FIREMAN

WP2 Deliverable 2.3
Important Rare Events Identification

Project Title:	Framework for the Identification of Rare Events via MAchine learning and IoT Networks

Title of Deliverable:	Important Rare Events Identification
Status-Version:	Final-1.0
Delivery Date:	30/4/2020

Contributors:	Daniel Gutierrez-Rojas (LUT), Pedro Nardelli (LUT), Ioannis T. Christou (AIT), Jean M. Sant'Ana (Oulu), Hirley Alves (Oulu), Merim Dzaferagic (TCD)
Reviewers:	i) Asst. Prof. Renan Moioli, Federal University of Rio Grande do Norte, Brazil, ii) Asst. Prof. Mateus Giesbrecht, State University of Campinas, Brazil
Approved by:	All Partners

Document Revision History

Version	Date	Description
Version under Internal Review	10 April 2020	Internal Review
Version under External Review	17 April 2020	External Review
Final Version	30 April 2020	Approved Form of Deliverable D2.3

Summary

This document describes the progress made so far in identifying rare events in industrial processes with the ultimate aim of addressing Predictive Maintenance objectives. We have introduced different methods for identifying rare events in datasets with a main focus on the QARMA algorithm. We present two sets of numerical results related to *i*) the Tennessee Eastman Process and *ii*) a fault selection on power grids. The research is performed within the context of the EU CHIST-ERA project FIREMAN.

Table of Contents

1 Introduction

1.1 Objective of the document

The objective of Deliverable 2.3 is to collect and document in a coherent manner all results of the research that has been carried out in the field of rare-event identification using Machine Learning (ML) and Data Mining (DM) techniques, within the context of FIREMAN project. Some further explanations are in order: with the term "rare event" we not only mean something that happens relatively infrequently, but also something that has an effect on an industrial process, that will eventually cause a need for some maintenance or other corrective action. In this context, therefore, rare-event identification certainly includes learning models that detect as early as possible the existence of a fault in an industrial process, and in fact it is this modeling and algorithmic development that plays an important role in this research. However, the consortium partners also seek to develop models that will learn *root causes* of "tear-and-wear" processes that eventually cause significant faults in a process that are significant from a production engineering point of view.

1.2 Structure of the document

The remainder of this document is structured as follows: Section 2 describes important rare events in general, with the help of dimensionality reduction techniques. Section 3 describes ML- and DM-based methods and pays particular attention to the QARMA algorithm and its variants (i.e., R4RE) for detecting rare events; here the focus is essentially on models that can learn accurately minority (i.e., rare) classes in the presence of significant class imbalances in the dataset. Section 4 contains preliminary numerical results along with a discussion on the quantitative insights of the proposed algorithms. Finally, Section 5 provides our concluding remarks.

2 Important Rare Events

2.1 Definitions and Preliminaries

Data dimensionality refers to the cardinality of the feature set used to describe a dataset. Due to the vast amount of available data used to describe a certain event, dimensionality reduction arises as a common necessity to improve the efficiency of learning algorithms. Considering that not all characteristics of an event are relevant to its analysis, dimensionality reduction allows us to reduce the cardinality of the feature set without loosing a lot of actual information about the event under analysis. There are several techniques for feature selection and identification of the most relevant features that allow to model the event – often these techniques preserve the meaning of the selected features; as well as, feature extraction that reduces the dimensions of the data. Principal Component Analysis (PCA) is a dimensionality reduction approach that applies orthogonal transformation to convert correlated sets of features into a linear uncorrelated one. Similarly, Linear Discriminant Analysis (LDA) describes dependent feature as a linear combination of other features. An overview of those techniques and their variations is provided in [1].

2.2 Dimensionality reduction, key features and relations

Dimensionality reduction in the context of FIREMAN is related to the amount of processing required for fault detection; in this regard, time-series compression by event-driven approaches, as in [2], allows data compression before the transmission. This technique, not only reduces the amount of traffic generated by the network which in turn leads to better resource allocation and interference handling, but also reduces the amount of redundant data in the aggregator and extends the lifetime of the wireless devices [3]. In [2], the authors propose devices in a smart grid scenario to transmit only when two events are identified: *i*) a certain amount of energy consumption is reached; or *ii*) there is a sudden change in the power demand. They compare these results with a time-based scheme with 30 minutes of sampling period. The event-based scheme presented a better signal reconstruction compared to the time-based. In their results, they adjusted their threshold parameters such that devices would transmit a similar amount of messages per day. Thus, it is important to note that the event-based scheme requires more information regarding the measured process. Hence, they propose usage of real-time algorithms that could adapt to these thresholds with historical data. Another technique with similar objectives is using Graph Signal Processing, as in [3], which allows the downsampling of information around correlated devices. Here, the authors divide the sensors into several subsets that are able to reconstruct the desired signal. Their novel sampling method focuses on increasing the network lifetime and ensures that all devices perform the same amount of transmissions on average, thus presenting similar energy consumption behavior.

The random forest algorithm is based on a smart way of exploiting a collection of decision trees. Intuitively, a decision tree can be thought of as a series of yes/no questions asked about a dataset which would eventually lead to a classification of an input sample. A decision tree is formed by determining the questions that, when answered, lead to the greatest reduction in Gini Impurity. Gini Impurity is a measure of how often a randomly chosen element from a set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in a subset of a tree. However, since the depth of the tree affects the variance and the bias, to limit the depth of the tree we combine many decision trees into a single ensemble model known as a random forest. The key concepts related to the random forest approach are:

1. Random sampling of training data points when building trees;

2. Random subset of features considered when splitting nodes.

Random Forests are often used for feature selection in a data science workflow, because they naturally rank the features by how well they improve the purity of the node. Nodes with the greatest decrease in impurity happen at the start of the trees, while nodes with the least decrease in impurity occur at the end of trees. By pruning trees below a particular node, we can create a subset of the most important features.

