

Fault Diagnosis in the Field of Additive Manufacturing (3D Printing) Using Bayesian Networks

<https://doi.org/10.3991/ijoe.v15i03.9375>

Abdelkabir Bacha ^(✉), Ahmed Haroun Sabry, Jamal Benhra
Hassan II University, Casablanca, Morocco
abdelkabir.bacha@ensem.ac.ma

Abstract—In this work, a new approach for fault diagnosis in the field of additive manufacturing (3d printing) using artificial intelligence will be given. This approach is based on the marriage of the Bayesian Networks theory and data acquisition techniques. Bayesian Networks are well known for their ability to infer probabilities and to give decisional support under uncertainty. In order to do so, these probability engines must be constructed and maintained by a big amount of data and information using learning algorithms. This work provides a methodology that uses sensors based data acquisition and processing to construct such networks. Some of these sensors are already available in most of the 3d printers available in the market, while other sensors were additionally embedded in a studied 3d printer in order to enrich the number of observational variables to gain a high level of fault diagnosis accuracy and support.

Keywords—Fault diagnosis; 3D printing; additive manufacturing; Bayesian Networks; data acquisition; sensor

1 Introduction

The increasing complexity and pervasive uncertainty of large-scale industrial systems have dramatically increased their vulnerability to failures. For some critical systems such as nuclear power plants, space vehicles, and chemical plants, although considerable efforts have been made to design better equipment and control systems, fault and failure management remains largely dependent on human operators. These humans are "supposed to" respond accurately to the location of the origins of the failures and take prompt and appropriate corrective action in an emergency. However, decision-makers face a variety of uncertainty issues that can arise from measurement bias, process noise, transmission loss, unmeasured exogenous influence, and so on.

To solve these problems, automatic fault diagnosis is helpful in helping technicians to detect, isolate and troubleshoot problems.

For this, attention must be given to accurately capture and model the changing causalities of the process in question. The representation and reasoning of uncertain, time-varying and sophisticated dependencies are difficult, especially for cases of complex and dynamic negative feedback.

This paper's work will focus firstly on the in-depth study of the state of the art concerning the diagnosis of failures of industrial machines. Several works on industrial diagnostics will be presented and cited. Then a critical analysis of each of these works will be given to finally lead to an opening towards the work that resolves our work.

The main idea of the paper is: how to use the already existent sensors in a 3d printer (such as thermistors, voltage sensors, A/D converters, stepper driver signals, etc.) and implement some new others (such as Hall Effect current sensors, voltage, temperature, etc.) in order to generate some datasets that will be useful to elaborate Bayesian networks using learning algorithms.

2 Industrial Systems Fault Diagnosis and Theoretical Materials

2.1 Fault diagnosis and monitoring in industrial systems

The race for companies and industries to optimize the gain/loss ratio as well as the growing demand for high-quality, specification-compliant products and/or services have contributed immensely to the evolution and imperative use of automated systems monitoring methods.

In literature, a lot of works were focused on fault identification, detection, and diagnosis using either model and data-driven approaches. For data-driven and AI approaches one can cite, for example, [1] who used SVM for fault diagnosis in the robotic field. Another studies such as [2], [3] concerned fault Diagnosis for Numerical Control Machine. These machines are very similar in functionality to 3D printing machines.

The study [4] gives a very detailed literature review on the application of BN in fault diagnosis.

2.2 Bayesian networks

Bayesian Networks (BN) are directed acyclic graphs that represent probabilities of a set of random variables. They have been initiated by J. Pearl et al [5], [6] and other researchers [7]–[9]. BN have other denomination such as Belief Networks [10] or Probabilistic graphical Model [11]. BN are based on probability theory, especially the well-known Bayes theorem and the joint probability (equation (1)).

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \pi_i) \quad (1)$$

BN can be constructed with different methods, first, they were elaborated manually by modeling the human knowledge, and then many algorithms were conceived especially for automating this task using data [12], [13].

BN parameter learning: BN parameter learning is technically trying to estimate probability distributions (or parameters of the relevant laws) from available data. The

most used methods for Bayesian networks learning, according to the kind of data in possession, and to their completeness (no values are missing the records) are : statistical learning, Bayesian learning for complete data, while learning with incomplete data uses specific algorithms such as the Expectation Maximization (EM) algorithm BN[12].

