ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA030

USING AI IN THE FAULT MANAGEMENT PREDICTIVE MODEL OF THE SKA TM SERVICES: A PRELIMINARY STUDY

M. Canzari, M. Di Carlo, M. Dolci, INAF, Osservatorio Astronomico D'Abruzzo, Italy
R. Smareglia, INAF, Osservatorio Astronomico di Trieste, Italy

Abstract

SKA (Square Kilometer Array) is a project aimed to build a very large radio-telescope, composed by thousands of antennae and related support systems. The overall orchestration is performed by the Telescope Manager (TM), a suite of software applications. In order to ensure the proper and uninterrupted operation of TM, a local monitoring and control system is developed, called TM Services. Fault Management (FM)[1] is one of these services, and is composed by processes and infrastructure associated with detecting, diagnosing and fixing faults, and finally returning to normal operations.

The aim of the study, introducing artificial intelligence algorithms during the detection phase, is to build a predictive model, based on the history and statistics of the system, in order to perform trend analysis and failure prediction. Based on monitoring data and health status detected by the software system monitor and on log files gathered by the ELK (Elasticsearch, Logstash, and Kibana) server, the predictive model ensures that the system is operating within its normal operating parameters and takes corrective actions in case of failure.

INTRODUCTION

The Square Kilometre Array (SKA) Project is aimed to build a radio telescope that will enable breakthrough science and discoveries, that would be impossible with current facilities over the next 50 years. In the overall SKA architecture, two telescopes (SKA MID and SKA LOW) are composed each one by several Elements covering all required functionalities, e.g. DISH and LFAA (the front-end Elements for direct radiation detection), CSP and SDP (data processing and delivery), SAT, SaDT and INFRA (support functionalities). The global orchestration of this huge system is performed by a central element called Telescope Manager (TM), which has three core responsibilities: management of astronomical observations (proposal and scheduling), management of telescope hardware and software subsystems (observation execution) and management of telescope engineering data.

TM is a complex (and distributed) system, mostly composed by software packages (TMC, OSO, Services), web applications and user interfaces running on a hardware & virtualization software platform. In order to ensure the proper and uninterrupted operation of TM, Software System Monitor and Fault Management of TM Services have the role to detect, isolate and recover faults.

TM failure situations are derived from accurate dependability analysis (like FMECA, FTA, ...) which are performed onto the system as it is being developed and built. The results are used by the Software System Monitor to immediately detect failures and by Fault Manager to easily isolate and correct the faults which caused them. This process, however, does not ensure the prediction of all possible failures, nor the tracement of every failure to a specific fault or set of faults, especially for complex systems like SKA TM.

In case of occurrence of an unexpected failure, its detection can be difficult and take some time: a much longer time, however, can be expected for the process of fault isolation (usually performed by manually drilling down monitoring data and inspecting log messages) and recovery. This could result in a significant and unacceptable outage period for the system.

On the other hand, the sudden (i.e. not preceded by abnormal, non-critical events) occurrence of a unpredicted failure is rather rare in a complex system: a deep post-event analysis of monitoring data usually reveals hidden correlations and structures which could not be taken into account in an a priori estimate and can be discovered only once the system is operating.

For all these reasons the use of Artificial Intelligence in discovering as many cause-effect relations as possible in TM, as well as speeding up the process of failure detection and fault isolation and recovery in TM, seems the most suitable to maximize its availability (defined as its ability to mask or repair faults such that the cumulative service outage period is within the required range).

In this paper a preliminary study of this approach is presented. The existing algorithms are reviewed and the selection of the most suitable ones for our application is discussed. Finally, the tools used for initial testing on a trial dataset are shown (Fig. 1).

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA030

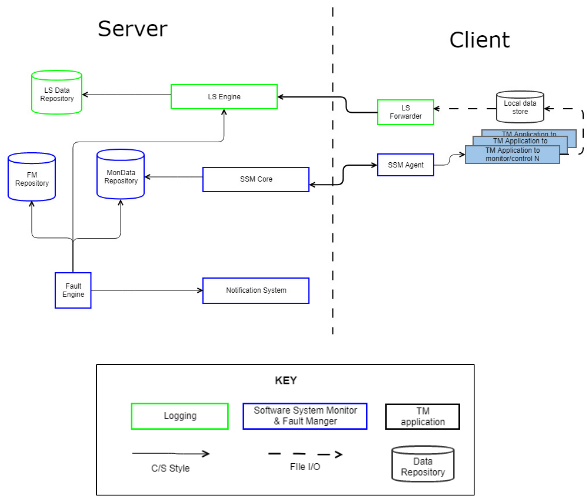

Figure 1: the architecture of SSM and Fault Manager. AI Algorithms are implemented in the Fault Engine Module.