2.2.1 Selecting Features Given an Extracted Rule-Set

It is also worth noting that in the context of FIREMAN, we have developed a hybrid greedy/Breadth-First Search (BFS) approach that takes as inputs an extracted set of rules that apply on a given set; and it outputs a near-minimum cardinality set of variables that are used in an appropriate subset of the input rules with the property that this subset of the rules covers at least a (user-defined, e.g., 95%) percentage of the training instances that the entire input rule-set covers. Of course, for this technique to work, and algorithm such as QARMA (presented in details in Section 3) that guarantees to find a rule-set that maximally covers a training dataset must have run before to produce the required input rule-set.

An implementation of this hybrid algorithm mentioned above is open-sourced, available in the package popt4jlib.MinExplainSet in the library (`https://github.com/ioannischristou/popt4jlib`). The pseudo-code is given in Algorithm 1, where n_{min}, K are user-defined values for controlling the search.

Algorithm 1: Min. Explaining Variables Algorithm

```
$sols \leftarrow \emptyset$;
$Q \leftarrow \{\emptyset\}$;
while $|Q| > 0$ do
    $vars \leftarrow Q.pop()$;
    $vars.add(targetVar)$;
    $satInsts \leftarrow getSatisfyingInstances(vars)$;
    if $|satInsts| > n_{min}$ then
        $sols.add(vars)$;
        return $getBest(sols)$;
    $vars.remove(targetVar)$;
    $candVars \leftarrow getBestOfRemainingVars(vars, K)$;
    while $|candVars| > 0$ do
        $v \leftarrow candVars.pop()$;
        if $index(v) \leq max_{a \in vars} index(a)$ then
            continue;
        $Q.add(vars \cup \{v\})$
return $getBest(sols)$
```

As shown in Algorithm 1, the function $getBest(sols)$ accepts as input a set of sets of variables and returns the one among them that has maximum training set coverage. The function $getSatisfyingInstances(vars)$ receives as input a set of variables and returns the set of training instances that satisfy at least one of the rules whose preconditions and postcondition are a subset of $vars$. Finally, the function $getBestOfRemainingVars(vars, K)$ returns the top K variables x that are *not* in the set $vars$ when the variables are sorted in ascending order of the quantity $(|var| + 1)/(getSatisfyingInstances(vars \cup \{x\}) + 1)$ and represents a greedy evaluation of the current cost of adding a variable to a candidate set.

FIREMAN adopts time-series compression as proposed by the event-driven approach, in addition to a Mutual Information (MI) entropy reduction technique that is used to infer how variables are correlated. This technique is detailed in Section 3.

3 Machine Learning Methods and QARMA

ML methods are statistical methods in nature whose main goal is to discover –statistically significant– dependencies between variables in data. Over the years, a vast array of different methods have been developed ranging from methods that are applicable to most datasets to highly specialized methods that are developed under specific assumptions on the nature of the input datasets; for example, many methods have been developed for datasets for which the feature variables -other than the class, or target variable- are all assumed to have no dependencies to each other. Most methods are also developed under the assumption that all classes are more or less equally represented in the training dataset (balancing assumption), etc. Rule-based learners, are particularly well-suited to rare-event identification exactly because they do not make the class-balance assumption just mentioned, which clearly does not hold when we seek to identify very small minority classes.

3.1 QARMA

QARMA is an efficient novel cluster–parallel algorithm for mining Quantitative Association Rules with a single consequent item, and many antecedent items with many different, not necessarily the same, attributes, in large multi–dimensional datasets [4],[5]. Using the standard support-confidence framework of Qualitative Association Rule Mining, it extends the notions of support, confidence and many other "interestingness" metrics so that they apply to quantitative rules. QARMA is built on a set of premises, as follows:

1. Items of any valid quantitative rule (antecedents plus consequent), must form a frequent itemset in the classical Boolean Association Rule Mining sense in the dataset for the given minimum support threshold specified; therefore, to produce all valid quantitative rules, it is sufficient to consider only rules whose items form a frequent itemset, as produced by an algorithm for frequent itemset mining in transactional databases stripped of any quantitative information, e.g., FP-Growth.

2. Producing all valid quantitative rules with given support and interestingness levels in a dataset is at face value an impossible task: indeed, if there is a single valid quantitative rule, there are infinitely many different quantitative rules that have the required support and interestningness level thresholds on the (finite) dataset: to see how this is so, given such a rule, simply subtract any sufficiently small number $\epsilon > 0$ from any antecedent item's quantified attribute value, to obtain a rule with the same support and interestingness as the original one. This leads us to the notion of rule dominance as a criterion for excluding from the result such useless rules.

3. The anti-monotone property of a rule's support (analogous to the same property in classical frequent itemset mining): given a quantitative rule, restricting any of the rule's item attributes to values above any threshold, can only decrease the rule's support. Thus, as soon as an item attribute in a rule restricts it so much that its support drops below the minimum required level, any further restrictions of this partially quantified rule are useless, and so the rule is discarded from further processing.

4. Based on an appropriate definition of rule dominance, a valid QAR r specifying a set of antecedent items B and a consequent item I having length $l = |B \cup I|$ may only be dominated by a rule of smaller length, or another rule having the same antecedents and consequent item as itself. Combined with the transitive property of dominance, this means that if all valid non-dominated rules of length less than l are generated before r is considered, and if all other quantifications of the base qualitative rule that restrict the rule less than the current restrictions have already been examined, then if r is not dominated by any of the rules that have already been found, the rule is guaranteed to be a valid non-dominated rule.