BN structure learning: Learning a BN structure is, in other words, trying to find the structure that better represents the studied problem. So learning the structure of a Bayesian Network from data means finding a graph representing a perfect map (called a P-map) of an independence model associated with a probability distribution that one has a sample from. So one must be sure of the existence of such P-map (loyalty) and the knowledge of all the variables (causal sufficiency) [8], [12], [14].

Learning the structure of a BN with exact or exhaustive methods is practically impossible, especially when the number of nodes is important [15]. The equation (2) demonstrates that the number of possible structures for a BN with “n” nodes is super-exponential. For $n = 5$, $NS(5) = 29281$; for $n=10$, $NS(10) = 4.2 \times 10^{18}$ [8].

$$NS(n) = \begin{cases} 1 & , \quad n = 0 \text{ or } n = 1 \\ \sum_{i=1}^n (1)^{i+1} \binom{n}{i} 2^{i(n-i)} NS(i-1) & , n > 1 \end{cases} \quad (2)$$

From this, one can imagine how huge can become the space of solutions. So, in order to overcome this problem, many heuristics were proposed. Some of them try to reduce the space of solutions to a space of trees. Some other methods try to be limited in the research of possible parents for each node (like the K2 algorithm[8], [10], [13]). While some other methods perform a greedy search in the space of the solutions.

2.3 Additive manufacturing and 3D printing

The Construction of 3D items through direct deposition of materials using 3D printing machines is entitled additive manufacturing (AM). 3D printing has many Paybacks including producing on-demand products rapidly, with customization possibility and less waste. While the usage of AM technology has been increasing, numerous challenges continue to block its more and more adoption.

3 Bayesian Network’s Structure Learning using Data Acquisition from a 3D Printer

3.1 The studied machine

The used 3D printing machine is called "Prusa I3 rework". This machine is part of "open-source" machines that can be built freely under the General Public License (GPL). Reproduction freedom here means both the software part that is downloadable in open-source software repositories such as "Github"[16]. This freedom also means

the material part with the availability of all the construction details. We have built this machine inspired by these free plans, for the purpose of conducting a study of and developing a diagnostic system capable of identifying the failures of this type of machines.

3.2 Hardware description

The Hardware of a 3D printer is composed of the following elements:

- Three axes (x, y, and z) of orthonormal motion with Cartesian coordinates.
- Stepper motors: NEMA 17 1.8 ° in steps, 200 steps per revolution. These are the main organs of the transformation of electrical signals into mechanical movements.
- An Extruder: Made of metal with a 0.3mm diameter nozzle equipped with a heating resistor and a thermistor to measure temperatures.
- A Heat Bed, which is used to heat the printing surface to reach 100 ° C for printing ABS.
- Another board of the 3D printer: Called RAMPS V1.4 (RepRap Arduino Mega Pololu Shield).
- A Powerbox: switching power supply (switch mode power supply) that converts 220V AC to 12V continuously with a maximum load of 30 Amps.



Fig. 1. A picture showing the experimental setup of the studied machine

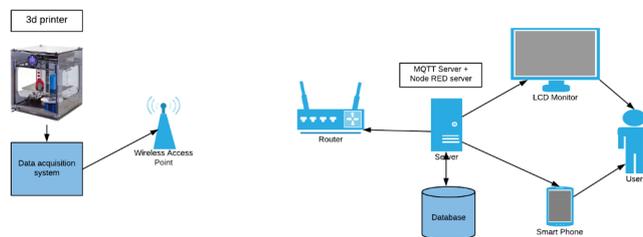


Fig. 2. 3D printer fault diagnosis synoptic diagram

In most nowadays open-source 3D printers, the main software that drives these types of machines is based on two sections. The first section must be installed in a PC and the other section is programmed into the embedded system (the main microcontroller) of the printer.

The first section contains a collection of software that must be used to work with a 3D printer. To start this journey, one must use CAD software (Computer Aided Design) with the design of the 3D model and by generating an STL file (quoting STL). To finish, one ends up by using CAM (CAM) tools to generate a list of instructions that the machine interprets with ease, this list is called: G-CODE.

