no feature selection, and the three features selection subsets (A), (B)

[5]http://sinda.crn.inpe.br/PCD/SITE/novo/site/index.php

[6]Dataset available in http://sinda.crn.inpe.br/PCD/SITE/novo/site/tabela.php?id=31974

Selection Features For Third Scenario Original Data								
Algorithm	Classifier AttributeEval	Correlation AttributeEval	GainRatio AttributeEval	InfoGain AttributeEval	OneR AttributeEval	ReliefF AttributeEval	SymmetricalUncert AttributeEval	CfsSubsetEval
Ranked	Attributes	Attributes	Attributes	Attributes	Attributes	Attributes	Attributes	Attributes
1	VelVento10m (m/s)	VelVento10m(m/s)	VelVentoMax(m/s)	VelVentoMax(m/s)	Bateria(Volts)	CorrPSol(Logico)	VelVentoMax(m/s)	DataHora(GMT)
2	DirVelVentoMax (oNV)	Bateria(Volts)	DirVento(oNV)	DirVento(oNV)	DirVento(oNV)	DirVento(oNV)	DirVento(oNV)	Bateria(Volts)
3	DirVento(oNV)	CorrPSol(Logico)	Pluvio(mm)	Pluvio(mm)	Pluvio(mm)	UmidInt(%)	Pluvio(mm)	ContAguaSolo100 (m3/m3)
4	UmidRel(%)	TempAr(oC)	VelVento10m(m/s)	VelVento10m(m/s)	VelVento10m(m/s)	UmidRel(%)	VelVento10m(m/s)	ContAguaSolo200 (m3/m3)
5	CorrPSol(Logico)	RadSolAcum (MJ/m2)	DirVelVentoMax (oNV)	DirVelVentoMax (oNV)	DirVelVentoMax (oNV)	VelVentoMax(m/s)	DirVelVentoMax (oNV)	ContAguaSolo400 (m3/m3)
6	ContAguaSolo400 (m3/m3)	DirVelVentoMax (oNV)	CorrPSol(Logico)	CorrPSol(Logico)	CorrPSol(Logico)	RadSolAcum(MJ/m2)	CorrPSol(Logico)	CorrPSol(Logico)
7	ContAguaSolo200 (m3/m3)	TempMin(oC)	ContAguaSolo400 (m3/m3)	ContAguaSolo400 (m3/m3)	ContAguaSolo400 (m3/m3)	ContAguaSolo200 (m3/m3)	ContAguaSolo400 (m3/m3)	DirVelVentoMax (oNV)
8	ContAguaSolo100 (m3/m3)	TempSolo100(oC)	ContAguaSolo200 (m3/m3)	ContAguaSolo200 (m3/m3)	ContAguaSolo200 (m3/m3)	DirVelVentoMax (oNV)	ContAguaSolo200 (m3/m3)	VelVento10m(m/s)
9	Bateria(Volts)	TempSolo400(oC)	ContAguaSolo100 (m3/m3)	ContAguaSolo100 (m3/m3)	ContAguaSolo100 (m3/m3)	VelVento10m(m/s)	ContAguaSolo100 (m3/m3)	--
10	Pluvio(mm)	ContAguaSolo200 (m3/m3)	PressaoAtm(mB)	PressaoAtm(mB)	PressaoAtm(mB)	TempAr(oC)	PressaoAtm(mB)	--