5. The fifth (and final) observation is that in order to generate all valid non-dominated rules, it is sufficient to consider quantifications of a rule considering only as value thresholds for the item attributes, values that actually appear in at least one transaction of a user history.

QARMA can be configured to produce rules of the form $I_1.attr_1 \in [l_{1,1}, h_{1,1}] \wedge \ldots I_n.attr_m \in [l_{n,m}, h_{n,m}] \implies J_0.p \geq v$ or alternatively to produce rules of the form: $I_1.attr_1 \in [l_{1,1}, h_{1,1}] \wedge \ldots I_n.attr_m \in [l_{n,m}, h_{n,m}] \implies J_0.p = v$; the latter form is very useful when it comes to supervised classification problems where the value of the target item attribute is essentially the class variable that we seek to learn. The QARMA algorithm works as follows: first, we apply any algorithm for generating all frequent itemsets in the input dataset according to the specified minimum support sought, and the frequent itemsets are partitioned into itemsets of increasing length (cardinality.) Then, the algorithm proceeds sequentially to produce all valid quantitative association rules from each itemset of length 2, then 3, then 4, etc. Within each phase of producing all valid rules of length $l = 2, 3, ...$, the algorithm considers in parallel all frequent itemsets of length l. For a given itemset, it produces all possible rules (with each attribute in the rule being unquantified in the beginning); for each such initially unquantified rule, a possibly different CPU core maintains a local rule set R (initially empty) and runs a modified BFS procedure that first assigns the consequent attribute to the highest possible value, and, as long as the resulting partially quantified rule has support above the threshold required, it adds it to a queue data structure Q. While this queue is not empty, the first rule inserted in the queue is retrieved and removed from the queue, and for each attribute that has not been quantified in it yet, the algorithm creates as many new rules as there are different values in the dataset for the attribute being examined in ascending attribute order value and enter the queue Q in this order, but only if the newly quantified rule exceeds the minimum support requirement. If the partially quantified rule also meets minimum confidence (or any other metrics set), then it is checked against the current set of local rules R to see if it is dominated by another rule in R. If no other rule in R dominates the current rule, the current rule is added in the set R. After having ran this BFS process in parallel for all frequent itemsets of length l, the various CPUs participating in the run, synchronize so as to get all rules from all the other ones, before moving to process the frequent itemsets of length $l + 1$.

Once the set of all non-dominated rules is computed, a classifier based on an ensemble of these rules works as follows:

1. Select all rules whose antecedent conditions are satisfied by this instance and add them to the set F;

2. Sort the ruleset F in decreasing order of confidence and decreasing order of support on the training set;

3. Remove all but the top-100 rules of the sorted set F;

4. Each rule in F carries a weight equal to its confidence on the training set; decide as the instance's class the weighted majority vote of the rules in F.

3.2 QARMA Benefits

As discussed above, QARMA –and its variant, R4RE, detailed in [4]– have a number of properties that make it particularly attractive for this particular domain of application. Its results are by default explainable: rules of the form $< precondition - set > \Longrightarrow < postcondition >$ are immediately understood by all humans, experts and non-experts alike, therefore QARMA belongs to the domain of so-called eXplainable AI (XAI). QARMA also has the nice property of producing rules that maximally cover the training set, a feature that is not only of theoretical interest, but also of practical one, because it allows the development of the greedy/BFS algorithm, described earlier in subsection 2.2.1, that can produce a minimal set of attributes that "explain" maximally the dataset. Last but certainly not least, QARMA may extract rules with arbitrarily (user-defined) small support and large confidence, making it ideal for datasets with very large class imbalances.

4 Numerical Results

4.1 Tennessee Eastman Process

4.1.1 Event-driven Data Acquisition

The main idea of an event-driven approach for this application is to perform data compression in order to transmit the meaning of information from the data acquisition point to the data fusion point. This approach can be described using the following steps: *i*) input data from all 52 sensors (N); *ii*) variable average estimation and margin selection (90% of lowest/highest values) from normal operation; *iii*) at every time slot (k) for each variable, if the values are out of the margins the sample is transmitted, otherwise, if nothing is received at the data fusion point the variable will maintain the average value estimated from the previous step; *iv*) compression rate calculation for each variable. The limit values for margin selection for each variable were chosen arbitrarily. An example of this approach is seen in Fig. 1 where the signal obtained from the sensors is shown with its limits. The samples transmitted are only the ones that are out of the upper and lower limits as seen in Fig. 2. This setup allows to transmit less data via any communications system. In the example mentioned above the compression rate is 92.60%, this means only about 7.4% of the samples are transmitted. The pre-processed time series based on the proposed event-driven method will serve as inputs to the data fusion and analytics, where anomalies should be detected, identified and diagnosed.

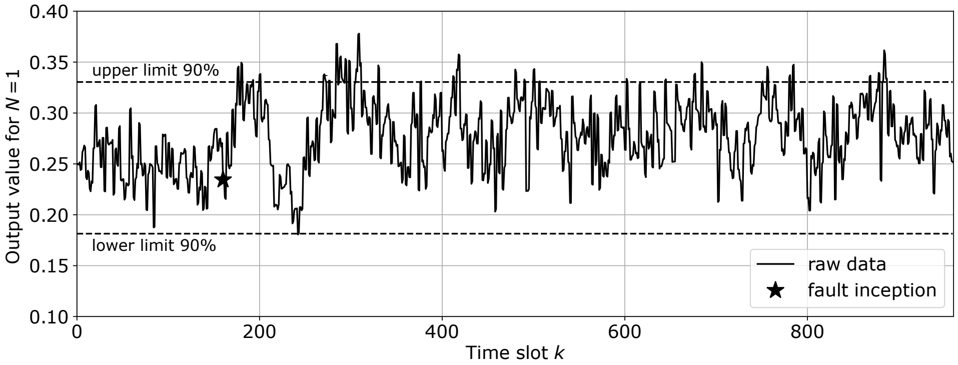


Figure 1: Signal from variable 1 at data acquisition point (before transmission) for fault number 2 of Tennessee Eastman dataset.

