

IPC2008-64664

ARTIFICIAL NEURAL NETWORKS ENSEMBLE USED FOR PIPELINE LEAK DETECTION SYSTEMS

Inhaúma N. Ferraz

Universidade Federal Fluminense
Departamento de Ciência da Computação
Rua Passo da Pátria 156, Bl E, Sl 326, 24210-240
Niterói, RJ, Brazil

Ana C. B. Garcia

Universidade Federal Fluminense
Departamento de Ciência da Computação
Rua Passo da Pátria 156, Bl E, Sl 326, 24210-240
Niterói, RJ, Brazil

Flávia C. Bernardini

ADDLabs – Laboratório de Documentação Ativa e Design Inteligente
Universidade Federal Fluminense
Av. Gal. Milton Tavares de Souza, s/nº, 24210-340
Niterói, RJ, Brazil

ABSTRACT

The physical and operational properties of pipelines vary greatly. There is thus no universally applicable method, external or internal, which possesses all the features and the functionality required for a perfect leak detection performance. The authors of this paper know quite well that traditional methods, in a low uncertainty environment, overcome artificial intelligence methods of leak detection systems. If one considers the real world as a creator of uncertainties, neural networks and fuzzy systems emerge as important promising technologies for the development of leak detection systems. In this work, we propose a method for constructing ensembles of ANNs for pipeline leak detection. The results obtained in our experiments were satisfactory.

T Temperature.
P Pressure.
M Density or Mass.

NOMENCLATURE

CPM Computational Pipeline Monitoring.
LDS Leak Detection Systems.
AI Artificial Intelligence.
ANN Artificial Neural Networks.
MLP Multilayer Perceptron.
T Set of instances.
y Neurons output
w ANN connection weights.
x Instance.
F Flow rate.

INTRODUCTION

The detection methods of product leaks along a pipeline can be classified into direct and indirect (or inferential) methods. Direct methods usually detect external leaks outside the pipeline. External methods encompass physical inspections, acoustic emissions, fiber optic sensing, liquid sensing and vapor sensing. Indirect methods, called Computational Pipeline Monitoring (CPM), use parameters for inferring product leaks. These parameters, such as pressure, flow, temperature, etc. are obtained from instruments that are internal to the pipeline. Internal Leak Detection Systems (LDS) include Volume Balance, Pressure Analysis (Rarefaction Wave Monitoring) and Real Time Transient Modeling methods.

The LDS selection for pipelines relies on certain qualities contained in pipelines and products. However, the physical and operational properties of pipelines vary immensely. There is thus no universally applicable method, external or internal, which possesses all the features and the functionality required for a perfect leak detection performance. The authors of this paper know very well that traditional methods, in a low uncertainty environment, overcome LDS Artificial Intelligence (AI) methods. By considering the real world as an uncertainty

generator, we propose an alternate method for LDS using AI techniques. Artificial Neural Networks (ANN) and fuzzy systems are important emerging technologies for the development of leak detection systems [1][2][7]. The performance of these computational intelligence techniques is surprising in relation to the reduced detection time of the leaks, to the small dimension of localized leaks, and to the reduced rate of emitted false alarms. Since traditional methods present uncertainty operational difficulties, the development of research on the application of computational intelligence techniques in the elaboration of leak detection systems acquires considerable importance.

In this work, we propose a method for constructing ensembles of ANNs for pipeline leak detection, combining different specialists (ANN) of the domain, each one viewing the problem from a different angle. Due to the lack of real world data, some experiments were performed using simulated data. Some insights in real data about normal operation in pipeline are also described in experiments section. The results obtained were satisfactory.

This paper is organized as follows: Section 2 describes external methods in use for pipeline leak detection; Section 3 describes Artificial Neural Networks; Section 4 describes methods for constructing ensembles of classifiers; Section 5 describes our proposal for constructing ensembles of ANNs for pipeline leak detection; Section 6 briefly describes the computational system in development that implements our proposal; Section 7 describes our experimental results; and, finally, Section 8 concludes this work.

