Soft Sensors: Software Development Applied to Aerospace Engineering Problems

Marcelo, Filho*. Gilberto, Reynoso-Meza*

** Industrial and Systems Engineering Graduate Program (PPGEPS), Pontificia Universidade Católica do Paraná (PUCPR), Curitiba (PR) 80215-901, Brazil; marcelo.feliciano@pucpr.edu.br (M.F.F.); g.reynosomeza@pucpr.br (G.R.-M.)*

Abstract: This paper aims present methods for implementing a software embedded Soft Sensor (SS) into real engineering problem, which was defined by AIRBUS in the International Federation of Automatic Control (IFAC) World Congress 2020 as a benchmark in aerospace engineering. Such software uses an embedded Soft Sensor to process flight simulation data from Simulink® by using Machine learning methods in Python language to classify Oscillatory Failure Errors. Then, a Systematic Literature Review presented basic idea about Machine Learning, a state of art about Soft Sensors and its key questions to guide the research. Three different machine learning representations have been implemented: Support Vector Machines (SVM), Decision Trees (DT) and Multi-Layer Perceptron (MLP). In such methods, the best was Decision Tree (DT) with 51.56% precision average in four scenarios.

Resumo: Este artigo objetiva o apresentar métodos de implementação de sensores virtuais (*Soft Sensors - SS*) em formato de software aplicado a um problema real de engenharia apresentado pela AIRBUS ao congresso mundial da IFAC (*International Federation of Automatic Control*) como benchmark de engenharia aeroespacial. Tal software utiliza uma simulação de SS para processar dados do Simulink® por meio de métodos de aprendizado de máquina para classificação de falhas de caráter oscilatório. Dessa maneira, a revisão sistemática da literatura apresentou ideias básicas sobre aprendizado de máquina, o estado de arte acerca dos SS e as questões chave que guiaram essa pesquisa. Ademais, o desenvolvimento do software e resolução do estudo de caso foi realizado por meio da aplicação de três métodos: máquina de suporte vetorizado, árvores decisórias e o Perceptron multicamadas. Dentre esses métodos, o que melhor performou foi o das árvores decisórias com 51.56% de precisão média dentro dos quatro cenários estipulados.

Keywords: Soft Sensors (SS); Machine Learning (ML); Python Software Development; Decision Trees (DT); Support Vector Machine (SVM).

Palavras-chave: Sensores Virtuais; Aprendizado de Máquina; Desenvolvimento de Software em Python; Árvores Decisórias; Máquina de Suporte Vetorizado.

1. INTRODUCTION

Soft Sensors (SS) are software-embedded methods with machine learning methods being applied to many features, such as: decision making, complex variable estimation, regression, prediction, and maintenance. Such applications are required for industry 4.0 scenario into smart factories, that requires Machine Learning (ML) methods to solve problems.

This paper will explore aerospace engineering problems solutions with SS by developing a software to emulate flying process and apply ML methods to identify Oscillatory Failure Cases (OFCs). The benchmark has been proposed by AIRBUS to IFAC, which has published a note named as "*Aerospace Industrial Benchmark on Fault Detection*". Such problem is related to flight control; however, SS are possible to apply in many areas at fourth industrial revolution by its versatility and benefits.

The main objective of this paper is to present methods for implementing a software embedded Soft Sensor to process flight simulation data from Simulink® by using ML methods in Python language. This software must have a User Interface (UI), communicate with MATLAB® through Application Programming Interface (API) and this Soft Sensor based software developed methods must attend to IFAC´s basic requirements.

Summarizing this paper, section 2 presents a Systematic Literature Review (SLR), section 3 shows benchmark´s specifications, requirements and models, section 4 is about results (software development and benchmarks solution), section 5 opens a discussion about presented results and section 6 concludes this paper by proposing future works and SS contribution to Industry 4.0 ML appliance.

2. SYSTEMATIC LITERATURE REVIEW (SLR)

In this section, the reader´s contextualization about ML concepts will be in 2.1, then Soft Sensors´ state of art and description as modern solution to industry 4.0 problems in 2.2. Afterwards, all defined questions will be solutioned to guide the software requirements development in 2.2 and in subsection 2.3 the technologies applied will be explain.