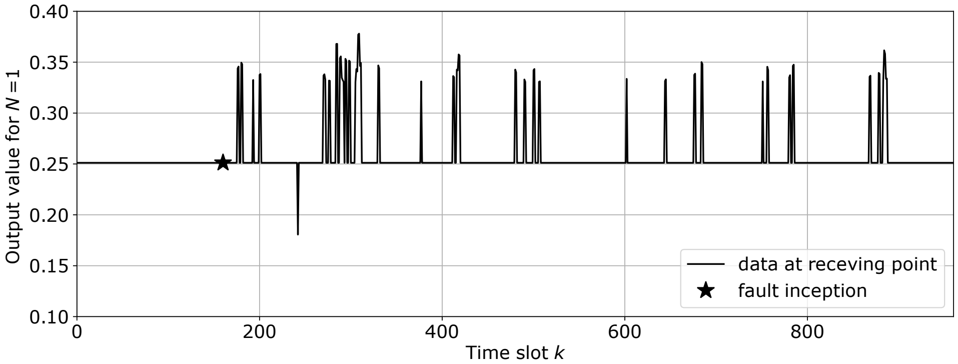


Figure 2: Signal from variable 1 at data fusion point (after transmission) for fault number 2 of Tennessee Eastman dataset.

4.1.2 Data Fusion based on Mutual Information

In order to further reduce the amount of processing required for fault detection, on top of the time-series compression as proposed by the event-driven approach, the proposed framework determines how the process variables are related to each other to discover and exploit their dependencies. The interdependencies between the process variables are determined automatically from the sampled measurements in the Tennessee datasets. Specifically, MI entropy reduction technique is used to infer how variables are correlated. The MI quantifies the amount of information that each variable contains about the other ones [6]. The tools used to determine correlations among the variables used in this work represents the distances between variables in terms of their statistical closeness, then it quantifies the correlation by providing links between the variables. Finally, it assigns directionality to the links [7]. Fig. 3 shows that aside from the auto-correlation, strong cross-correlation is present between a high number of variables; in fact, a high correlation above 80% is present in 23% of the variables (12 out of 52) while a modest correlation above 50% is present in 65% of the variables (34 out of 52).

As an example of statistical closeness, Fig. 3 demonstrates the correlation for the 52 process variables in the Tennessee datasets. This result showcases that not every single variable needs to be observed at every given moment but, depending on the fault under investigation, it is

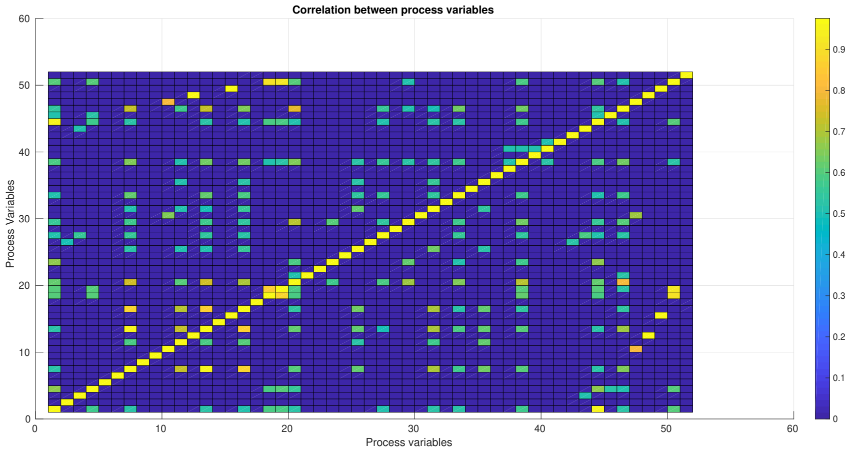


Figure 3: Correlation between Process Variables

possible to observe variables carrying a high amount of MI with other variables involved in the process. This reduces both data and computational load considerably.

4.1.3 Quantitative Association Rule Method for Anomaly Detection

Quantitative association rule mining is a natural extension of classical qualitative association rule mining where the difficult task is the extraction of frequent itemsets from a dataset containing transactions. The extracted rules are statistical rules of the form $Item_1 AND Item_2 ... AND Item_k \implies Item_n$, that hold with certain support and confidence values (or other metrics) that are above user-defined thresholds. Association rule mining is one of the most heavily researched areas in data mining to this date. In our context, items correspond to the features in our dataset, and transactions are in one-to-one correspondence with the rows of the dataset; the consequent item ($Item_n$) in particular is constrained to always be the target variable in our dataset. The problem then becomes one of quantifying each item in each (qualitative) rule extracted from the dataset by constraining its value to lie in a specified interval, so that the target variable assumes a particular value. This is achieved by a modified parallel BFS algorithm, i.e., QARMA, which guarantees that all (and none other) non-dominated quantitative association rules that hold in the dataset will be found (see [8]).