Figura 9. Feature Selection - Third Scenario Original Dataset

Selection Features For Third Scenario Attacked Data								
Algorithm	Classifier AttributeEval	Correlation AttributeEval	GainRatio AttributeEval	InfoGain AttributeEval	OneR AttributeEval	ReliefF AttributeEval	SymmetricalUncert AttributeEval	CfsSubsetEval
Ranked	Attributes	Attributes	Attributes	Attributes	Attributes	Attributes	Attributes	Attributes
1	VelVentoMax(m/s)	ContAguaSolo200 (m3/m3)	ContAguaSolo200 (m3/m3)	DataHora(GMT)	TempMin(oC)	ContAguaSolo200 (m3/m3)	ContAguaSolo200 (m3/m3)	Bateria(Volts)
2	DirVelVentoMax (oNV)	Bateria(Volts)	TempMin(oC)	ContAguaSolo200 (m3/m3)	ContAguaSolo200 (m3/m3)	TempMin(oC)	TempMin(oC)	ContAguaSolo100 (m3/m3)
3	DirVento(oNV)	TempSolo100(oC)	TempMax(oC)	TempMin(oC)	TempMax(oC)	TempMax(oC)	TempMax(oC)	ContAguaSolo200 (m3/m3)
4	PressaoAtm(mB)	TempSolo200(oC)	Bateria(Volts)	TempMax(oC)	CorrPSol(Logico)	CorrPSol(Logico)	Bateria(Volts)	CorrPSol(Logico)
5	CorrPSol(Logico)	VelVentoMax (m/s)	TempSolo400(oC)	CorrPSol(Logico)	UmidInt(%)	UmidInt(%)	TempSolo400(oC)	TempMin(oC)
6	ContAguaSolo400 (m3/m3)	TempMax(oC)	TempSolo200(oC)	UmidInt(%)	Bateria(Volts)	Bateria(Volts)	CorrPSol(Logico)	TempSolo100(oC)
7	ContAguaSolo200 (m3/m3)	DirVelVentoMax (oNV)	CorrPSol(Logico)	Bateria(Volts)	TempSolo400(oC)	ContAguaSolo100 (m3/m3)	TempSolo200(oC)	TempSolo200(oC)
8	ContAguaSolo100 (m3/m3)	TempSolo400(oC)	TempSolo100(oC)	ContAguaSolo100 (m3/m3)	ContAguaSolo100 (m3/m3)	TempSolo400(oC)	UmidInt(%)	TempSolo400(oC)
9	Bateria(Volts)	UmidInt(%)	UmidInt(%)	TempSolo400(oC)	ContAguaSolo400 (m3/m3)	ContAguaSolo400 (m3/m3)	TempSolo100(oC)	UmidInt(%)
10	Pluvio(mm)	TempMin(oC)	ContAguaSolo100 (m3/m3)	ContAguaSolo400 (m3/m3)	TempSolo200(oC)	VelVentoMax(m/s)	ContAguaSolo100 (m3/m3)	VelVentoMax(m/s)

Figura 10. Feature Selection - Third Scenario Attacked Dataset

and (C), and the results are shown in Figure 12. We verified the accuracy and the mean absolute error for all of them.

Corroborating the results of the previous scenarios, the attribute selection did not harm the classification accuracy, on the contrary, the accuracy improved from 97.56% (original dataset with no features selection) to 99.86% for the selection subset (A), 99.89% for the selection subset (B) and 99.93% for the top-5 attributes selection. We found an improvement in the Mean Absolute Error, which was reduced from 0.0448 to 0.0014, 0.0010 and 0.007 respectively. The same was observed regarding the execution time, which was also reduced from 0.33 to 0.14, 0.13 and 0.09 seconds for the same subsets.

Figure 13 shows a snapshot of the classification process output. The tool manages to identify the instances where the failures have been injected. So, by using the proposed process it is possible to filter different classes of information in the tests and monitoring results. The figure shown on the tool screen presents the distribution of classes, where each ”x”corresponds to a correctly sorted instance and the squares correspond to a incorrectly sorted instances. The adjacent windows present the view of the classification of each instance, with information about the predicted class and the actual class to which the instance belongs.

Selected Features For Third Scenario Original Data (A)	
Attribute	Weight
VelVento10m(m/s)	0,6625
DirVento(oNV)	0,6625
CorrPSol(Logico)	0,6125
DirVelVentoMax(oNV)	0,4500
VelVentoMax(m/s)	0,4500
Pluvio(mm)	0,4125
Bateria(Volts)	0,3750
ContAguaSolo200(m3/m3)	0,3000
ContAguaSolo400(m3/m3)	0,2750
ContAguaSolo100(m3/m3)	0,2000

Selected Features For Third Scenario Attacked Data (B)	
Attribute	Weight
ContAguaSolo200(m3/m3)	0,8750
TempMin(oC)	0,6500
Bateria(Volts)	0,6125
TempMax(oC)	0,5500
CorrPSol(Logico)	0,5250
TempSolo400(oC)	0,3375
UmidInt(%)	0,3250
ContAguaSolo100(m3/m3)	0,3000
TempSolo200(oC)	0,2625
VelVentoMax(m/s)	0,2250

Selected Features For Third 5-Top (C)	
Attribute	Weight
Bateria(Volts)	0,4938
ContAguaSolo100(m3/m3)	0,2500
ContAguaSolo200(m3/m3)	0,5875
CorrPSol(Logico)	0,5688
VelVentoMax(m/s)	0,3375