2.1 Machine Learning (ML) Concepts

According to Orzechowski et al. (2018), ML enables data transformation from large amounts databases into concise information eligible to human understanding by applying methods or algorithms. Such methods are divided into five main tribes, as describes Domingos (2015): Symbolists, Bayesians, Evolutionaries, Analogizers and Connectionists.

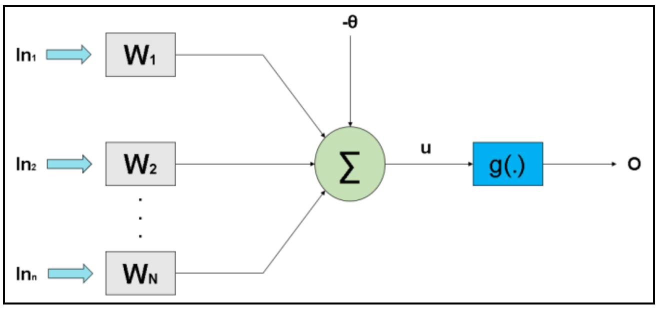
Symbolists uses logics to create models as decision trees (for example) and by backpropagation techniques can solve complex problems. By the other hand, Domingos describes Bayesians as a probabilistic tribe, so only logics are not capable to explain probabilistic events in graphical models, this tribe main problem is to deal with variance and covariance from samples. The author points this method appliance as recommendation system on Xbox Store™ solution example.

Then Evolutionaries creates models based on evolution theory, in which each generation on epoch will perform a grade on cost function, and best set of generations will create next epoch´s generations, until cost function constrains be solved or error function tends to minimum possible.

Meanwhile, Analogizers are focused on constant optimization and clustering by similarity, creating ‘families’ of problems, its main algorithm is Support Vector Machine (SVM). According to Kumar et al. (2020), this method uses statistical learning to cluster classes by sample´s data, such process demands hyperparameters and cost function optimization to reach performance.

Connectionists tribe aims to solve problems such as biological brains do, by using Artificial Neural Networks (ANN), in which synaptic connections are done multiplying weights and these are defined while training network. According to Domingos (2015), it processes inputs (In) and multiply it by weights (W) to sum it multiplied by normalization coefficient (θ) which results on activation potential value (u), function g(.) will output the result (O). Figure 1 presents a simple ANN:

Fig. 1. Simple Artifical Neural Network model.



According to L. Ma et al. (2019), to solve complex models its necessary to add more layers to this ANN, creating the Deep Learning (DL) network to process data from many different inputs (sensors). This real time processing makes intelligent decision making possible and to reach such results advanced computational resources are often required.

2.2 Soft Sensors: State of Art

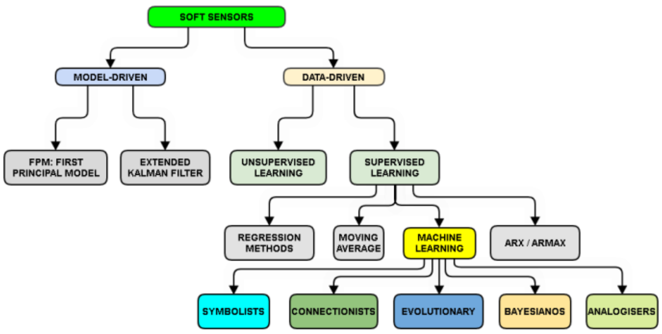
According to Souza, Araújo and Mendes (2016), Soft Sensors (SS) are defined as inference tools that process sensor data by ML methods to measure complex variables. Such ML methods are applied to information quality improvement by removing biases, outliers and creating models to data understanding.

Going into word´s etymology, the researchers above defined SS as junction between ‘software’ and ‘sensors’, in which data is collected by sensors and interpreted by software. As occurs in supervisory systems, however SS can apply ML methods to heal human decisions or predict model´s behaviour.

SS can be also divided into two groups: model driven SS and data driven SS, as explains Maggipinto (2019). According to author, the first group is based on real-time analysis, allowing short-time predictions and fast decision making by managers or technicians. By the other hand, the second group is created by robust database analysis and models are generated by ML techniques into reliable predictions based on past experiences.