In the case of the Tennessee dataset in particular, the dataset is fully dense in the sense that every row contains values for every dataset feature, which makes the application of the qualitative association rule mining part useless. Instead, we construct all itemsets of size less than 4, and quantify each one of them separately, and in parallel, making sure that all itemsets of size s are fully processed before starting to process itemsets of size $s+1$. To take into account the time-dependent nature of the data, whereby the values of any feature in the dataset are to some degree dependent on the values at the immediate previous times, we expanded the data to include for each feature, the difference between the feature's value and the feature's value in the previous two time-steps, resulting in a dataset having 156 different fully dense features. The QARMA algorithm took several days to run on this expanded Tennessee dataset, producing a total of 63008 non-dominated rules, that could predict all different modes of operation except mode 0 (normal operation), and hard-to-detect faults 3, 9 and 15 (none of the 63008 rules imply faults 3, 9 or 15). The distribution of the rules among the faults is also highly skewed: faults 6 and 18 are implied by 23624 and 23884 rules respectively, whereas faults 5, 10 and 21 are implied by 9, 3, and 8 rules respectively. During testing, an instance for which more than 10 rules fire, is predicted to belong to the fault that is specified by the majority of rules firing on that instance. Using this majority vote rule ensemble, we obtained an overall rule-accuracy that exceeds 62% on the test set. This accuracy is significantly better than the one reported by decision trees (J48), or artificial neural networks (MultiLayerPerceptron) as implemented in the WEKA ML/DM software suite, all of which reported accuracy less than 50% on the test set.

We also implemented another modified BFS algorithm to search to find a minimum cardinality set of variables that contain all the variables necessary for an appropriate subset of the discovered rules to cover 85% of all instances that the entire set of rules cover; we say that "rule r covers instance i" if and only if in the instance i the values of the features that form the antecedents in rule r are within the intervals specified for them by the rule, and the instance indeed belongs to the operation mode (fault number) that is predicted by the rule. Interestingly, only 14 of the 156 variables are enough to "explain" 85% of the entire dataset covered by the rules found; this result implies that possibly a much smaller set of variables need be monitored in order to derive safe conclusions about the state of the process. The total rules found covered more than 70% of the training set. We consider our first results as encouraging in that rules using only up to 2 features at a time are able to form an ensemble of rules that outperformed other well-known ML algorithms in test-set performance. We expect that dimensionality reduction will allow larger number of antecedent features to be examined and eventually provide much higher accuracies measured by detection rates/false alarm rates per class.

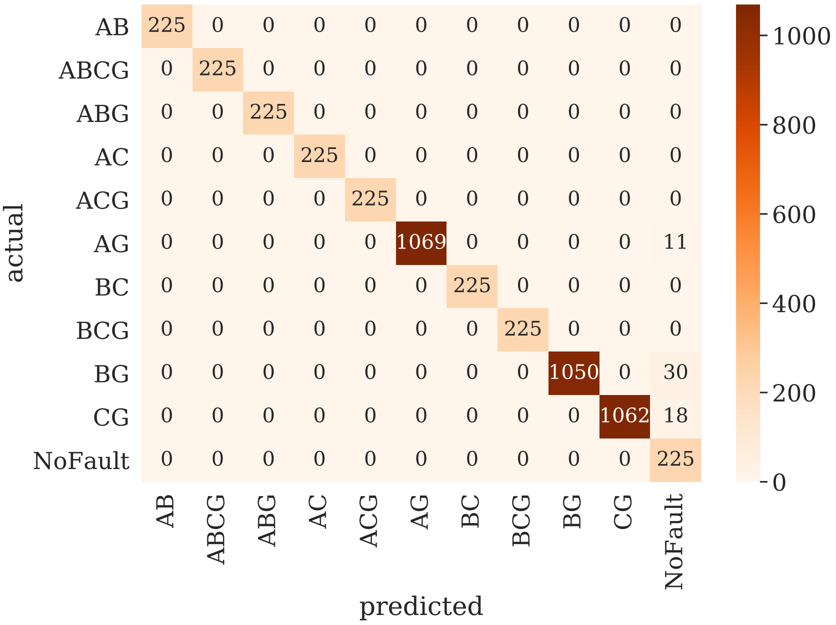
4.2 Fault Classification in Power Grids

The generated dataset was split into two subsets: 75% of a random shuffle of the dataset was kept for training and the remaining 25% was used to validate the accuracy of the trained models. The exact same split was used for all experiments with all different algorithms. Table 1 shows the results of running several well-known ML algorithms for supervised learning on the produced dataset, and Fig. 4 shows the results for the classification task. The accuracy achieved with the Deep Learning (DL) model setup was remarkably high, 98.47%. This high accuracy is due to the large size of the simulated fault dataset and equally importantly because of the *balance between the sample sizes of the various classes*; the strong success of the DL model is also because all the voltages and currents from both lines were available, including neutral currents. Further, perfect communication and without any problems related to latency, availability, or synchronization was considered. With this setup, the importance of availability of all features was tested. Table 2 lists the number of features tested, and Fig. 5 shows the results with Artificial Neural Network (ANN). With fewer features, the ANN does not perform as well, emphasizing the importance of neutral current estimation. However, when only one-end currents and voltages are available, the accuracy of the algorithm is still adequate for the task.