Figura 11. Feature Selection - Third Scenario Attacked Dataset

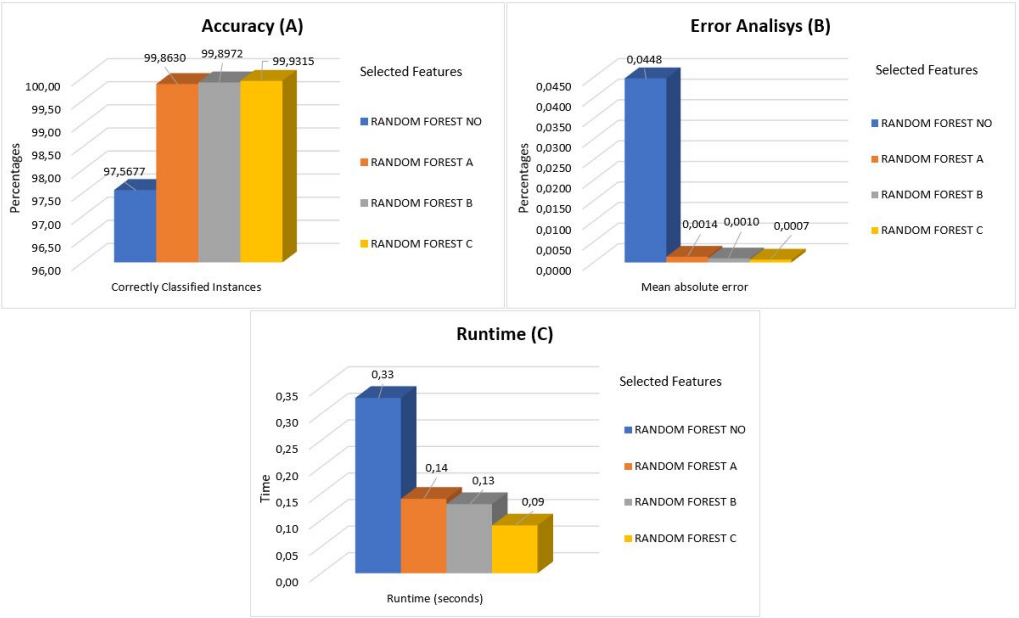

Figura 12. Ensemble Feature Selection - Third Scenario

Just for information, the experiments were performed on a computer with Windows 10 operating system, 10th generation Intel i5 processor, 256GB SSD disk and 8GB of memory.

5. Conclusion and Future Work

This work presents a feature selection study to automate the analysis of CPs test and monitoring result datasets by the support of machine learning. Three different scenarios that use large datasets from CPs were used in the analysis. The first dataset is reported by PLCs that monitors a water system controlled by SCADA, the second dataset was extracted from an IoT context including malicious data and the last dataset came from the satellite context that monitors environmental data, where some outliers values were injected to emulate malfunctioning sensors.

The results of the experiments were fundamental to understand how feature selection helps in the machine learning process and can help to identify anomalous instances provided by CPs. In the first scenario, the importance of an expert vision on the Initial Dataset Evaluation was demonstrated. Observing the large amount of identical data in the timestamp attribute the expert realizes that it is better to reorganize the data