The second section: Called the firmware of the printer, this part is a sophisticated program that resides in the main microcontroller which is, in this case, an "Arduino Mega" [17], [18]. This program interprets the G-code of the 3D model to be printed and converts it into machine movements. By this, the desired object is constructed. Our approach is applied. We use in our case a firmware called "Marlin V 1.4"[19].

3.3 The fault diagnosis system used for the 3D printing machine

The data acquisition system for this studied machine consists of two main elements. The first element is an electronic card that contains several sub-blocks. Each sub-block fulfills a specific function that we will detail later. The second element is a microcontroller card which is responsible for processing the data and converting it from analog to digital mode and transmitting it over radio waves (IEEE 802.11n standard) to a computer containing the learning algorithms we want to use to generate the targeted Bayesian network.

The data acquisition interface

Hardware description of the Acquisition Interface: The data acquisition interface is shown in the figures 3 and 4. This interface is based on a microcontroller that handles many tasks such as receiving data from sensors, analog/digital conversion and wireless data transfer via WIFI to the unit that will handle this data. The microcontroller has a clock frequency evaluated at 16MHz. This will allow it to perform these tasks very quickly, in addition to faster data transfer. The type of data acquisition architecture used in this context is different from that previously used in the numerical control machine, which states a "push-based" architecture [20]. The architecture used here is of the type: "Publish and subscribe" [21] deployed in a communication protocol called MQTT [22]. The data is exchanged in a Wi-Fi support (802.11n [23]) with a server deployed in an embedded system running the Linux operating system. In the latter server, an MQTT server called broker is executed to handle the transfer of messages. Then in the same embedded system, one can find another server called "Node RED server" [24] which manages the interfacing with the end user. This user can access the data of our system using several channels, especially via HTTP protocol. That is to say via a web interface whether by a desktop computer connected to the same networks or by a simple mobile phone having the possibility of executing a web browser. This is possible using the majority of nowadays smartphones. It should be noted that these solutions become the standard in the field of IOT applications "Internet of things" or connected objects.

A synoptic diagram is given in figure3. This synoptic diagram summarizes the different exchanges of data that are generated in this approach. Starting from the raw data extracted from the 3D printer, then, finishing with those data specific to the user of the fault diagnostic system. In fact, if the user decides to act on the functioning of the printer (Stop printing after detection of an anomaly for example) he can push some button from his user interface phone or PC.

This interface is constituted, as can be seen in images, of some elements called sub-blocks. The first block is the one that contains an integrated circuit at the top right of figure 4. This integrated circuit, whose reference is HEF4051 [25], performs the function of multiplexing a plurality of analog inputs to make it possible to combine the reception of the data of several sensors in a single analog input in order to meet a limitation concern in terms of the number of analog input lines in the main microcontroller.

After we have at the center of the card, a group of 6 resistors which are mounted as voltage divider bridges to allow the reading of some voltages that are higher than the voltage range allowed by the microcontroller. This voltage division concerns the reading of the supply voltage which is of the order of 12 V. It does concern also the reading of the voltage at the terminals of the heating plate of the 3D printer and finally, the reading of the voltage across the filament heater nozzle.

Right next to the resistance bridge we find the temperature acquisition block for DS18B20 digital sensors [26]. Then at the bottom right we have a real-time clock (RTC) module that will allow an accurate measurement of the time for the chronological recording of the sensor readings. At the top of this sensor, we have the Wi-Fi module that allows wireless communication with a remote computer.