For Kadlec e Gabrys (2009), the SS hierarch groups by ML techniques and methodologies can be explain by figure 2:

Fig. 2. Soft Sensors techniques schema.



This schema shows main techniques into regression and ML tribes, showing many branches on data-driven SS pat and two ways for model-driven SS, which are FRM (First Principal Model) and Kalman Filter. Such filters can estimate processes dynamics in closed loop control by analyzing discrete data from sensors considering noise disturb, according to Welch and Bishop (2006).

Into data-driven SS family branches, Zambonin et al. (2019) defines this approach as statistical techniques employed to transform low-cost sensor´s data into variables hard or expensive to measure. Another linear method mentioned by Kadlec e Gabrys (2009), is ARX (AutoRegressive with eXogenous inputs) and ARMAX (AutoRegressive moving-average Model with eXogenous inputs) Besides that, Wo Jae Lee et al. (2019) denominate such tool to work with real time data, as timeseries to provide decision making into smart factories scenario, from predictive maintenance to business strategy.

2.3 SLR Key Questions Summary

While searching, performing a Systematic Literature Review Hannah Snyder (2019) it´s define main questions to synthetise the research´s findings and creating goals. In present paper, the main questions are shown by Table 1 and they were primary important to objectives definition.

Table 1. Key Questions Summary.

KEY RSL QUESTIONS
Q.01 – What are the industry problems that SS can solve?
Q.02 – Which are the benefits brough by its implantation?
Q.03 – Which techniques are employed to SS development?
Q.04 – Which ML methods are employed to SS?

To respond first question, about problems solved by SS, Negash et al. (2016) estimated petroleum reservoir´s volume using model-driven SS, instead destroying expensive sensors to get precise data, low-cost sensors were deployed. Another application has been developed by Jalee and Aparna (2016), used data-driven SS in petroleum fractionated distillation process to predict final product´s composition by analysing sensors data, instead submitting samples to lab´s analysis. Smart Products (SP) can be improved by SS, as Zambonin et al. (2019) applied it to wash machines, which can be configured by user and read (with sensors) clothes data, such as it´s weight and humidity to choose the best combination for water flow and cleaning products. SP appliances goes further on Shaoming et al. (2020) paper, who explored a literature review about low-cost SS to self-driving cars, it´s benefits and limitations.

Second question was answered by Souza, Araújo and Mendes, (2016) research about main benefit they considered data processing and synthesizing in real time to decision making and easily data-driven indicators development. Negash et al. (2016) and Jalee and Aparna (2016) have pointed the cost reduction as main advantage, meanwhile Maggipinto et al. (2019) constated the main advantage on SP improvement and resources economy in clothes washing.

About main SS techniques (question three), Souza, Araújo and Mendes (2016) leads to a five steps path: data selection and collection (Principal Component Analysis – PCA and anti-aliases filters); input data selection (cost functions and main data after PCA); choose SS model (data-driven or model-driven SS) and it´s ML training method; fold-validation (train data and test data using minimum Squared Error – MSE) and model continuous maintenance with learned data.

The last question showed the main methods presented by RSL articles; Lee Wo et al. (2019) used Pareto optimal front – POF method to identify best machine cycle on metal surface roughness decrease. Then, Support Vector Machine (SVM) clustering method has been used by Lee Wo et al. (2019) to find the best cutting tools on machining process optimization. Concurrently, Zambonin et al. (2019) saw on Genetic Algorithm (GA) the solution to petrochemicals composition estimation model. Into image processing field, Lei Ma et al. (2019), uses Deep Learning (DL) in SS monitoring system to detect variations, classify objects or areas and increase image resolution. Meanwhile, Kumar et al. (2020) applies Random Forest (RF) and Decision Trees (DT) algorithms to train data-driven models.

