Abstract—Today, the improvement of the product value in consumer goods, such as new services to increase the positive customer experience, is the subject of many research activities. In a context where the product complexity becomes ever greater and the product life-cycle is always shorter, the use of intelligent tools for supporting all phases of the product life-cycle is very important. One of the aspects that is taking interest is to support the consumer in fault management. This analysis are well-known practices in the industrial, automotive fields, etc. but less used for consumer electronics. This paper analyzes a Cloud service based on a Machine Learning (ML) approach used to provide fault detection capabilities to household appliances equipped with electric motors and compare the results with on premise ML algorithms provided research tools. The purpose of this paper is to perform a preliminary comparison of ML algorithm performances provided by two software, namely Microsoft Azure (cloud solution) and MATLAB (on premise solution), on a study case. In detail, the vibration data of an asynchronous motor installed in an oven extractor hood for commercial restaurant kitchen have been analyzed. To this end, two classification algorithms have been selected to implement fault diagnosis techniques.

Index Terms—Artificial Intelligence, Machine Learning as a Service, Fault Diagnosis.

I. INTRODUCTION

Nowadays, a lot of research is focused on improving energy efficiency and product value in consumer goods [1]. In this context, the research is focused on the definition of new services for enhancing the positive experience of the customers with their products [2]. Smart homes and smart appliances are an example of possible applications. They are based on the use of Internet of Things (IoT), which is an important technology for enabling smart appliances. In fact, IoT can create a direct channel for services from manufacturers to consumers for enabling services such as the replacement of parts, maintenance, energy savings, or fault prediction [3]. Nowadays, many appliances are sold directly equipped with network cards for providing an Internet connection; however, not all of them are connected to internet [4]. Perhaps, current services do not encourage the connection of some appliances such as professional wash machines, extractor hoods, fridges, coffee machines, etc. Fault detection and diagnosis are well-known practices in the industrial, automotive, avionic fields, etc. but less used for consumer electronics [5]. Fault detection algorithms are important in a context where the product complexity and the shorter product life-cycle require intelligent tools for supporting all phases of the product life-cycle.

This paper proposes an IoT service to provide fault detection capabilities to household appliances equipped with electric motors. The proposed service is based on a Machine Learning (ML) approach to detect or predict a fault by a vibration frequency analysis. Even if similar IoT services for fault detection and diagnosis are already studied in the literature [6], no attention has been paid to the comparison between the performances of cloud-based ML services with on premise ML algorithms provided academic/research tools (such as MATLAB). Generally, Cloud-computing services for ML concern the possibility to use external resources and services which can be deployed on the web. Therefore, this method is also called as “ML as a service” (MLaaS). Without installing any software, the developer user can apply cloud-computing services for data visualization, recognition, processing, predictive analysis and deep learning. As an example, Google provides Cloud AutoML which is a commercial service that allows users to develop a machine learning model using Google's environment [11]. Lee et al. studied the workflow for using autonomic machine learning tools based on minimizing expert intervention [7]. Even if ML tools are easy to use with cloud-computing services, a comparison between cloud ML services and on premise ML algorithms has not yet been evaluated in the literature. The purpose of this paper is to perform a preliminary comparison of ML algorithm performances provided by two well known software, namely Microsoft Azure (cloud solution) and MATLAB (on premise solution), on a consumer study case. In particular, the vibration data of an asynchronous motor installed in an oven extractor hood for commercial restaurant kitchen have been analyzed. To this end two-class classification and multiclass classification algorithms have been selected to implement fault diagnosis techniques. The paper is organized as follows. The description of cloud MLaaS Azure Machine Learning service has been presented in section II. Section III details the case of study and the comparison results between the Azure ML algorithms and the MATLAB solutions. Remarks and future works conclude the paper.
II. MICROSOFT AZURE MACHINE LEARNING

A. Description

Microsoft Azure Machine Learning encompasses cloud services that, through the Microsoft datacenters global network, permits the creation and deployment of applications by developers. This cloud computing model realizes a cloud platform which is completely flexible, agile and scalable. This means that the infrastructure can be used by developers in order to quickly create, test, debug, retrain, and deploy ML model without purchasing or provisioning an expensive hardware. In the past, Machine Learning tasks were performed using different type of programming languages or frameworks, like Python, Java, R, Hadoop, etc.. This platform mixes all the statistical and programming concepts together using highly recognized industry standard algorithms. The programming interface is based on visual paradigm and it applies the drag and drop functionality in order to connect together pre-programmed modules realizing a data analysis pipeline, which includes data pre-processing, algorithm selection, exploration, validating modeling results, etc.. Actually, in the Azure ML service the main analytics algorithms proposed by Microsoft are: classification, regression, and clustering [12]. In detail, Azure ML addresses more than 100 techniques such as regression, anomaly detection, binary and multiclass classification, text analysis, etc, however it allows as well the use of python and R languages for user models customization [13]. The basic process of creating Azure Machine Learning solutions is composed of a repeatable pattern of workflow steps that are summarized in Figure 1.