EXTERNAL METHODS FOR PIPELINE LEAK DETECTION

While pipelines are an efficient and economic means of transporting hazardous fluids over long distances, the risks associated with accidental releases are high. Leaks in pipelines carrying fluids such as oil, ammonia, gasoline, or chlorinated solvents can cause serious pollution, injuries and fatalities, if they are not promptly detected and repaired. For example, a leak in an NGL pipeline was responsible for the death of about 700 persons in Russia in 1988. For the period 1970-1984 in the United States alone, there were 46 serious accidents associated with natural gas pipelines leading to 86 casualties. Large leaks cause significant changes in pressure gradients and differences in mass flow rates at measurement points, and are therefore easy to detect. On the other hand, small leaks are more difficult to detect because changes in the standard measurement procedures are small. However, leaks as small as 1% of the nominal flow rate can cause the discharge of a large amount of dangerous fluid before they are detected, usually by the impact they have on the surrounding environment. The early detection of such small leaks is then the main goal of a leak-detection system. In what follows, some indirect methods for detecting leaks in pipelines are described [1].

Many methods for creating leak-detection systems in liquid and gas pipelines have been proposed, mainly based on process variables (pressure, flow rate, and temperature) usually measured in pipelines. Perhaps the most common is the line volume balance method [4], based on mass conservation of the fluid in the pipeline. Data for line volume balance come from flow meters. Usually a short-term and a long-term balance are calculated: the short-term balance provides fast response for large leaks, while it is claimed that the long-term balance will detect a 0.5% leak (of the nominal flow rate) in 3-6 hours. The volume balance can only identify a leak as being located somewhere in the line in which the fluid flow is measured. Since flow meters are usually installed at input and exit points and seldom anywhere in between, this technique usually does not give any information on leak locations along the length of a pipeline.

Acoustic methods have been proposed as well. In principle, they can detect very small leaks in a short span of time, but they do not always work well for large networks, where there may be background noise from compressors and valves [4]. Furthermore, spacing between detection stations must be of the order of 100 m, otherwise the reliability is low.

Finally, pressure waves generated by the leak provide another potential method of leak detection by measuring the pressure disturbances that travel along the line [6]. Signal conditioning is required in order to monitor the temperature and pressure variation in the pipeline (correcting the sound speed for any variation) and to account for process operations, eliminating the pressure disturbances deriving from normal processes.

ARTIFICIAL NEURAL NETWORKS

A training dataset T is a set of N classified instances $\{(x_1, y_1), \dots, (x_N, y_N)\}$ for some unknown function $y = f(x)$. The x_i values are typically vectors of the form $(x_{i1}, x_{i2}, \dots, x_{im})$ whose components are discrete or real values, called features or attributes. Thus, x_{ij} denotes the value of the j -th feature X_j of x_i . In what follows, the i subscript will be left out when implied by the context. For classification purposes, the y values are drawn from a discrete set of k classes, i.e. $y \in \{C_1, C_2, \dots, C_k\}$. Given a set $S \in T$ of training examples, a learning algorithm induces a classifier h , which is a hypothesis about the true unknown function f . Given new x values, h predicts the corresponding y values.

Many Artificial Neural Network architectures are available [5]. The architecture is chosen based on the kind of problem that should be solved. Since in this work we consider pipeline leak detection as a classification problem, we indicate the Multilayer Perceptron (MLP). The MLP is formed by at least 3 (three) layers: an input layer, an output layer, and one or more intermediate layers. Each element of the output layer on our neural network model produces the output calculated by Equation 1, whereas y_i^o represents the output of the i^{th}

processing element, w_{ij}^O and w_{ij}^H represent the connection weights between processing elements i and j in output and hidden layers, I_k represents the input of the k th processing element and f represents the transfer function for processing elements. If we express the overall action of the neural network by φ then $y(t) = \varphi(x(t))$ where $x(t)$ is a sample of the data to be classified.