QARMA was also ran on the same training set with the user-defined support threshold

Table 1: ML Results on the Fault-Grid Dataset

Classifier	**Accuracy**
Decision Tree	94.62%
ANN (1 hidden layer)	95.18%
ANN (2 hidden layer)	**98.47%**
SVM	89.05%
Ripper-k	86.17%
Naïve Bayes	59.42%
Logistic Regression	78.47%
AdaBoost.M1	17.81%
QARMA	**98%**



	AB	ABCG	ABG	AC	ACG	AG	BC	BCG	BG	CG	NoFault
AB	225	0	0	0	0	0	0	0	0	0	0
ABCG	0	225	0	0	0	0	0	0	0	0	0
ABG	0	0	225	0	0	0	0	0	0	0	0
AC	0	0	0	225	0	0	0	0	0	0	0
ACG	0	0	0	0	225	0	0	0	0	0	0
AG	0	0	0	0	0	1069	0	0	0	0	11
BC	0	0	0	0	0	0	225	0	0	0	0
BCG	0	0	0	0	0	0	0	225	0	0	0
BG	0	0	0	0	0	0	0	0	1050	0	30
CG	0	0	0	0	0	0	0	0	0	1062	18
NoFault	0	0	0	0	0	0	0	0	0	0	225

Figure 4: Fault classification task confusion matrix

of 3.5% and the confidence threshold of 90% to obtain 5333 rules covering 97.8% of the entire training set. Then, a slight variant of the decision making algorithm described in the previous section, based on weighted voting, was used: for each instance in our test set, as long as the instance is covered by more than 100 rules, the instance's class is decided upon by the majority vote of the *top 10 firing rules having the highest confidence on the training set; instances that fail the minimum coverage requirement are not classified*. This algorithm resulted in an accuracy more comparable with the one obtained by DL, around 98% but at the following cost: a longer training time (around 2.5 h of wall clock time on a 3rd generation Intel core i-7 CPU with 8 logical cores). For a small percentage of testing instances, approximately 4%, QARMA was not able to provide a decision, because of the small number of rules firing on them. However, we expect that QARMA and its decision-making components will compare equally well or even outperform DL techniques in training sets that are more highly skewed.

Table 2: Feature selection on original dataset

Round	Feature
1	all features (local and remote current and voltages including I_R)
2	$L_{I_A}, L_{I_B}, L_{I_C}, L_{I_R}$
3	$R_{I_A}, R_{I_B}, R_{I_C}, R_{I_R}$
4	$L_{I_A}, L_{I_B}, L_{I_C}, L_{V_A}, L_{V_B}, L_{V_C}$
5	$R_{I_A}, R_{I_B}, R_{I_C}, R_{V_A}, R_{V_B}, R_{V_C}$
6	$L_{I_A}, L_{I_B}, L_{I_C}$
7	$R_{I_A}, R_{I_B}, R_{I_C}$
8	$L_{V_A}, L_{V_B}, L_{V_C}$
9	$R_{V_A}, R_{V_B}, R_{V_C}$

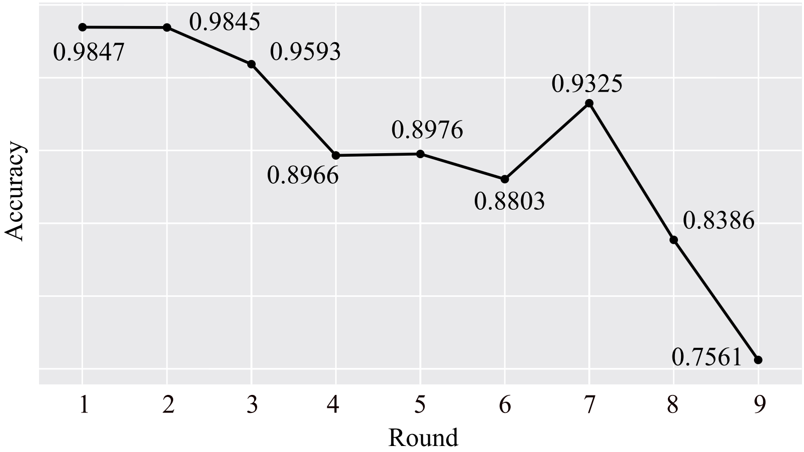


Figure 5: Fault classification task confusion matrix

On the other hand, QARMA produces as a model a set of quantitative rules that are much easier to understand and reason about than most other models, and DL models, in particular. This makes QARMA results much easier to explain to humans than any other model. Every extracted rule is trivially checked against the training dataset for validation purposes, and is also trivial to understand "what it means," as the precondition of the rule are nothing more than a conjunction of restrictions of the attributes that comprise the rule's antecedents to certain intervals. This ease of understanding of rules is what has made them particularly attractive since the beginning of the AI and ML research. In fact, already since the 1980s, there have been attempts to extract the knowledge that is embedded in neural network models into sets of rules [9] as such rule sets were recognized from the beginning as the most obvious knowledge representation that can exist. Therefore, QARMA is, in general, a particularly good fit for the newly emerging XAI paradigm, the term "explainable" meaning that the resulting model that the algorithm produces can be easily understood by humans.

We also ran an symmetrical algorithm method using the dataset for this paper. The method is used by one top relay manufacturer. The results can be seen in Fig. 6. The accuracy of this method for single phase faults can is represented in Table 3 and also the false positive single phase detection. False positive in this context is defined as the number of single-phase