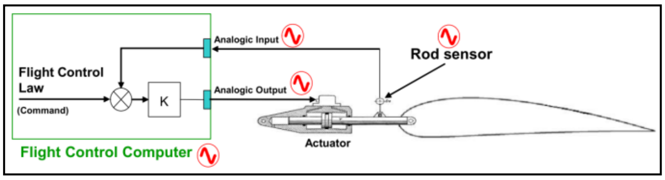
3. BENCHMARK PRESENTATION

The aerospace engineering problem has been presented by AIRBUS on IFAC (*International Federation of Automatic Control*) conference, at beginning of 2020 in South Africa. The main benchmark objective is to identify Oscillatory Failure Cases (OFC´s) at Flight Control Systems (FCS) used on commercial flights airships. This section structure will be presented on following sub sections: at 3.1 benchmark physical model will be explained, then 3.2 presents system requirements and 3.3 the Simulink model.

3.1 Benchmark´s Phisical model

Benchmark´s physical model is considering only OFCs located in the servo-loop control of the moving, it´s the actuator, Flight Control Computer (FCC) and the control surface, including the rod sensor, as shown on figure 3:

Fig. 3. AIRBUS benchmark mechanism.



3.2 Solution requirements

IFAC and AIRBUS have proposed some minimum requirements to solution, these are pointed at table 2:

Table 2. Benchmark´s Requirements.

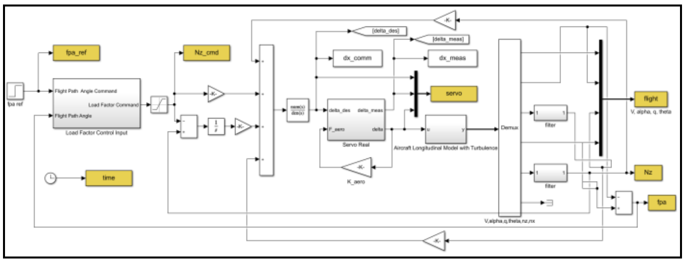
R.01 – Minimum amplitude OFCs must be detected.
R.02 – OFC´s signal frequency from 1 to 10hz must be detected by SS.
R.03 – OFC signals shall be detected within three periods of oscillation, in any OFC frequency.
R.04 – Liquid and Solid OFCs must be detected;
R.05 – OFC signals at the servo current input or at the rod position sensor shall be detected.
R.06 – There should be no false alarm on normal flight situation (any turbulence) and load factor step input, sine input, or chirp signals.

According to these requirements, the Simulink model was provided to researchers focus on attending it, and this article´s software proposition has been developed to solve it. By implementing a Soft Sensor, which can process real time data from flight control system and identify oscillatory failure cases and software´s contribution is allowing the user to setup simulation and extracting reports/charts to decision making and, further maintenance prediction.

3.3 MATLAB® And Simulink models

Benchmark´s Simulink model presents flight simulation process effects on aircraft and rod sensor (shown on fig. 3), figure 4 shows the complete Simulink model:

Fig. 4. AIRBUS benchmark Simulink model.



Simulink model above defines a closed-loop flight control system and can be divaded in four groups: flight path control, load factor control, OFC surface detection and dynamic airplane´s turbuence simulator. Then it´s possible to setup scenarios throughout some paramentrs. More details about benchmark´s simulation can be found on IFAC´s paper.

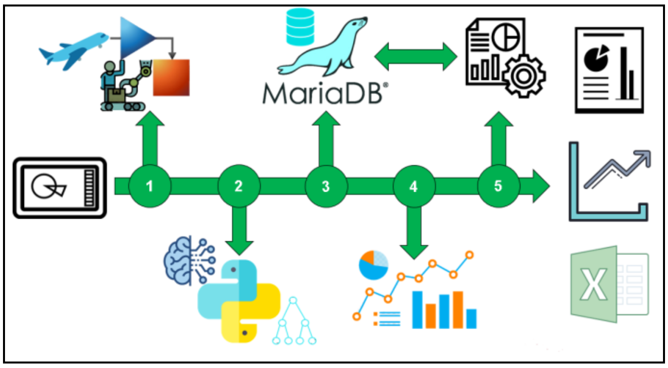
4. RESULTS

After developing the software, obtained results are separated into software interface/usability section 5.1 and benchmark´s solution using ML methods embed on software section 5.2.

4.1 Software Interface and Usability

Software working process can be explained by fig. 5 diagram:

Fig. 5. Software mental-map flowchart.