$$y_i^O = f\left(\sum_{j=1}^m w_{ij}^O \cdot f\left(\sum_{k=1}^n w_{jk}^H \cdot I_k\right)\right) \quad (1)$$

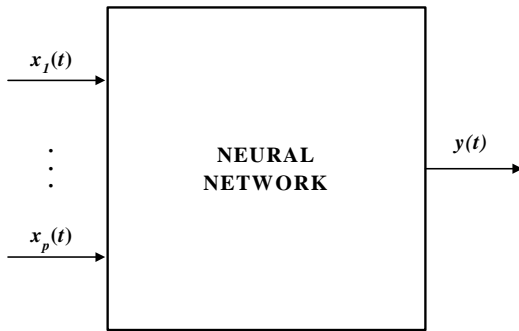


Figure 1 - ARTIFICIAL NEURAL SYSTEMS ARCHITECTURE

The learning algorithm used for training the networks in this work is back propagation, which updates the weights using the error rate calculated on the network output with the desired output (label).

ANN exhibit properties that can actually render them well apt for processing routine measurements made in pipelines, and can be used in quite innovative leak-detection systems. An ANN may be regarded as a nonlinear mathematical function that transforms a set of input variables into a set of output variables (Bishop, 1994), the transformation function depending on weights that are determined based on a set of training examples. Training can be computationally very expensive and depends on the sizes of the networks and the number of training examples. Once weights are calculated, processing the new data is fast. In addition to offering very high processing speeds, ANNs are, in principle, capable of learning a general solution to a problem from a limited number of examples. Nevertheless, the use of ANN does not yet appear to have received much attention for leak detection. However, there are many successful applications in tasks of similar complexity. A general review is presented in Bishop (1994). There are some specific reasons why using ANN for leak detection is quite promising (Bishop, 1994), namely, (1) it is difficult to find an adequate first-principle or model-based solution; (2) new data must be processed at high speed; and (3) the system must be impervious to noise. Another important aspect for the development of ANN, as outlined by Bishop

(1994), is that a large set of data must be available. In general, this information is either available or can yet be developed for pipelines.

METHODS FOR CONSTRUCTING ENSEMBLES OF CLASSIFIERS

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by obtaining a vote of their predictions. There are three fundamental reasons for constructing ensembles of classifiers [3]. The first reason is a statistical one. A learning algorithm refers to searching for a space H of hypotheses to identify the best hypothesis in this space. The statistical problem arises when the amount of training data available is too small compared to the size of the hypothesis space. Without sufficient data, the learning algorithm can find many different hypotheses in H that all result in the same accuracy in training data. By constructing an ensemble out of all of these accurate classifiers, the algorithm can average their votes and reduce the risk of selecting the wrong classifier. Figure 2 (top left) depicts this situation. The outer curve denotes the entire search space H ; the inner curve denotes the hypothesis that gives good accuracy in training data. The f point depicted in the figure denotes the unknown true hypothesis, and we can see that, by averaging out the accurate hypothesis, we can find a good estimate of f .

The second reason is computational. Many learning algorithms work by performing some form of local search that may get stuck in local optima. For example, neural network algorithms employ gradient descent to minimize an error function over the training data. In cases where there is enough training data (so the statistical problem is absent), it may still be very difficult, in terms of computation, for the learning algorithm to find the best hypothesis. Indeed, optimal training of neural networks is NP-hard. An ensemble constructed by running the local search from many different starting points may provide a better estimate of the true unknown function, as shown in Figure 2 (top right).

The third reason is representational. In most applications of machine learning, the true function f cannot be represented by any of the hypotheses in H . By forming weighted sums of hypotheses drawn from H , it may be possible to extend the space of representational functions. Figure 2 (bottom) depicts this situation.

The next section describes the proposed method for combining classifiers for pipeline LDS.