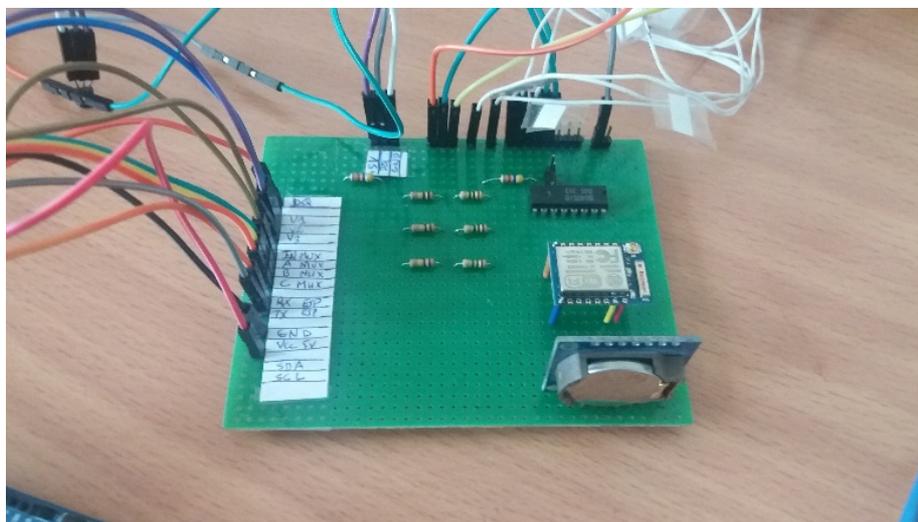


Fig. 3. A picture that shows the used data acquisition card for the FDM machine

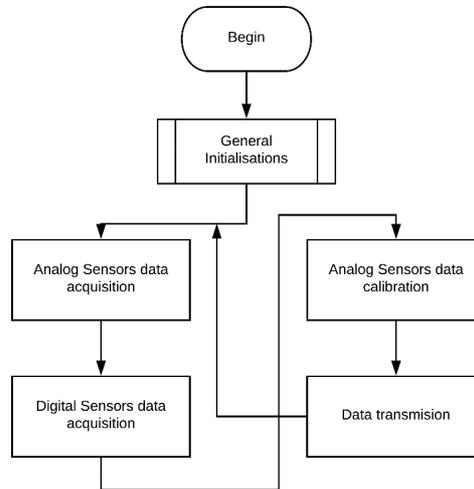


Fig. 4. Flowchart of the data acquisition system

Software description of the Acquisition Interface: Figure 4 gives the flowchart of the management program for the data acquisition system that was deployed for the studied 3D printer. That is to say the software part responsible for the acquisition of data from the sensors of the diagnostic system described above, in the preceding paragraphs.

The frequency of measurements: The Only variables that represent high-frequency variation concern the directions of the stepper motors and their numbers of 'steps'. These ones represent the only kind of variables that cannot be measured by this system. The monitoring of these variables is not interesting in our study because it will depend on the part to be manufactured and the printing speed chosen by the user of the printer. So the chosen value for sampling all the variables that interest us is 50ms as a time period.

The variables of the printer’s fault diagnosis system: The variables that were used in this work are represented in both tables 1 and 2. In fact, some variables are measured directly from the machine. They can be either continuous or Boolean. There is also another type of variables generated inside the computer that executes the learning algorithms. Table 2 groups the variables representing faulty situations. These latter variables are added manually to perform a supervised learning by adding decisions to each instance recorded from the measured variables.

The preprocessing: This step concerns the cleaning and filtering of the used dataset. In fact, an unsupervised discretization method based on the works of Cooper [13] et al and Friedman et al [27] was used in order to make possible the learning of the BN structure with continuous data collected from analog sensors. This step was executed in the machine learning Weka environment [28]. As for Boolean attributes, the same algorithm was applied but this time with only 2 Bins and a Make Binary option.

Table 1. Description of the variables generated from the 3D printer fault diagnosis system.

Reference	Variable	Description	Type
1	Volt X	X axis stepper motor driver Voltage	Continuous
2	Curr X	X axis stepper motor driver Current	Continuous
3	Tem X	X axis stepper motor driver Temperature	Continuous
4	Lim X	X axis limit switch status	binary
5	Volt Y	Y axis stepper motor driver Voltage	Continuous
6	Curr Y	Y axis stepper motor driver Current	Continuous
7	Tem Y	Y axis stepper motor driver Temperature	Continuous
8	Lim Y	Y axis limit switch status	binary
9	Volt Z	Z axis stepper motor driver Voltage	Continuous
10	Curr Z	Z axis stepper motor driver Current	Continuous
11	Tem Z	Z axis stepper motor driver Temperature	Continuous
12	Lim Z	Z axis limit switch status	binary
13	Volt Ext	The voltage of the Extruder	Continuous
14	Curr Ext	The current of the Extruder	Continuous
15	Tem Ext	The temperature of the Extruder	Continuous
16	Volt Heatbed	The voltage of the Heat-Bed	Continuous
17	Curr Heatbed	The current of the Heat-Bed	Continuous
18	Tem Heatbed	The temperature of the Heat-Bed	Continuous