According to fig.5, the first step is the user to set parameters on interface, then once all data was set up, software opens MATLAB® API with python and starts training or using trained ML models (SVM, DT or Multi-Layer-Perceptron – MLP) according to user´s choice. Afterwards classified data is real-time input on MariaDB database local instance to keep data (for posterior system´s maintenance or improvement), then graphics are generated. User can, after simulation ends, export reports, from database or software, in many formats, PDF, Excel™ plan worksheet, txt or image (.jpg). Software´s simulation result is seen on fig 6.

Fig. 6. Software output example.

According to fig. 6, it´s possible to see emulated flight process (in blue), green line shows the control signal from FCC and red crosses are the OFCs identified by selected ML method. User can set some simulation parameters, by input fields, for example turbulency (none, light, moderate or severe), OFC´s source (sensor, current, cs or cs_sensor), OFC´s type (solid or liquid) and control type (FPA, NZ_step, NZ_chirp or NZ_sine). It´s possible to set more specific parameters, as frequency (from 0.1 to 25hz), OFC´s amplitude (from 0.01 to 10mm or mA) and sensor bias (from 0.01 to 1 times the amplitude).

4.2 Benchmark Solution

Aiming to reach benchmark´s requirements presented on table 2 and paper´s objective, after implementing Decision Tree (DT) method in Python language and creating software UI based on User Experience (UX), the tests have been separated in four scenarios: ideal, normal, severe and stormy. All scenarios are tested for DT method, Support Vector Machine and Multi-Layer-Perceptron, without cross-validation, all data being emulated and predicted in real-time. These situations parameters can be emulated on software and are shown on table 3:

Table 3. Benchmark´s Scenarios Parameters.

Parameters	**Scenario**			
	Ideal	**Normal**	**Severe**	**Stormy**
ML method	Decision Tree (DT)			
OFC´s Source	Sensor			
OFC Type	None	Liquid	Solid	Liquid
Turbulency	None	Light	Moderate	Severe
Control Type	FPA Control			
Amplitude (mm)	0.10	0.50	1.50	3.00
Bias (mm)	0.01	0.10	0.30	1.00
Frequency (rad/s)	3.0π	1.5π	0.5π	0.1π
Simulation Time (s)	10			
Sampling Time (ms)	25			

For each of these scenarios there will be a simulation and one Confusion Matrix (CM), all related with DT method, FPA control on OFC source as sensor and 10 seconds of simulation. For each CM, the Error Type I will be non-OFC classified as OFC (false positive) and Type II is real OFC not identified by SS. The ML method has been trained with real data obtained by AIRBUS in Python language using Pandas library to read csv data and SkLearn® to train with static hyperparameters, instead optimizing performance. Besides, sampling time is 25ms, then 400 registers are done in 10s.

In Ideal scenario, the software identified 40 OFC´s in low amplitude signal, only noise was present on simulated flight signal surface deflection which ranged in 0.168° from -0.08° to 0.084°. Then it´s result can be seen on CM presented by table 4:

Table 4. Confusion Matrix for Ideal Scenario.

Confusion Matrix		DT Model Prediction	
		No OFC	OFC
Simulink™ Data	No OFC	50.25%	4.75%
	Real OFC	39.75%	5.25%

Then CM trace gives the method precision for ideal scenario, which is 55.50% match with Simulink OFC identification, then false positive rate is 4.75% and false negative is around 39.75%.

By simulating the normal scenario, software identified 168 OFCs and flight surface deflection ranged in .120° from -0.1° and 0.020°. It´s result is shown by CM in table 5:

Table 5. Confusion Matrix for Normal Scenario.

Confusion Matrix		DT Model Prediction	
		No OFC	OFC
Simulink™ Data	No OFC	16.75%	14.75%
	Real OFC	41.25%	27.25%

The DT method in normal scenario performed with 44.00% accuracy and false positive rate is 14.75%, real OFC occurred in this scenario on simulation 68.50% cases.