A PROPOSAL FOR COMBINING CLASSIFIERS WITH PIPELINE LEAK DETECTION

In order to model a problem, the first step of the process is selecting what features are related to it. The referred literature [1][2][4][6] suggests that the features affecting models of pipelines for leak detections are flow rate, density, temperature, and pressure. The authors find that distinguishing a

relationship

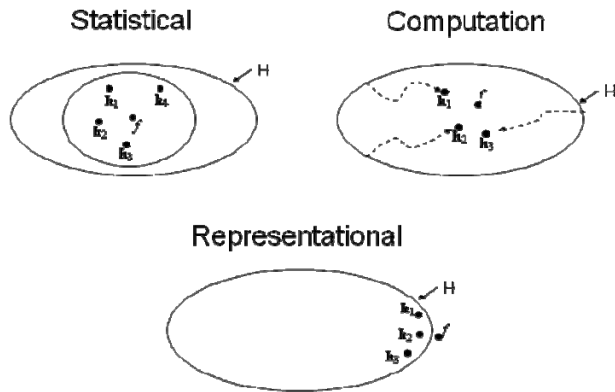


Figure 2 - THREE FUNDAMENTAL REASONS FOR CONSTRUCTING ENSEMBLES OF CLASSIFIERS [4]

between measured values in the pipeline input and output in order to detect abnormal patterns is the purpose of internal methods for leak detection.

An induced ANN is a model that correlates non-linearly the features that describe the problem. In our application, we use a neural network to correlate the features, separating the instances into normal and abnormal operations of the pipeline (classification problem). Because of this, we need instances that represent the normal and the abnormal situations. Each instance is a set of values of flow rate, temperature, density and pressure read in some instance t in pipeline's input and output. Therefore, an ANN detects normal and leak situations if it is presented with instances of normal and leak situations in training phases. However, in real applications, it is hard to find data with leak situations, which leads us to combine other artificial specialists in our LDS. Normal events in the pipeline, like start and stop pumping, adjustment of valve sets and batch changes, are responsible for false alarms in LDS. Thus, an ANN induced for classifying these events can help the LDS in detecting leaks.

Our LDS is composed of 4 (four) specialists, whose decisions are combined to formulate an answer to the operator. One specialist is responsible for classifying the read values (instances) into normal or leak classes; another specialist is responsible for classifying the same input instance into start and stop pumping, adjustment of valve sets, batch changes or continuous flow; yet another specialist is responsible for detecting the percentage of the leak when the latter is detected; and the last specialist is responsible for detecting the location of the leak in the pipeline. Each specialist is an ANN, and is trained separately. The decisions of the first two specialists – the first one who detects the presence or absence of leaks as well as the second one who classifies the event that is occurring – are combined to determine what is occurring inside the pipeline. The answers of the specialists are the events occurring

and, when a leak is detected, the answer can be, for example, “there is a leak or the pumping is starting”. The operator will decide what is happening – if the operator knows that there is not a start pumping, than a leak is detected and the other two specialists pronounce the location and the leak's flow percentage.

Our proposal allows us to train the ANN for the leak detection of a pipeline initially using simulated data and, as the operator interacts with the system, declaring that an event is occurring, and not a leak, new instances are collected. With these new instances, the network can be re-trained with the old one and this new data. This process allows the model to adapt over time.

Another observation relates to the available data: if there is no instance about leaks, only the specialist (ANN) in events can be initially trained. When the ANN detects an event that is not really happening, if the operator verifies that a leak is taking place, in time the system is fed by leak instances, which allows an ANN for leak detection to be induced in the future.

Since Neural Networks find non-linear correlation in input variables, and these variables are likely to be pipeline-dependent, one ensemble for each pipeline is necessary. However, further studies using real and simulated data will be necessary to verify this statement.