Learning the structure and parameters of the model: To learn the structure of the BN, The K2 algorithm offered by [15] was used after confronting several learning algorithms (K2, Simulated annealing, Hill climber) [29] because it has given the best results in the evaluation compared to his competitors. This algorithm works only with nominal and no missing values. It can be initiated in a way that will force the generation of a Naive Bayes structure. Also, a Markov Blanket correction can be activated to the learned classifier. This guarantees that all variables in the network are part of the Markov blanket of the main classifier node. Finally, an additional important parameter of this algorithm concerns the number of parents that each node can have. Conditional probability tables (CPT) were learned using “simple CPT estimator” algorithm from the Weka ML software [28]

Table 2. List of added variables representing faulty situations

Reference	Variable	Description	Type
F1	X Over Curr	X axis stepper motor driver Overcurrent	binary
F2	Y Over Curr	Y axis stepper motor driver Overcurrent	binary
F3	Z Over Curr	Z axis stepper motor driver Overcurrent	binary
F4	E Over Curr	Extruder axis stepper motor driver Overcurrent	binary
F5	X Short Cir	X axis stepper motor driver Short circuit	binary
F6	Y Short Cir	Y axis stepper motor driver Short circuit	binary
F7	Z Short Cir	Z axis stepper motor driver Short circuit	binary
F8	E Short Cir	Extruder axis stepper motor driver Short circuit	binary
F9	X Lim Fault	X axis Limit switch toggle fault	binary
F10	Y Lim Fault	Y axis Limit switch toggle fault	binary
F11	Z Lim Fault	Z axis Limit switch toggle fault	binary
F12	X Ovrht Fault	X axis stepper motor driver Over heating Fault	binary
F13	Y Ovrht Fault	Y axis stepper motor driver Over heating Fault	binary
F14	Z Ovrht Fault	Z axis stepper motor driver Over heating Fault	binary
F15	E Ovrht Fault	Extruder axis stepper motor driver Over heating Fault	binary
F16	HB Ovrht Fault	Heat bed Over heating Fault	binary
F17	Fault	General Faulty situation in the machine	binary

Learning a Naive Bayes structure using theK2algorithm: TheK2 algorithm was started with the option to learn a Naive Bayes. A Naive Bayes network is a network that hasan edge connection from the classifier node to every other node. If this option is not used, an empty network will be used instead as an initial network structure. The maximum number of parents that each node can have was set to 1 (P=1). Figure 6 shows the structure of the learned network.

Learning the structure of a BN using theK2 algorithm with max number of parents greater than 5: In the next step, a modification of some options of the algorithm is recommended, in order to find the configuration that gives the optimum in score functions. So, the parameter called “Max Nb of Parent” that labels the maximum number of parents that each node in the graph can have was increased. Figure 7, shows the learned BN after finding the right number of parents which is in this case the number 3. So, increasing the said parameter for values greater than 3will always give the same graph.

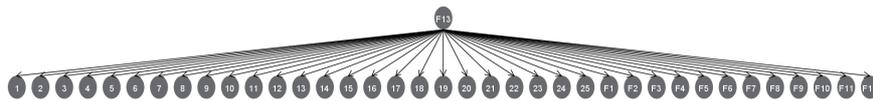


Fig. 5. A Naive Bayes network structure learned from the data of the 3D printer (P=1).

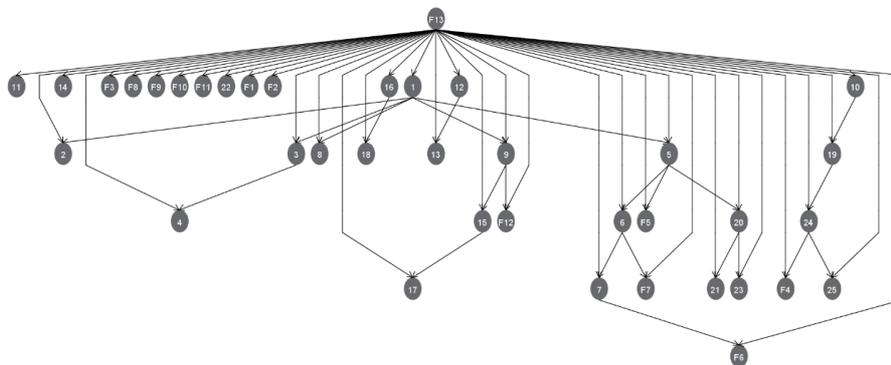


Fig. 6. Bayesian Network structure learned from the data of the 3D printer (P>2)

4 Results and Discussion

The two BN graphs that were generated and presented in figures 6 and 7 were used in classification tasks in order for the authors to evaluate their performance. In this perspective, a 10-fold cross-validation[30] test was run. This method gives more accuracy in classifier performance tests. The results of this validation method are presented in tables 3, 4 and 5.

