





POLITECNICO DI BARI

D.R.R.S

05

Doctor of Philosophy in Risk and Environmental,  
Territorial and Building Development

2022

Coordinator: Prof. Michele Mossa

XXXIV CYCLE

Hydraulic And Maritime Construction And Hydrology

DICATECh

Department of Civil, Environmental, Building  
Engineering and Chemistry

Bilal Derardja

**Innovative Approaches for Mapping the  
Pressurized Irrigation Systems Performances  
Under Unsteady Flow Conditions**

Prof. Umberto Fratino  
DICATECh  
Polytechnic University of Bari

Dr.Eng. Nicola Lamaddalena  
CIHEAM-Bari



POLITECNICO DI BARI

**D.R.R.S**

**05**

Dottorato di Ricerca in Rischio e Sviluppo Ambientale, Territoriale ed Edilizio

2022

Coordinatore: Prof. Michele Mossa

XXXIV CICLO

Costruzioni Idrauliche e Marittime e Idrologia

**DICATECh**

Dipartimento di Ingegneria Civile, Ambientale, del Territorio, Edile e di Chimica

Bilal Derardja

**Approcci innovativi per la mappatura delle prestazioni dei sistemi irrigui in pressione in condizioni di moto vario**

Prof. Umberto Fratino  
DICATECh  
Polytechnic University of Bari

Dr.Eng. Nicola Lamaddalena  
CIHEAM-Bari



Politecnico  
di Bari

Department of Civil, Environmental, Land, Building  
Engineering and Chemistry

RISK AND ENVIRONMENTAL, TERRITORIAL  
AND BUILDING DEVELOPMENT

Ph.D. Program

SSD: ICAR/02- Hydraulic and Maritime Constructions and  
Hydrology

**Final Dissertation**

---

Innovative Approaches for Mapping  
the Pressurized Irrigation Systems  
Performances Under Unsteady Flow  
Conditions

---

by

DERARDJA Bilal:

Supervisors:

Prof. Umberto Fratino

Dr. Eng. Nicola Lamaddalena

*Coordinator of Ph.D. Program:*

*Prof. Michele Mossa*

---



Politecnico  
di Bari

Department of Civil, Environmental, Land, Building  
Engineering and Chemistry  
**RISK AND ENVIRONMENTAL, TERRITORIAL  
AND BUILDING DEVELOPMENT**  
Ph.D. Program  
SSD: ICAR/02- Hydraulic and Maritime Constructions  
and Hydrology

**Final Dissertation**

---

# Innovative Approaches for Mapping the Pressurized Irrigation Systems Performances Under Unsteady Flow Conditions

---

by

DERARDJA Bilal:

Referees:

Prof. Andre Daccache

Prof. Emilio Camacho Poyato

Supervisors:

Prof. Umberto Fratino

*U. Fratino*

Dr. Eng. Nicola Lamaddalena

*Coordinator of Ph.D Program:*

*Prof. Michele Mossa*

## ***EXTENDED ABSTRACT (English)***

Nowadays, the management of pressurized irrigation networks requires plenty of information to provide an efficient and reliable service to farmers. An approach called MASSPRES is being developed as a collaboration between FAO and CIHEAM-Bari with the goal of developing a reliable modernization strategy and improving the performance of pressurized irrigation systems. Mapping the perturbation, which is represented by the unsteady state flow analysis, is one of the most significant steps of this approach. The perturbation in irrigation networks is often created when sudden changes in flow rates occur in the pipes. This is essentially due to the manipulation of hydrants (service outlets) according the operational scenarios called configurations.

During the perturbation occurrence, pressure waves propagate through the networks pipes that may lead to a signification pressure variation. This variation could expose the irrigation system's components to a substantial danger that could cause significant damage.

To model such a phenomenon, several computational algorithms have been developed. The majority of these models aimed to simulate the unsteady state conditions induced by the farmer's behavior. The most recent ones are efficient enough to provide a good image of the perturbation occurrence through different indicators, however, one of the main draw backs of such model is the significantly high time and computational costs.