B. Machine Learning Algorithms

In Azure ML there are different machine learning language algorithms available such as Two-Class Classification, Multiclass Classification, Neural network, etc., but the focus is placed on the algorithms which are comparable with their counterparts in Matlab. In detail, the Logistic Regression and the Decision Tree algorithms have been chosen for Two-Class Classification, while the Decision Forest and the Decision Jungle algorithms have been tested for the Multiclass Classification.

- Logistic regression is a statistical linear algorithm used to model the probability of the occurrence of a certain class or event and, generally, it is used as a prediction model for simple problems. It predicts values by applying statistical analysis [14].
- Decision Trees are one of the most popular algorithms used for classification and regression problems in machine learning. They are based on sequential models, which logically combine together a sequence of tests and have the advantage over black-box models, such as neural networks, in terms of comprehensibility [15].
- The Decision Forest algorithm is a learning method consisting of an ensemble of decision trees, which means that it can construct decision trees each with a different classification. The decision forest selects the decision tree with the most votes [16].
- The Decision Jungle is a recent extension of the Decision Forest which is composed by rooted decision directed acyclic graphs (DAGs). Unlike standard decision trees that only generates one path to every node, in a decision jungle a DAG permits multiple paths from the root to each leaf [17].

III. CASE OF STUDY

This section shows the methodology for developing the diagnostic model starting from a real dataset acquired by different sensors placed on an asynchronous motor used in an oven extractor hood. The methodology is divided into the following steps:

A. Definition of Objectives
B. Bench Test
C. Data Preparation
D. Implementation of the Diagnostic Model
E. Preliminary Comparison between Microsoft Machine Learning Services and MATLAB

A. Definition of Objectives

The goal of the developed model in Azure ML is to provide a specific solution to the problem of diagnosing faults on asynchronous motor rotating bearings based on the statistical vibrations analysis. In summary, to recognize the presence of bearing faults based on statistical parameters, calculated starting from acceleration measurements. In addition to this, the developed model has to recognize, from one hand, the presence faults by using a binary classification, and on the other hand, to determine the location of the fault (Inner Race, Outer Race & Balls) using a multiclass classification.

B. Bench Test and Results

The case of study is composed by a double suction box centrifugal fan, with belt transmission driven by an asynchronous motor with rated output power of 1.46 Hp (1.1 kW), one of the classical solutions used for extractor hoods in professional kitchens. The motor speed is directly managed by
the four-speed selector of the hood electronic control board. To simulate different work load, a magnetic brake unit has been installed on the belt transmission. Three accelerometers has been used to acquire vibration. One accelerometer has been placed on the drive end, the second one on the belt transmission end of the motor housing, and the third one on the right-hand side surface along the axial direction. The speeds were increased from 500rpm up to 1250rpm in four steps. At each speed, four loads were applied to the belt transmission at the percentages of 0%, 33%, 66%, and 100% of the maximum load. The sampling rate was 12kHz and 48kHz while the slot time was 30s. At the end the data were sampled, using a 16 channel DAT recorder with 24-bit resolution A/D.

C. Data Preparation

Initially, the dataset containing the readings acquired by the accelerometers were structured in a single file and therefore, in order to use them, a first revision has been made, using the MatLab tools, adding a new column containing, for each reading, the type of associated fault (No Fault: 0, Inner Race Fault: 1, Bearing Ball Fault: 2 and Outer Race Fault: 3). Labels have also been added to identify the columns. After labelling operation, a second dataset elaboration has been made, using two Scripts written in R language. Through a sampling operation, the 24000 samples have been divided into about 2000 groups from about 120 readings each. The following statistical parameters have been evaluated for each of these groups:

- Standard Deviation. It is an index of statistical dispersion, which indicates how much data is placed around the average value.
- Range: A statistical parameter which indicates the difference between the maximum and the minimum value of a population.
- Kurtosis is a statistical index that indicates the departure from the normal distribution. It describes a measure of the “thickness” of the density function tails, or the degree of “flattening” of a distribution.
- Deviance. It is used in the Variance and Standard Deviation calculation and is defined as the sum of the squares of residuals.