In comparison between the two graphs, one can observe that increasing the parameter P=“MaxNbofParent” have a positive effect in term of classification performance of the network. But after exceeding experimentally the number 3, the graph structure converges and stays the same.

Table 3. Summary of the 10-fold cross-validation

Metrics	P=1	P>3
Correctly Classified Instances	92.5556 %	98.1848 %
Incorrectly Classified Instances	7.4444 %	1.8152 %
Kappa statistic	0.5902	0.8424
Mean absolute error	0.0712	0.0209
Root mean squared error	0.2323	0.1271
Relative absolute error	55.3921 %	16.2227 %
Root relative squared error	91.684 %	50.1774 %
Total Number of Instances	4903	4903

Table 4. Detailed accuracy

for P=1									
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class Fault
	0.908	0.073	0.479	0.908	0.627	0.628	0.986	0.886	0
	0.927	0.092	0.993	0.927	0.959	0.628	0.986	0.999	1
Weighted Avg.	0.926	0.0909	0.957	0.926	0.936	0.628	0.986	0.991	
for P>3									
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class Fault
	0.757	0.002	0.973	0.757	0.852	0.85	0.994	0.941	0
	0.998	0.243	0.982	0.998	0.99	0.85	0.994	1	1
Weighted Avg.	0.982	0.226	0.982	0.982	0.981	0.85	0.994	0.996	

In table 4, we give detailed classification metrics such as:

For P=1:

TP rate: true positive rate, which is 75.7% for “0” state of Fault variable, while it is 99.8% for the state of “1” for the “Fault” variable. For the average of both values, we have the ratio of 98.2%. False Positive rate: of “Fault” = “0” state is equal to 0.2% in contrast to the TP rate. False Positive rate: of “Fault” = “1” state is equal to 24.3%. On average we have 22.6% for both cases.

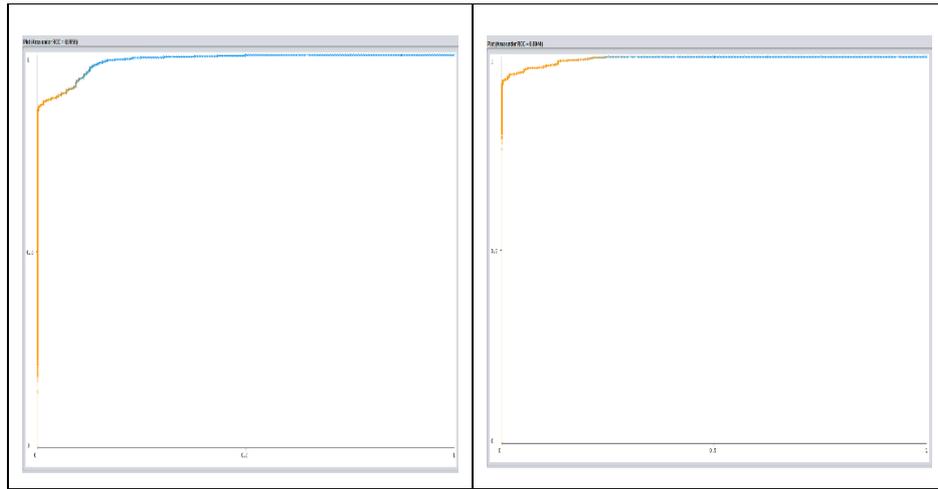
For P>3:

TP rate: true positive rate, which is 90.8% for “0” state of Fault variable), while it is 92.7% for the state of “1” for the “Fault” variable. For the average of both values, we have the ratio of 92.6%. False Positive rate: of “Fault” = “0” state is equal to 7.3% in contrast to the TP rate. False Positive rate: of “Fault” = “1” state is equal to 9.2%. On average we have 9.09% in both cases.