Hence, severe scenario resulted in 225 OFCs, the surface deflection ranged in maximum deflection allowed by model, from -33° to 20° and control command from -11° to 22°, it´s due high bias and amplitude set to this situation. Then, table 5 brings the CM results after 10 seconds simulation:

Table 5. Confusion Matrix for Severe Scenario

Confusion Matrix		DT Model Prediction	
		No OFC	OFC
Simulink™ Data	No OFC	18.50%	25.25%
	Real OFC	17.25%	39.00%

The precision in this scenario is around 57.50% and false positive rate is 25.25%, and false negative rate goes to 17.25% due severe flight situations and method´s imprecision.

Finally, stormy scenario, which emulates the worst scenario possible, resulted in 257 OFCs identified by software, the surface deflection ranged less than previous scenario, by fluctuating from -0.68° to 0.61° and control command deviated from -2.15° to 0.25°. The result of this scenario can be seen from table 6:

Table 6. Confusion Matrix for Stormy Scenario.

Confusion Matrix		DT Model Prediction	
		No OFC	OFC
Simulink™ Data	No OFC	19.50%	16.25%
	Real OFC	34.00%	30.25%

According to this table, software precision in this scenario is 49.75% and false positive rate is 16.25%, real OFC occurred in this scenario on simulation 68.25% cases, false negative has occurred on 34.00% cases.

About benchmarks requirements, from table 2, the R.01 has been reach because on ideal scenario software identified minimum amplitude OFC (0.1mm). Such as R.02, because software identified OFC in stormy, R.04 on normal and severe scenarios have been identified. By the other hand, R.03 three periods of oscillation aren´t identified and R.06 (no false alarm) haven´t been reach. Table 7 shows the accomplished ones reached by software:

Table 7. Benchmark requirements reach.

Benchmark Requirement	Reach?
R.01 – Minimum amplitude OFCs must be detected.	Yes
R.02 – OFC´s signal frequency from 1 to 10hz must be detected by SS.	Yes
R.03 – OFC signals shall be detected within three periods of oscillation, in any OFC frequency.	No
R.04 – Liquid and Solid OFCs must be detected.	Yes
R.05 – OFC signals at the servo current input or at the rod position sensor shall be detected.	Yes
R.06 – There should be no false alarm on normal flight situation (any turbulence) and load factor step input, sine input, or chirp signals.	No

5. DISCUSSION

Starting by SLR is has been conducted by four main questions defined on table 1 and its responses have supplied the SS applications on industry worldwide, main problems solved by it, the development techniques to deployment and ML methods employed to make predictions, clustering, classifications and more over Soft Sensor´s data.

The software development based in UX resulted in friendly GUI (Graphical User Interface) by which user can set parameters on Simulink™ model to simulate the desired scenario. The communication occurs trough MATLAB® API which returns data to Python process, store on MariaDB™

database and then ML methods embed make predictions about data window to check if there is an OFC or not. These ML methods can be trained by user, before simulation, with specified datasets in .csv extension or uses by default Pickle stored datasets. Besides that, a graphical model of simulation is plot to user and it offers the possibility to export graph, report on txt format and database query in .xlsx Excel readable extension.

This integration is thought to data science methods on accessible software which can read .xlsx plans or inside excel. Although, MariaDB can be integrated to data science software in parallel to provide metrics about emulated process or data being acquired in real time. About benchmark simulation results, around DT method performance on flight simulation and OFCs identification, the summarized result for proposed scenarios is presented by table 8:

Table 8. DT Method Simulation Summarized Results.

Scenario	Control Signal Amplitude	Sensor Signal Amplitude	OFC Identified	Accuracy
Ideal	0.0075°	0.164°	10.00%	55.00%
Normal	0.12°	0.50°	33.50%	44.00%
Severe	33°	50°	56.25%	57.50%
Stormy	1.29°	2.40°	64.25%	49.75%

By analyzing table 8, it´s possible to check that as scenario goes to aggressive behavior the OFC identification percentage increases and accuracy deviates for each scenario keeping average in 51.56%.

On table 8, it´s possible to check the accomplished requirements over IFAC´s benchmark, R.03 wasn´t reach because sometimes three periods of oscillation could not be identified and R.06 because lack of false alarms, only 4.75%. It can be improved by working on engineering characteristics, however this paper didn´t explore such features.