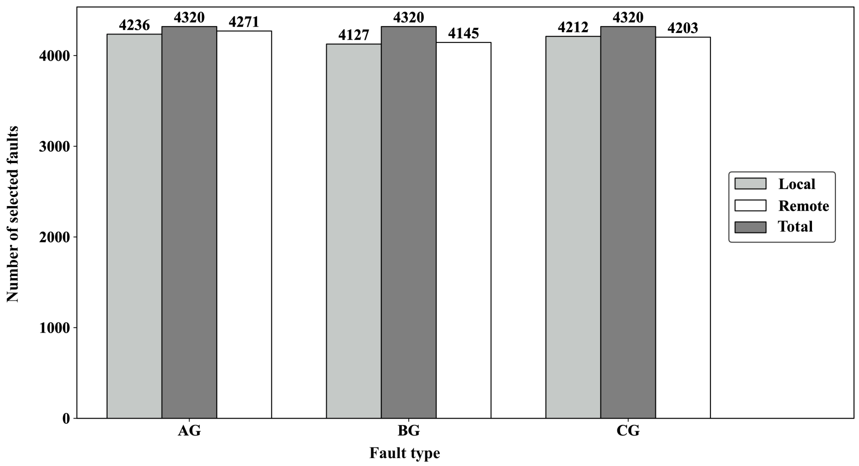


Figure 6: Fault classification by symmetrical angle method

Table 3: Results obtained by replicating symmetrical method

	Dependability		**Security**	
	Local end	Remote end	Local end	Remote end
AG	98.05%	98.86%	910	910
BG	95.53%	95.94%	896	909
CG	97.50%	97.29%	880	880

faults selected by the symmetrical method given that in the real fault was involved at least 2 phases. Under those conditions, the classification strategy takes the system to a situation less secure than with a tripolar tripping. The results shown in Fig. 4 demonstrate that errors on the proposed method occurred in the lack identification of some faults. Since fault selection systems are meant to be associated with protection algorithms, those errors would cause an unnecessary tripolar breaker opening - security error. Considering an interconnected system, security errors are less likely to cause system wide power outages than protection dependability errors. Therefore, the proposed solution would promote better system stability than the traditional method's results in Table 3.

5 Conclusions

This report investigated the identification of rare events, mainly focused on the performance of QARMA algorithm. Our results indicated that QARMA is capable of identifying in an explainable manner the relevant relations between features while also indicating ways to reduce the dimensionality of the dataset. Numerical results indicated that QARMA is at this point the most appropriate candidate to be deployed in the solution to be proposed by FIREMAN. This deliverable related to T2.3 provides valuable inputs to WPs3-5.

Appendix

Tennessee Eastman model

The dataset material consists of several faulty cases of an industrial plant, as produced by the Tennessee Eastman problem [10]. The process has five major equipments, namely a condenser, a vapor-liquid separator, a reactor, a product stripper, and a recycle compressor (Fig. 7). Its objective is to obtain the products G and H from the reactants A, C, D and E. This is reached by a set of four chemical reactions, in which components B and F are, respectively, an inert and a byproduct. More details can be found in [11, 12]. This benchmark is suitable to evaluate process monitoring schemes and control strategies based on data driven analysis. Besides the normal operation, 21 abrupt or incipient faulty conditions caused by common disturbances in practice are simulated [13]. There are 52 monitoring variables or features, being 11 manipulated variables and 41 measured variables. Once a faulty condition occurs, all are generally affected with changes in their respective values.

The Tennessee dataset was generated in a process simulator that has been widely used by the process monitoring community[1]. It is composed by 22 subsets named *dXX_te.dat*, where *XX* $= 0, 1, 2, \cdots, 21$. The file *d00_te.dat* refers to the normal operating condition. Each of the other ones regards to a particular fault, that is, a different shift from this reference condition. The subsets consist of 960 observations of the 52 variables, which are sampled every 3min with a Gaussian noise. The faults are introduced after 8 simulation hours. Table 4 presents the proposed framework applied in TEP, which provides the boundary conditions of the anomaly detection design.

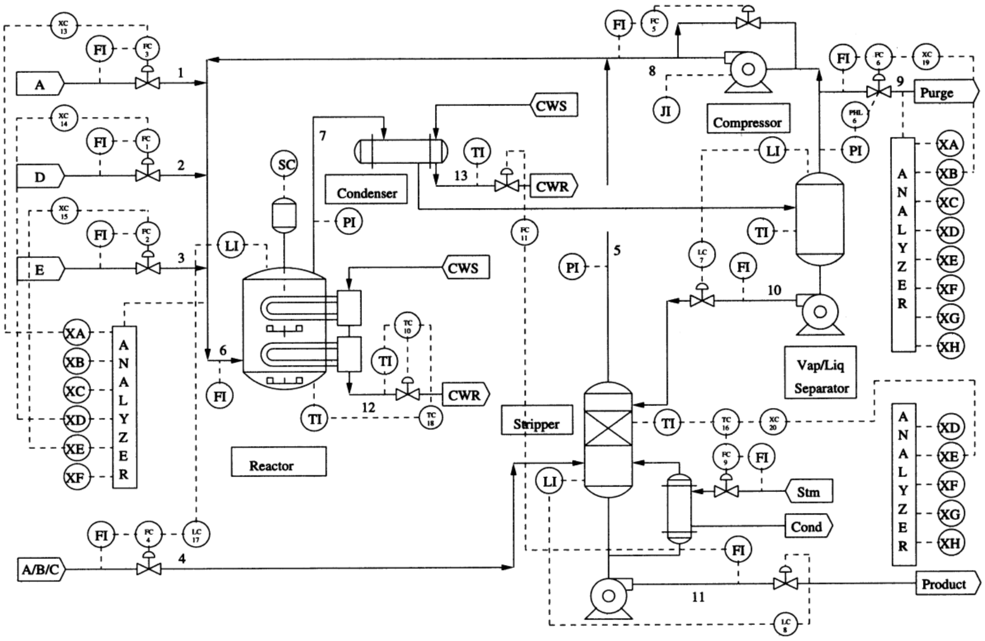


Figure 7: Process flow diagram of the Tennessee Eastman problem [12].