ARTIFICIAL INTELLIGENCE AND PREDICTIVE MODEL

A Predictive Model consists of the construction of a model based on data source useful to make predictions. It is primarily needed to prevent future events, but it can be applied also to past unknown events, regardless when they have occurred. There are a lot of methodologies that allow to create predictive models. The current state-of-the-art of technologies, together with the increase of data volume and the processing power, makes it possible to apply Artificial Intelligence algorithms and machine learning to create powerful predictive models that learn certain properties from a training dataset in order to be able to make predictions. Several Artificial Intelligence algorithms are available from literature, every one created for a specific purpose, it is also possible to use combinations of algorithms on the same dataset or through the application of a certain algorithm to the output of another computation. The choice of the algorithm, or combination of algorithms, most suitable for the specific application provides a better convergence to the solution and to a minimization of time and computational steps. In the next map the most used Artificial Intelligence algorithms in literature are reported (Fig. 2).

METHOD STUDIED

The technique of machine learning applied to the predictive model can be divided in two different areas [2]: Regression and Pattern Classification. The Regression consist in the study of the relations between a dependent variable and one or more independent variables (or predictors). At variance with it, the Pattern Classification is focused on the recognition of patterns and regularities in data and the assignment of discrete class labels to particular observations.

Supervised and Unsupervised Machine Learning Algorithms

In our study the attention has been focused on the Pattern Classification: in particular, it can be further grouped in two subcategories: supervised and unsupervised. In the supervised learning, the class labels in the dataset, which are used to build the classification model, are knows. More generally, in a given set of input variables (X) and one output variable (Y), a supervised algorithm learns the mapping function from the input to the output

$$Y = f(X)$$

Once the function of mapping from the learning dataset, the aim of the algorithm is to compute values of (Y) from new given values of (X). It is called supervised learning because the process is supervised by a "teacher" (a human operator), who knows the real situation and can confirm, correct or reject the predictions the algorithm makes on the basis of what it has learnt. Learning stops when the algorithm achieves an acceptable level of performance. Most common algorithms of supervised learning are:

- Linear regression for regression problems.
- Random forest for classification and regression problems.
- Support vector machines for classification problems.

On the contrary, unsupervised learning deals with unlabelled instances, and the classes have to be inferred from the unstructured dataset. More generally, it consists in knowing a set of input variables (X) without knowing output variables. The aim of unsupervised learning is modelling the structure or the data distribution in order to know them more. This learning is called unsupervised because there is no verification process by a "teacher": algorithms are left to their own devises to discover and present the interesting structure in the data. In particular, unsupervised learning can be further divided in:

- Clustering: used to discover groups of data
- Association: allows to identify rules that associate big portions of data

Most common algorithms are:

- k-means for clustering problems.
- Apriori algorithm for association rule learning problems. column).