Each group is replaced in the dataset by a single record containing the values previously calculated plus the error indicator. For the binary classification model, the datasets containing the occurrences classified as “Normal Behaviour (0)” and “Inner Race Fault (1)” has been considered. For multiclass classification, on the other hand, the data set is made up of the union of all four datasets. The data pre-process model is shown in Figure 3.

D. Implementation of the Diagnostic Model

After importing the dataset, a “Remove Duplicate Rows” filtering module has been developed in order to eliminate all duplicates from the dataset and reduce the workload during the training phase, as well as to avoid the problems of Overfitting and overtraining. This allows to provide to the following module, the “Split Data Module”, the optimal dataset to separate training and testing data, reserving 80% of the dataset for the training phase. After preparing the datasets, each algorithm is connected in input to the “Train Model Module”, which requires the training dataset at the input (see Figure 4). In this module, the Target Feature is selected, which is the feature used as subject of the diagnosis, in this case the “Fault”. The Train Model generate a “Trained Model”, which is tested by passing it to the Score Model. The Score Model, starting from the input test data, produces a dataset called Scored Dataset containing the forecasts made by the “Trained Model” (see Figure 5). Through the “Evaluate Model” module, a comparisons has been obtained between the different algorithms in terms of precision and accuracy (see Figure 6).
E. Preliminary Comparison between Microsoft Machine Learning Services and MATLAB

The evaluation metrics used in this experiment, in order to provide a preliminary quantitative assessment of the performances of the Azure ML and the similar algorithms in MATLAB, are the accuracy and the confusion matrix.

1) Two-Class Classification: In this case, the comparison has been made on the same algorithms, due to the fact that, for the binary classification, both Azure and Matlab provide the same algorithms.

The chosen algorithms are Boosted Decision Tree and Logistic Regression. In Table I is shown that both Azure ML and Matlab provide algorithms with excellent accuracy. However, there is no a perfect correspondence between the algorithms. Performing further tests, it has been noticed how the simple algorithm based on Decision Trees (Coarse Trees), in MATLAB, gives an accuracy of 99.8% like Logistic Regression.

2) MultiClass Classification: In this case, as there is no correspondence between the algorithms present in the two environments, it has been chosen to show which are the ones that give the best results in both cases. As for Azure, they are:

- Multiclass Decision Jungle: 73.7%
- Multiclass Decision Forest: 72.3%

while in Matlab:

- Fine Tree: 71.5%
- Boosted Decision Tree: 71.6%

Both programming environments guarantee good accuracy, over 70% even if the most accurate is Azure ML Studio. Very important in the Multiclass Classification case is the comparison of Confusion Matrix. In particular, Figure 7 shows the Confusion Matrix of Azure ML’s Multiclass Decision Jungle algorithms and Matlab’s Boosted Decision Tree algorithms.

IV. CONCLUSION AND FUTURE WORKS

In this paper a preliminary comparison of Microsoft Azure ML services with on premise ML algorithms provided by MATLAB is proposed. In particular, an asynchronous motor installed in an oven extractor hood for commercial restaurant kitchens has been analyzed, and acceleration data provided to several ML algorithms to perform fault diagnosis. to perform a ML method through vibration data and to execute Accuracy comparison between the two ML environment. MATLAB is a powerful mathematical tool which permits to perform complex calculations and data operations, but Azure ML service is useful for creating Machine Learning models in a simple and quick way and have a good accuracy. Azure ML, thanks to its Cloud nature, tends to increase its speed over on premise solutions over time, and also allows not only to manage larger datasets, but to operate on them quickly and easily,
thanks to the programming language R. Moreover, thanks to
the block programming interface, the creation of diagnostic
and predictive models is facilitated. A decisive element that
makes Azure the first choice for commercial applications is
the possibility of publishing the model created through Web
Services in a few simple steps.

REFERENCES
“Enhancing products and services using smart appliance networks,”
data analytics for smart homes with fog and cloud computing,” Future
approach for service innovation of smart product-service systems,”
for the conceptual design of energy-efficient cooker hoods,” Applied
Test and Diagnosis, pp. 91138, 1994.
using IoT device and cloud environment,” Journal of Manufacturing
machine learning platform,” International Journal of Information
2015.
for decision making in the era of Big Data evolution, challenges and
“Speeding up distributed machine learning using codes,” 2016 IEEE
[12] J. Barnes, “Azure machine learning: Microsoft azure essentials,” Mi-
using machine learning techniques for stock markets,” 2015 IEEE
Advances in Computing, Communications and Informatics (ICACCI),
pp. 834-840.
decision tree classification and logistic regression”, Journal of Expert
using decision tree forest on SAR image data”, ISPRS Journal of
“Decision jungles: Compact and rich models for classification”, Journal