ADDUT SYSTEM

In order to evaluate and put our proposal to practice, a system, called ADDut, is under construction. This tool has two main functions. The first one is to help the user:

1. Visualize the operations behavior, based on the signals obtained in time of pressure, density, flow rate, temperature, and others that should be considered – Figure 4, Annex A, shows a print screen of the system that depicts the signals over the time. The depicted data are from simulated operations in Pipeline Studio®; and
2. Use an ensemble constructed for the pipeline under observation and check if there is any leak in the loaded data – Figure 5, Annex A, demonstrates the resulting analysis of the ensemble.

The second one is to help train the specialists (neural networks) into their own specific task. The interface of this part of the system is in development.

EXPERIMENTS

Due to the difficulties in obtaining data from real leak situations in oil pipelines, we first tested our approach using simulated dataset. In this phase, our dataset was simulated using Pipeline Studio®. Figure 5, Annex A, shows a schema of the pipeline on which the dataset was simulated. The pipeline is 50 km long. Figure 6, Annex A, shows the instances plot in time from left to right. Each graph is related to a feature (F_{in} represents the flow read at input point of the pipeline; F_{out}

represents the flow read at output; and so on). The time between two instant readings is 15 seconds. There are two weeks of simulated operations, totaling 80641 instances. Initially, two class features were created. The features are leak, containing the values “normal” (66629 instances, or 82.62% of the entire dataset) and “leak” (14012, or 17.38%); and event, containing the values “normal” (77895 instances, or 96.60% of the entire dataset), “SP” – Stop Pumping (1010 instances, or 1.25%); “SPVS” – Stop Pumping and Valve Setting (1168, or 1.45%); “SPBCVS” – Stop Pumping, Batch Changing and Valve Setting (489, or 0.60%); and “VS” – Valve Setting (79, or 0.10%). We induced two ANN, each of them having one of the class labels. The overall error rate obtained in each network are 1.81% using “leak” feature as the class feature (6.03% of error rate in “leak” class and 0.82% in “normal” class) and 0.99% of error rate using “event” feature (0.01% in “normal”, 11.58% in “SP”, 55.91% in “SPVS”; 1.02% in “SPBCVS”; and 17.72% in “VS”). Figures 6 and 7, Annex A, shows, respectively, the results obtained with the ANNs induced, which shows, for each instance, the labeled and predicted classes.

Figure 7, Annex A, shows that all leak situations were detected; in three occasions, there was a leak but it was not detected and there were only two false alarms.

Real data were obtained only for normal situations. The dataset obtained allowed us to observe that, in real situations, many events occur in parallel and in four days of operations – in this dataset, we observed eight batch changing, many stop and start pumping and many valve adjustments. Due to confidentiality agreements, we cannot show the signals of the operations. On the other hand, the overall error rate obtained in event network is 2.99% (0.14% in “normal”, 66.15% in “VS”, 45.91% in “SP”; 44.75% in “BC”; 58.36% in “STP” – Start Pumping; and 14.29% in “BCSTP” – Batch Changing and Start Pumping). These results, though not so good, were expected, since real data has more uncertainties associated to it. Also, we could observe that many from the events were classified as undetermined by the ANN, which means that some events should be joint, for example Stop and Start Pumping, since both patterns are similar.

CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a method for constructing ensembles of ANNs for pipeline leak detection. The method we proposed makes it possible to group different specialists, each one looking into the problem from different points of view. The results in our experiments were promising. The next step of our work includes testing our proposal in real world data. This is a difficult task since obtaining this kind of data is not a simple endeavor. We also intend to substitute the event ANN by one specialist to detect each event.