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA030

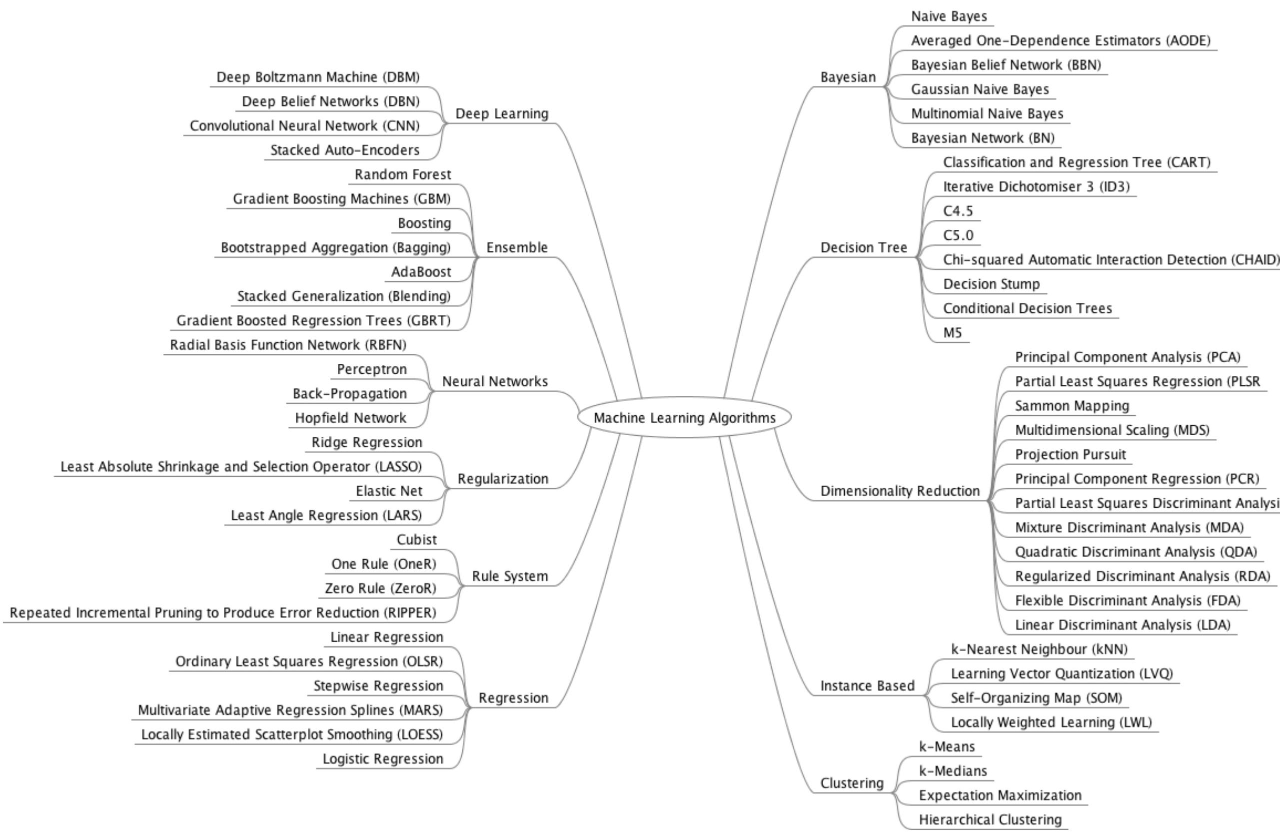

Figure 2: A mind-map of artificial intelligence algorithms.

CASE STUDY

According to the study carried out, two different fields have been identified where the use of machine learning helps to improve the availability of the system. The first one is the validation of FMECA. For the validation of FMECA supervised learning algorithms, like Random forest for classification and regression problems, have been identified. The second one is the fault prediction of the system. In fact, FMECA analysis has the purpose of analyzing effects and the severity of failures on the whole system. So, by defining (X) as the possible inputs of the system that can cause failures and (Y) as the response of the system to the failure, FMECA identifies with a study the failure mode f(X), as the behaviour of the system when (X) occur. Studied and established failure mode in the FMECA, it is possible to test f(X) using Random forest for classification and regression problems algorithm. to verify if the values of the (Y) obtained do not differ from (Y) estimated.

In order to perform fault prediction, unsupervised algorithms have been preferred. This because all the f(X) already known and already studied in FMECA have been validated with supervised methods. Nevertheless, it is possible that FMECA does not cover all the possible cases in which the system can evolve. So, it is necessary to predict anomalous behaviours knowing only the values of (X). Among the most used tools the machine learning module of X-Pack (formerly Prelert) of Kibana [3][4] has been taken into consideration. This unsupervised learning patented software uses algorithms of Clustering based on Bayesian network. During the preliminary testing, has been simulated the functioning of a Tango Device Server which produced log based on his functioning and an error in the device server has been simulated. Previously there was a script that detected the error of the device according to parameters that would have been obtained in FMECA stage (Fig. 3).

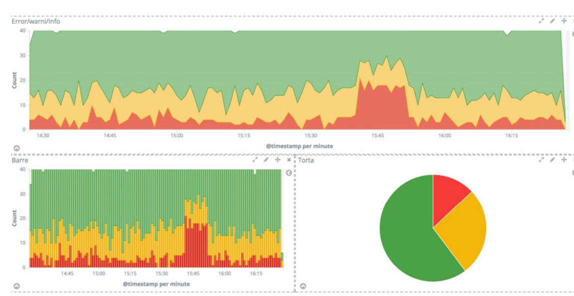

Figure 3: The dataset and the simulation of fault.

An alert was sent upon the occurrence of this condition. In order to test the algorithm the detecting script has been disabled. For the learning stage the system has been working for 6 days, logging about 80.000 documents of logs. Once the learning stage was over, in Tango Device Service has been simulated an error (in this case unexpected in FMECA) that has modified the functioning of the system. Machine learning algorithm, according to the learning already made, detected that the system was not working properly and reported the error (Fig. 4).

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA030

Figure 4: The AI algorithm detects the anomaly.

In order to evaluate the performances and the effectiveness of algorithms the following parameters have been individuated:

- Sensitivity: It measures the correctness of the predicted model. It is defined as the ratio of classes correctly predicted to be fault prone
- Specificity: It also measures the correctness of the predicted model. It is defined as the ratio of classes predicted that will not be fault prone.
- Accuracy: It is defined as the ratio of number of classes (including faulty and non- faulty) that are predicted correctly to the total number of classes file.

PRELIMINARY CONCLUSIONS AND FUTURE WORK

In this paper a test has been performed on Artificial Intelligence algorhytms with the aim to automatically detect a failure or a fault in a system where these conditions had not been predicted by a preliminary FMECA. The test used a limited simulate dataset but successfully proved the goodness of the adopted approach.

The results, even being very preliminary (the used dataset, made of a limited amount of simulated data, is not sufficient to describe the functioning of a complex system), are very encouraging and open the way to a more extensive work.

Next steps will use much larger datasets (at last one year of continuous operation) related to real telescopes, which will be processed by applying all the methods identified in this paper.

REFERENCES

[1] M. Di Carlo *et al.*, “Tm Services: An Architecture for Monitoring and Con-Trolling the Square Kilometre Array (Ska) Telescope Manager (Tm)”, presented at ICALEPCS’17, Barcelona, Spain, Oct. 2017, paper TUPHA207, this conference.

[2] Python Machine Learning, Sebastian Raschka.

[3] T.J. Veasey and S.J. Dodson, “Anomaly Detection in Application Performance Monitoring Data”, *International Journal of Machine Learning and Computing*, Vol. 4, No. 2, April 2014.

[4] “System and method for visualisation of behaviour within computer infrastructure”, Patent EP 2645257 A2.