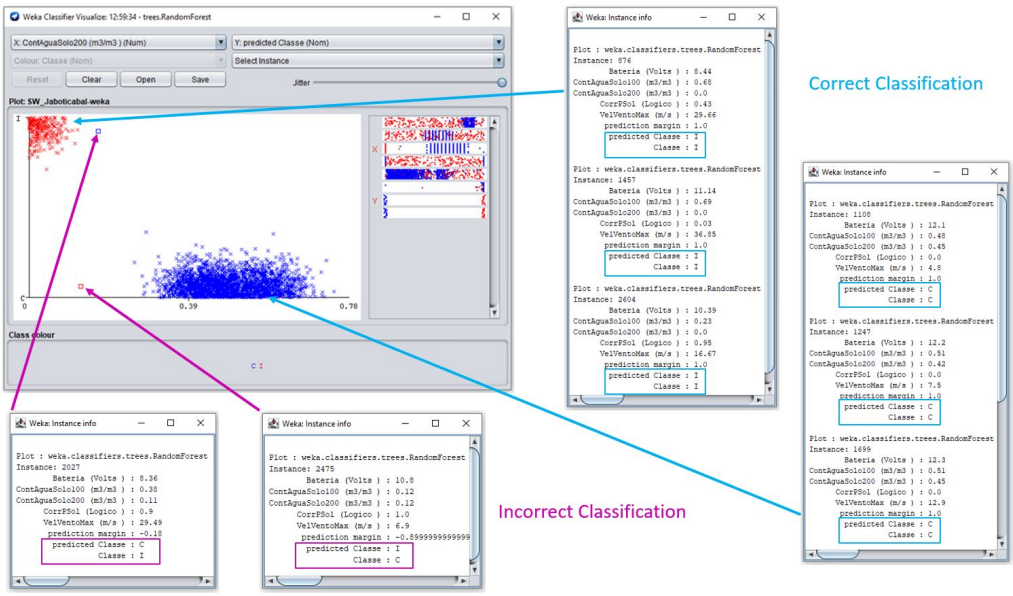

Figura 13. Process output example

and after regrouping the sensor data, the accuracy was improved. We believe that this process allows the application of the proposed method to data sets of the majority CPs. Furthermore, the experiments showed that not only high dimensionality is harmful to the classifiers performance but too small dimensionality is not beneficial either. It means that the initial dataset evaluation process in the proposed method benefits the classifier results. The reorganization of dataset instances changed the dimensionality from 3 to 12 attributes (still close to what is recommended in the literature, which is 10 attributes) and the gain in the accuracy was very relevant. We understand that although there is no magic number for feature selection, 10 attributes can be viable in an initial or automated analysis.

The second and third scenarios of the experiments showed that the Ensemble created for feature selection is feasible to automate the reduction of the dimensionality of datasets, without losing the accuracy of the classifier and improving the classification runtime. In these experiments, the classification runtime was almost 3 times shorter for the reduced subset.

The emulation of sensors malfunctioning through fault injection in the third scenario were important to understand that the Random Forest algorithm is a good option to automate the process of identifying anomalies in test and monitoring results provided by CPs, according to the proposed methodology. Feature selection resulted in a slight increase in accuracy for the top-10 and top-5 attributes subset. The top-5 used due to the evidence that for the emulated dataset the subset could be even smaller, did not show evidences of better accuracy when compared to the 10-Top subset, but the runtime was reduced by approximately 30% (0.13 to 0.09 seconds). Still, for this dataset, the top-5 or the top-10 subsets can be used in the methodology without prejudice.

The selection of the ideal number of features for each dataset can be studied in future work. The construction of a dashboard to monitor the identification of anomalous data in CPs, as well as the verification of the need to retrain the classification model, may be objects of future work. In the future, the proposed methodology can be incorporated

into a more comprehensive methodology to characterize a data quality score for machine learning.

Acknowledgment

This work has been partially supported by **ATMOSPHERE** (https://www.atmosphere-eubrazil.eu/ - Horizon 2020 grant agreement No 777154 - MCTIC/RNP) and **ADVANCE** (http://advance-rise.eu/ - Horizon 2020-MSCA-RISE grant agreement No 2018-823788) projects. Also, it was financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), Finance code 001.

Referências

[Angelis et al. 2006] Angelis, V., Felici, G., and Mancinelli, G. (2006). Feature selection for data mining. In *Data Mining and Knowledge Discovery Approaches based on Rule Induction Techniques*, pages 227–251. Springer.

[Bolón-Canedo and A.Alonso-Betanzos 2019] Bolón-Canedo, V. and A.Alonso-Betanzos (2019). Ensembles for feature selection: A review and future trends. *Information Fusion*, 52:1–12.

[Hindy et al. 2018] Hindy, H., Brosset, D., Bayne, E., Seeam, A., and Bellekens, X. (2018). Improving siem for critical scada water infrastructures using machine learning. In *International Workshop on the Security of Industrial Control Systems and Cyber-Physical Systems - SECPRE*, pages 3–19. Springer.

[Husna and Adiwijaya 2018] Husna, A. and Adiwijaya, A. (2018). A clustering approach for feature selection in microarray data classification using random forest. *Information Process Systems*, 14:1167–1175.

[Kumar 2021] Kumar, S. (2021). Automate your feature selection workflow in one line of python code. URL: https://towardsdatascience.com/automate-your-feature-selection-workflow-in-one-line-of-python-code-3d4f23b7e2c4 [Last access on June, 2021].

[Mafarja and Mirjalili 2018] Mafarja, M. and Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, 62.

[Pipino et al. 2019] Pipino, L. L., Lee, Y. W., and Wang, R. (2019). Data quality assessment. *Computer Reviews Journal*, 4.

[Rothermich J. 2021] Rothermich J. (2021). Finding machine learning ready data. URL: https://www.refinitiv.com/perspectives/ai-digitalization/finding-machine-learning-ready-data/ [Last access on May, 2021].

[Sraavnthi et al. 2019] Sraavnthi, K., Shamila, M., and Kumar, T. A. (2019). Cyber physical systems: The role of machine learning and cyber security in present and future. *Computer Reviews Journal*, 4.

[Tang et al. 2014] Tang, J., Alelyani, S., and Liu, H. (2014). *Feature Selection for Classification: A Review*, pages 37–64. Number 5.

[Vidyavathi 2019] Vidyavathi, B. M. (2019). A new approach to feature selection for data mining. *Computational Intelligence Research*, 7(3).