REFERENCES

- [1] Belsito, S.; Lombardi, P.; and Benerjee, S. “Leak Detection in Liquefied Gas Pipelines by Artificial Neural Networks”. *AIChE Journal*, volume 44, number 12, pages 2675-2688. (1998).
- [2] Carvalho, A.A.; Rebello, J.M.A.; Sagrilo, L.V.S.; Camerini, C.S. and Miranda, I.V.J. “MFL Signals and artificial neural networks applied to detection and classification of pipe weld defects”, *NDT&E International*, volume 39, pages 661–66 (2006).
- [3] Dietterich, T. G. “Ensemble methods in machine learning”. In First International Workshop on Multiple Classifier Systems. *Lecture Notes in Computer Science*, volume 1857, pages 1–15, New York. Springer Verlag (2000).
- [4] Ellul, I. “Pipeline Leak Detection”. *Chem. Eng.*, number 461 (1989).
- [5] Haykin, S. *Neural Networks*. Macmillan College Publishing Company, Inc, New York (1994).
- [6] Silk, M.G., and Carter, P. “A Review of Means of Pipeline Leak Detection Independent of Flow Measurement”. EC Project DEPIRE DGXII Meeting, Harwell, UK (1995).
- [7] Silva, H.V.; Morookab, C.K.; Guilherme, I.R.; Fonseca, T.C. and Mendes, J.R.P. “Leak Detection in petroleum pipelines using a fuzzy system”, *Journal of Petroleum Science and Engineering*, volume 49, pages 223-238 (2005).

ANNEX A

ADDUT SYSTEM INTERFACE



Figure 3 - VISUALIZATION OF THE SIGNALS IN TIME. FROM TOP TO BOTTOM: FLOW RATE (INPUT AND OUTPUT), TEMPERATURE (INPUT AND OUTPUT), PRESSURE (INPUT AND OUTPUT) AND DENSITY (INPUT OUTPUT).