The same analysis can be applied to all other metrics of the table.

Table 5. Confusion matrix

for P=1			for P>3		
<i>a</i>	<i>b</i>	<i><-- classified as</i>	<i>a</i>	<i>b</i>	<i><-- classified as</i>
307	31	a = 0	256	82	a = 0
334	4231	b = 1	7	4558	b = 1



ROC curve for P=1 (Area under curve = 0,986)

ROC curve for P>1 (Area under curve = 0,994)

Fig. 7. ROC curves for the two graphs

In table 5, for P=1, in the first column and the first line of the matrix, 307 instances (faults) were given the “0” state. While in the first column but second line 31 of them were not given the “0” state by the Naïve Bayes classifier while they should do. This represents a mistake in the classification.

In the case of the second column first line, the instances that are present there constitutes an incorrect situation. In fact, they are classified as “0” state while they should have been classified as “1” state. We have “31” instances in that situation.

The fourth case (second column second line) contains “4231” instances that were in the correct situation.

In figure 7, Receiver Operating Characteristic (ROC) curves were given for both of the learned graphs. For both cases, this metric is giving good results because the areas under the curves are higher than 90% of the overall area of the curve. Plus, the ROC curve for P>3 (The learned BN) is better than the P=1 (Naïve Bayes graph).

5 Conclusion

In this paper, a methodology for fault diagnosis was proposed in the field of additive manufacturing and 3D printing using a probabilistic approach. This approach is based on learning Bayesian networks from data streamed in real time from an acquisition system that was specially designed and experimentally tested for this purpose by

the authors. In this perspective, the effectiveness of Bayesian networks in the diagnosis and detection of failures in industrial systems was presented. Then, a detailed description was given to all of the composing elements of both the studied 3D printer and the data acquisition system. After applying preprocessing algorithms such as discretization, missing values removal, supervising fault instances, etc... the Data generated from this experimental setup was used to learn the structure and conditional probability tables for two final Bayesian networks: A Naïve Bayes graph structure and a normal Bayesian Network. After the learning phase, a classification using 10-fold cross-validation method was run in order to test the performance of these learned BN graphs. The second one outperforms the Naïve Bayes structure in the majority of the used performance metrics.

6 References

- [1] X. Yuan, M. Song, F. Zhou, Y. Wang, and Z. Chen, "A Novel Fast Training Method for SVM and Its Application in Fault Diagnosis of Service Robot," *Int. J. Online Eng.*, vol. 11, no. 6, p. 4, 2015. <https://doi.org/10.3991/ijoe.v11i6.4846>
- [2] A. On and U. For, "Fault Diagnosis Based on Multi-sensor Data Fusion for Numerical Control Machine," vol. 12, no. 2, pp. 29–34, 2016.
- [3] A. Bacha, J. Benhra, and A. H. Sabry, "A CNC Machine Fault Diagnosis Methodology based on Bayesian Networks and Data Acquisition," *Commun Appl Electron*, vol. 5, pp. 41–48, 2016. <https://doi.org/10.5120/cae2016652353>
- [4] B. Cai, L. Huang, and M. Xie, "Bayesian Networks in Fault Diagnosis," *IEEE Trans. Ind. Informatics*, vol. 13, no. 5, pp. 2227–2240, 2017. <https://doi.org/10.1109/TII.2017.2695583>
- [5] J. Pearl, "Evidential reasoning using stochastic simulation of causal models," *Artif. Intell.*, vol. 32, no. 2, pp. 245–257, 1987. [https://doi.org/10.1016/0004-3702\(87\)90012-9](https://doi.org/10.1016/0004-3702(87)90012-9)
- [6] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.
- [7] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, no. 2–3, pp. 131–163, 1997. <https://doi.org/10.1023/A:1007465528199>
- [8] P. Leray, "Réseaux bayésiens: apprentissage et modélisation de systèmes complexes," Université de Rouen, 2006.
- [9] R. E. Neapolitan and others, *Learning bayesian networks*. Pearson Prentice Hall Upper Saddle River, 2004.
- [10] G. F. Cooper and E. Herskovits, "A Bayesian method for constructing Bayesian belief networks from databases," in *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, 1991, pp. 86–94. <https://doi.org/10.1016/B978-1-55860-203-8.50015-2>
- [11] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [12] O. François and P. Leray, "Etude comparative d'algorithmes d'apprentissage de structure dans les réseaux bayésiens," *Rencontres des Jeunes Cherch. en IA*, 2003.
- [13] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Mach. Learn.*, vol. 9, no. 4, pp. 309–347, 1992. <https://doi.org/10.1007/BF00994110>
- [14] D. Margaritis, "Learning Bayesian network model structure from data," US Army, 2003.