6. CONCLUSION

In conclusion, SS are fundamental tools for applying ML on industry 4.0 smart factories scenario, which requires intelligent decision making guided by data analytics or ML methods. This paper main objective has been reached by SS software development that can be download on GitHub by cloning the repository: https://github.com/marcelo-feliciano-filho/TCC_MFF. Meanwhile, IFAC´s benchmark has been partially solved by it, however this paper´s main contribution isn´t only software development methods. It is to inspire future works to focus on data-driven maintenance and engineering characteristics to improve Decision Tree model hyperparameters or include more ML methods to software.

ACKNOWLEDGMENTS

This study was financed in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), the Fundação Araucária (FAPPR) - Brazil Finance Codes: 310079/2019-5-PQ2, 437105/2018-0-Univ, 51432/2018-PPP, PRONEX-042/2018.

REFERENCE

Domingos, P. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. (2015). 1 ed. Basic Books: New York, USA.

Hannah Snyder. Literature review as a research methodology: An overview and guidelines. Journal of Business Research. Oslo, V.104, n.01, 333-339 p.

IFAC2020. Aerospace Industrial Benchmark on Fault Detection. Avaiable at: <https://www.ifac2020.org/program/competition/aerospace-industrial-fault-detection>. Access in: 12 de may. 2020.

Jalee, A. E; Aparna, K. Neuro-Fuzzy Soft Sensor Estimator for Benzene Toluene Distillation Colum. (2016). Procedia Technology. n. 25. 92 – 99p.

Kadlec, P; Gabrys, B. Soft sensors: where are we and what are the current and future challenges. IFAC Proceedings Volumes (IFAC-PapersOnline). Bournemouth: UK. v.02, n.01, 2009, p. 572-577.

Lee, W; Mendis, G; Sutherland, J. Development of an Intelligent Tool Condition Monitoring System to Identify Manufacturing Tradeoffs and Optimal Machining Conditions. Procedia Manufacturing. (2019). West Lafayette: USA. v.33. n.16, 256-263p.

Ma, L; Liu, Y; Zhang, X; Ye, Y; Yin, G; Johnson, B. Deep learning in remote sensing applications: A meta-analysis and review. (2019). ISPRS Journal of Photogrammetry and Remote Sensing. Nanjing. v.152, n.03, p. 166-177.

M. W. Lee, J. Y. Joung, D. S. Lee, J. M. Park, S. H. Woo. Application of a Moving-Window-Adaptive Neural Network to the Modeling of a Full-Scale Anaerobic Filter Process. (2005). Industrial & Engineering Chemistry Research. v.44, n.11, 3973–3982p.

Maggipinto, Marco et al.; Laundry Fabric Classification in Vertical Axis Washing Machines Using Data-Driven Soft Sensors. (2019). Energies. Padova: Italy, v.12, n.21, p. 1-14.

Negash, Berihun et al. Conceptual Framework for Using System Identification in Reservoir Production Forecasting. (2016) Procedia Engineering. Seri Iskandar: Malaysia, v.148, n.01, p.878-886.

Orzechowski, P.; La Cava, W.; Moore, J.H. (2018). Where are we now? A large benchmark study of recent symbolic regression methods. *In Proceedings of the Genetic and Evolutionary Computation Conference*, Kyoto, Japan, ACM: New York: USA, p.1183–1190.

Shaoming, H; Shin, H-S; Xu, S; Tsourdos, A. Distributed estimation over a low-cost sensor network: A Review of state-of-the-art. (2020). Information Fusion. Cranfield: UK. v.53, n.11, p. 21-43.

Souza, Francisco A. A.; Araújo, Rui; Mendes, Jérôme. Review of Soft Sensors Methods for Regression Applications. (2016). Chemometrics and Intelligent Laboratory Systems. v.152, n.01, p.69-79.

Welch, G; Bishop, G. An Introduction to the Kalman Filter. In Pratice. (2006). Chapell Hill, North Carolina: USA. v.07, n.01,1-16p.

Zambonin, Guiliano et al. Machine Learning-Based Soft Sensors for the Estimation of Laundry Moisture Content in Household Dryer Appliances. Energies. Padova, Italy, v.12, n.20,2019, 1-24 p.