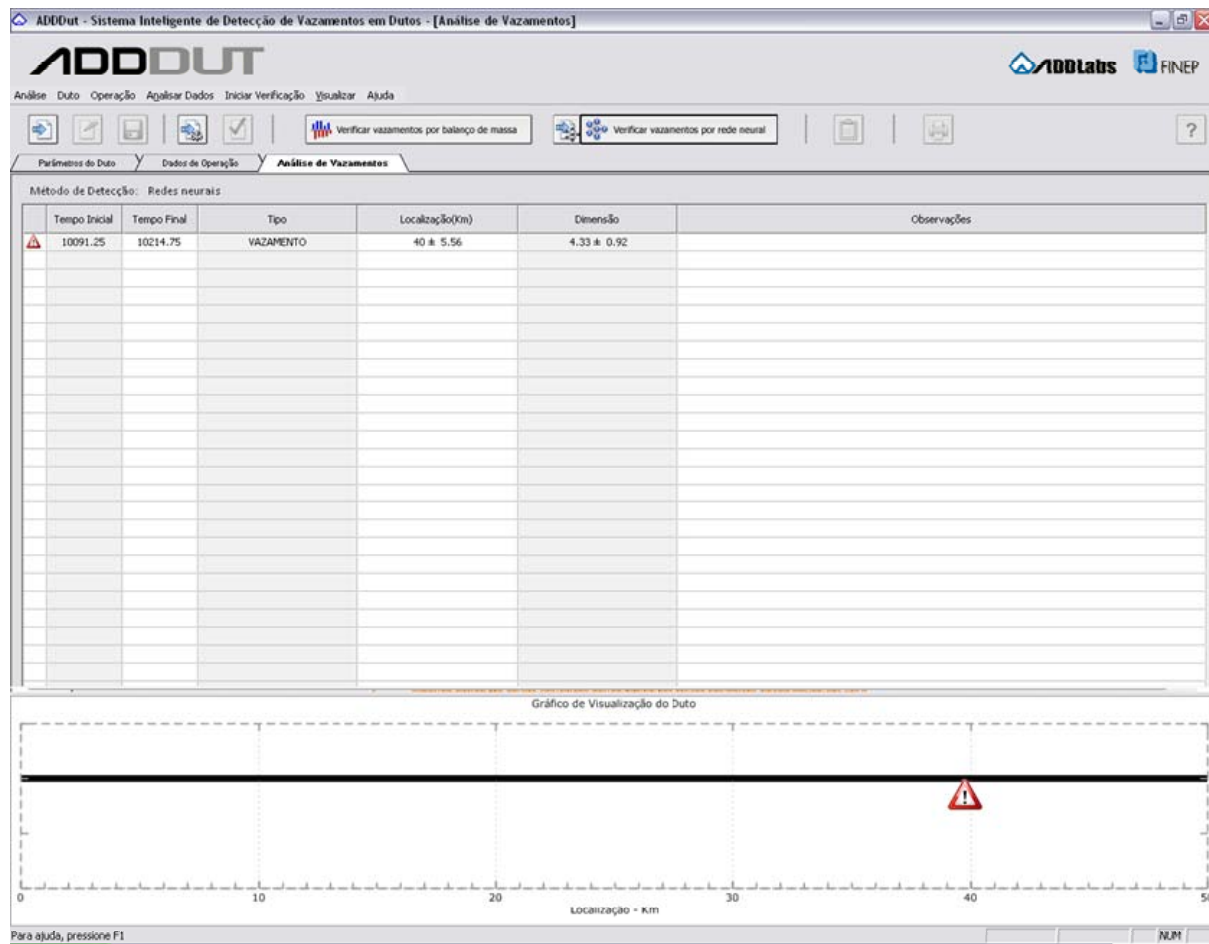


Figure 4 - OUTPUT OF THE ADDDUT SYSTEM: TIME ELAPSED, TYPE OF ALARM, LOCALIZATION IN PIPELINE AND PERCENTAGE OF THE FLOW RATE.

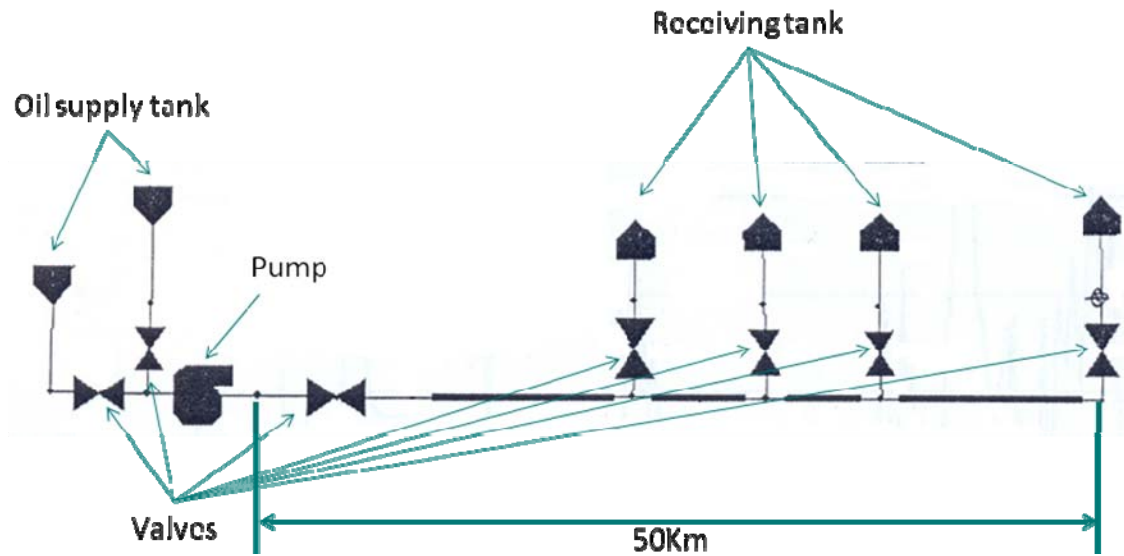


Figure 5 - PIPELINE SCHEMA OF THE SIMULATED DATASET.