- [15] G. F. Cooper, “The computational complexity of probabilistic inference using Bayesian belief networks,” *Artif. Intell.*, vol. 42, no. 2–3, pp. 393–405, 1990. [https://doi.org/10.1016/0004-3702\(90\)90060-D](https://doi.org/10.1016/0004-3702(90)90060-D)
- [16] L. L. C. GITHUB, “GITHUB Project.” GITHUB, LLC.
- [17] Y. A. Badamasi, “The working principle of an Arduino,” in *Electronics, Computer and Computation (ICECCO), 2014 11th International Conference on*, 2014, pp. 1–4. <https://doi.org/10.1109/ICECCO.2014.6997578>
- [18] V. M. Cvjetkovic and M. Matijevic, “Overview of architectures with arduino boards as building blocks for data acquisition and control systems,” *Int. J. Online Eng.*, vol. 12, no. 7, pp. 10–17, 2016. <https://doi.org/10.3991/ijoe.v12i07.5818>
- [19] B. B. Patel and Z. J. Khan, “3-Axial CNC Router to 3D Printer.”
- [20] P. Jahnke, “Machine Learning Approaches for Failure Type Detection and Predictive Maintenance,” tu-darmstadt, 2015.
- [21] N. Fotiou, D. Trossen, and G. C. Polyzos, “Illustrating a publish-subscribe internet architecture,” *Telecommun. Syst.*, vol. 51, no. 4, pp. 233–245, 2012. <https://doi.org/10.1007/s11235-011-9432-5>
- [22] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, “MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks,” in *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*, 2008, pp. 791–798. <https://doi.org/10.1109/COMSWA.2008.4554519>
- [23] G. Holland, N. Vaidya, and P. Bahl, “A rate-adaptive MAC protocol for multi-hop wireless networks,” in *Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 236–251. <https://doi.org/10.1145/381677.381700>
- [24] M. Blackstock and R. Lea, “Toward a distributed data flow platform for the web of things (distributed node-red),” in *Proceedings of the 5th International Workshop on Web of Things*, 2014, pp. 34–39. <https://doi.org/10.1145/2684432.2684439>
- [25] I. N. C. Nexperia, “HEF4051B 8-channel analog multiplexer/demultiplexer.” Nexperia, INC, 2016.
- [26] I. Maxim Integrated Products, “DS18B20 Programmable Resolution 1-Wire Digital Thermometer.” Maxim Integrated Products, Inc, 2015.
- [27] N. Friedman, M. Goldszmidt, and others, “Discretizing continuous attributes while learning Bayesian networks,” in *ICML*, 1996, pp. 157–165.
- [28] G. Holmes, A. Donkin, and I. H. Witten, “Weka: A machine learning workbench,” in *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, 1994, pp. 357–361. <https://doi.org/10.1109/ANZIIS.1994.396988>
- [29] R. E. Neapolitan and others, *Réseaux bayésiens*. Editions Eyrolles, 2004.
- [30] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

7 Authors

Abdelkabir Bacha, Ph.D. in OSIL Team. He is associated with LRI Laboratory, ENSEM, Electricity and Mechanics National High School, Hassan II University, Casablanca, Morocco.

Ahmed Haroun Sabry, Ph.D. in OSIL Team. He is associated with LRI Laboratory, ENSEM, Electricity and Mechanics National High School, Hassan II University, Casablanca, Morocco

Jamal Benhra is a Professor and Head of the OSIL Team. He is associated with LRI Laboratory, ENSEM - Electricity and Mechanics National High School, Hassan II University - Casablanca, Morocco

Article submitted 10 August 2018. Resubmitted 19 September 2018. Final acceptance 14 October 2018. Final version published as submitted by the authors.