[1] https://github.com/camaramm/tennessee-eastman-profBraatz

Table 4: Proposed framework applied in TEP

Q#	Answer
Q0	Lack of accuracy to detect anomalies (21) from measurements
Q1	Each of the 52 variables are sensors, no other can be added
Q2	Periodic sampling; all 52 variables are sync (3 min.)
Q3	It can be considered as in [14]
Q4	Not constrained; freedom to test as in [15]
Q5	Open question; focus of research in the field
Q6	Open question; focus of research in the field

Fault classification model

A 400 kV, 50 Hz power system (Fig. 8) was simulated to extract features and then generate the dataset of currents and voltages based on the DFT at a fault point (when there is a fault). The electrical system under study is composed of a double-circuit transmission line typical, for example, in Finland and other European countries. These types of lines represent a challenge for correct fault identification and selection owing to the strong impact of mutual impedance on the fault resistance. As for the communication channel, data were gathered by Intelligent Electronic Devices (IED) from both ends and sent via a wireless link (e.g., 4G or 5G) to the fault selector, as shown in Fig. 8.

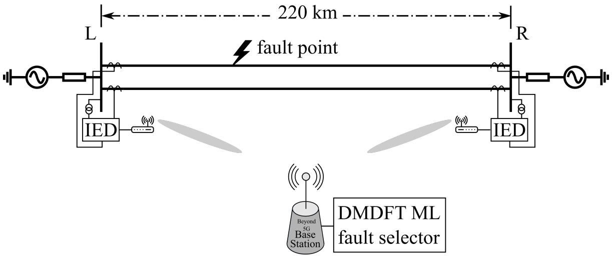


Figure 8: 400 kV double-circuit transmission line

The training and testing datasets were collected in the preprocessing phase. All the simulations were carried out in the MATLAB/Simulink environment; these simulations were prepared with the specifications shown in Table 5. Both normal operations and different fault types (10 in total) were simulated along with different fault resistances (24), fault inception angles (2), line parameter errors (5), high and low power flow (2), and fault locations along the line (9). The simulation comprise 20160 simulation rounds to collect statistically significant data. The resulting dataset is publicly available upon request to the corresponding authors, and will be shortly made available as open-source. The transmission line parameters of the line from which the one-end signal datasets were gathered are listed in Table 6.

Table 5: Simulation input parameters

Fault	**Training dataset**	**Testing dataset**
Type	None, AG, BG, CG, ABG, BCG, CAG, AB, BC, CA, ABC	None, AG, BG, CG, AB, ABG, BCG, CAG, BC, CA, ABC
Resistance (Ω)	0.01, 0.1, 1, 5 and 10–200 (steps of 10)	random
Distance (%)	10–90 (steps of 10)	random
Inception angles	2 (45° and 90°)	random
Power flow variation	2	random
Line parameter error	5	random
Total size	**15120**	**5040**

Table 6: Transmission line parameters

Parameter	**Transmission Line L-R**
Voltage (kV)	400
Length (km)	220
Positive-sequence resistance (Ω/km)	0.0033564
Positive-sequence inductance (H/km)	0.00057347
Positive-sequence capacitance (F/km)	2.0423e^{-8}
Zero-sequence resistance (Ω/km)	0.27073
Zero-sequence inductance (H/km)	0.0039052
Zero-sequence capacitance (F/km)	7.9939e^{-9}

References

[1] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction (Information Science and Statistics)*. Springer, 2007.

[2] M. de Castro Tomé, P. H. Nardelli, and H. Alves, "Long-range low-power wireless networks and sampling strategies in electricity metering," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1629–1637, 2019.

[3] A. Chiumento, N. Marchetti, and I. Macaluso, "Energy efficient wsn: a cross-layer graph signal processing solution to information redundancy," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, 2019, pp. 645–650.

[4] I. T. Christou, "Avoiding the hay for the needle in the stack: Online rule pruning in rare events detection," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2019, pp. 661–665.

[5] I. T. Christou, E. Amolochitis, and Z.-H. Tan, "A parallel/distributed algorithmic framework for mining all quantitative association rules," *arXiv preprint arXiv:1804.06764*, 2018.

[6] A. Chiumento, B. Reynders, Y. Murillo, and S. Pollin, "Building a connected ble mesh: A network inference study," in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, April 2018, pp. 296–301.

[7] A. F. Villaverde, J. Ross, F. Morán, and J. R. Banga, "Mider: Network inference with mutual information distance and entropy reduction," *PLOS ONE*, vol. 9, no. 5, pp. 1–15, 05 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0096732

[8] I. T. Christou, E. Amolochitis, and Z.-H. Tan, "A parallel/distributed algorithmic framework for mining all quantitative association rules," 2018.

[9] G. G. Towell and J. Shavlik, "Knowledge-based artificial neural networks," *Artificial Intelligence*, vol. 70, no. 1–2, pp. 119–165, 1994.

[10] J. J. Downs and E. F. Vogel, "A plan-wide industrial process control problem," *Computers and Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993.

[11] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567 – 1581, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959152412001503

[12] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault detection and diagnosis in industrial systems*. Springer, 2001.

[13] E. L. Russell, L. H. Chiang, and R. D. Braatz, "Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 51, no. 1, pp. 81–93, 2000.

[14] Y. Liu, M. Kashef, K. B. Lee, L. Benmohamed, and R. Candell, "Wireless network design for emerging iiot applications: Reference framework and use cases," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1166–1192, June 2019.

[15] W. Dai, H. Nishi, V. Vyatkin, V. Huang, Y. Shi, and X. Guan, "Industrial edge computing: Enabling embedded intelligence," *IEEE Industrial Electronics Magazine*, vol. 13, no. 4, pp. 48–56, 2019.