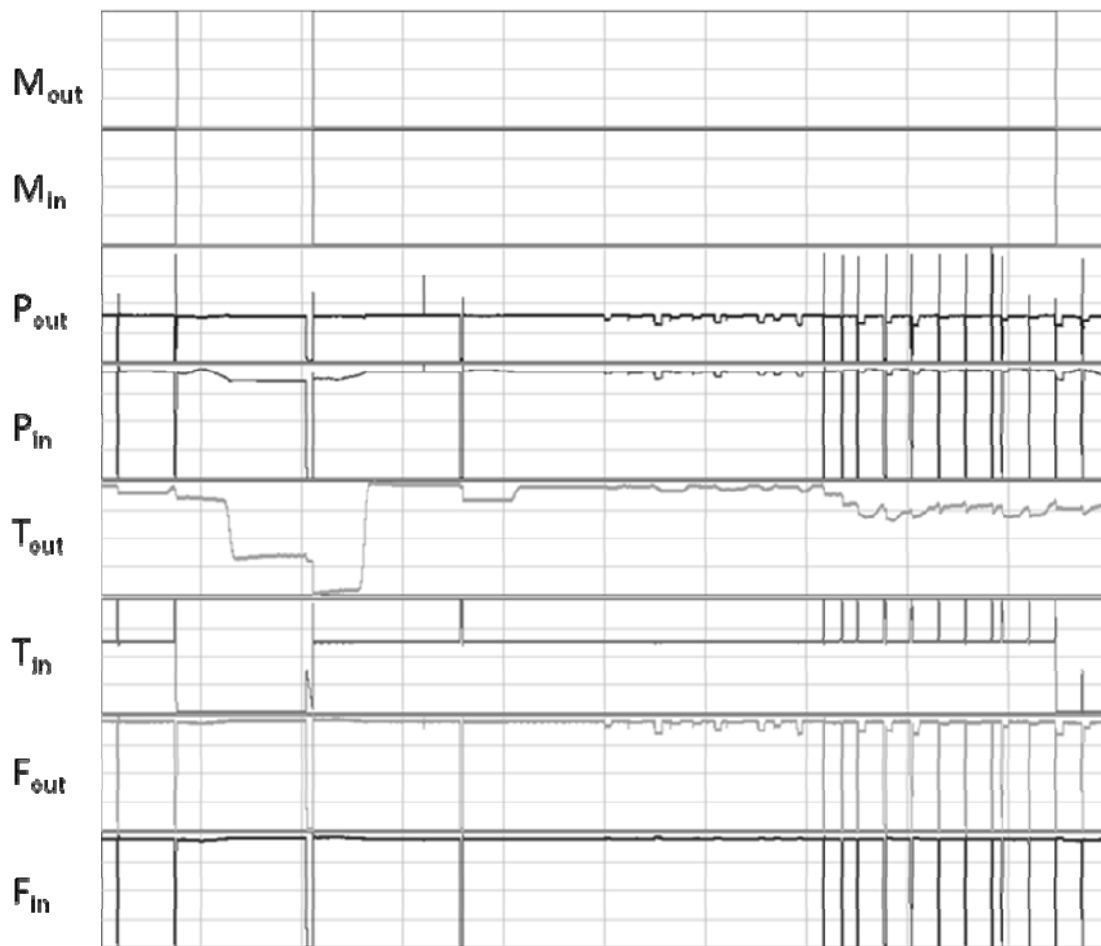


Figure 6 - VISUALIZATION OF THE SIMULATED DATASET.



Figure 7 - VISUALIZATION OF THE LABELED INSTANCES IN TIME AND THE NEURAL NETWORK PREDICTION. RED REPRESENTS INSTANCES LABELED AND/OR PREDICTED AS NORMAL AND GREEN REPRESENTS INSTANCES LABELED AND/OR PREDICTED AS A LEAK.



Figure 8 - VISUALIZATION OF THE LABELED INSTANCES IN TIME AND THE NEURAL NETWORK PREDICTION. RED REPRESENTS INSTANCES LABELED AND/OR PREDICTED AS NORMAL AND OTHER COLORS REPRESENT INSTANCES LABELED AND/OR PREDICTED AS OTHER EVENTS (PUMP START, VALVE ADJUSTMENT AND/OR BATCH CHANGES).