In the present work, two different generations of models were developed. The first is a directly programmed model that was devaloped based on the method of characteristics and two indicators have been introduced: i) The hydrant risk indicator (HRI), which is defined as the ratio between the participation probability of hydrant no.

x in the riskiest configurations and its total number of participations; and ii) the relative pressure exceedance (RPE), which provides the variation of the unsteady state pressure with respect to the nominal pressure. The two indicators could help managers better understand the network behavior with respect to the perturbation by defining the riskiest hydrants and the potentially affected pipes.

Although, knowing the riskiest hydrants in the network is an important piece of information, managing ramified networks in real time will remain a difficult task to handle in real time. Thus, the need of developing a real time Decision Support System (RTDSS) that could process such information and guide the manager in real time is crucial.

For this aim, two thousand configurations (operational scenarios) were simulated using the directly programmed model from the first step and fed to train a new model based on deep learning with the objective of forecasting the maximum pressure occurred due to the perturbation at each section. The occurred pressure is represented as classes according to the case sensitivity and the required precision. Steps of 1, 2 and 3 bars were simulated. The model proved to be significantly time saving compared to previous approaches as the results are produced instantaneously with a forecasting accuracy of 85 %. Furthermore, using the confusion matrix, the error committed by the model is of one class lower or higher that may be considered tolerable according to the system sensitivity.

This approach was applied on a pressurized on-demand irrigation system located in south of Italy that consists of 19 hydrants and covers 57 hectares. Nonetheless, the deep learning-based model needs to be trained on each section. Thus, as a main step of the method of characteristics, the network was discretized into 1017 sections of 3 meter each. Training the deep learning model for such number of sections is not practical and time consuming. For this reason, a code was developed using autoencoding combined with t-distributed stochastic neighbor embedding (t-SNE) algorithm for features extraction and their visualization respectively. It is principally to cluster the sections according to their behavior to the perturbation, thus, reduce the

number of trainings of the previously mentioned model. Nine zones of similar behavior were determined by the present developed code and the deep learning model will be trained only on these zones representing all the sections.

The two last developed codes could be integrated for a decision support system (DSS) for modelling the perturbation in the on-demand pressurized irrigation networks that would add a significant contribution to provide practical recommendations for real-time decision-making processes. which was not possible using directly programmed software.

**Keywords:** pressurized irrigation systems; on-demand operation; perturbation; unsteady state flow; method of characteristics; deep neural networks; feature extraction.





## ***EXTENDED ABSTRACT (Italian)***

Attualmente la gestione delle reti di irrigazione in pressione richiede operatori competenti e qualificati capaci di interpretare una larga mole di dati e informazioni per fornire un servizio efficiente e affidabile agli utenti. Un approccio tecnico scientifico, denominato MASSPRES, utile ad affrontare questa problematica è stato proposto, grazie a una collaborazione tra FAO e CIHEAM-Bari, con l'obiettivo di sviluppare una strategia di modernizzazione affidabile e migliorare le prestazioni dei sistemi irrigui in pressione. La mappatura delle variazioni di pressione, che è definita dall'analisi del moto vario che si osserva nelle reti irrigue in pressione, è uno dei modelli più significativi tra tutti quelli proposti. La perturbazione nelle reti di irrigazione si genera per effetto di improvvise variazioni della portata trasportata nelle condotte; evenienza dovuta per lo più alla movimentazione degli idranti che si osserva in ragione delle esigenze idriche delle colture e/o delle consuetudini (comportamenti abituali) degli agricoltori (scenari di esercizio o configurazioni).

Quando si genera la perturbazione, attraverso le tubazioni della rete si propagano onde di pressione che possono portare ad una sua variazione significativa, il che espone l'infrastruttura a rischio perché detta variazione potrebbe causare danni significativi alle condotte che compongono la rete irrigua.

Per modellare tale fenomeno, nel tempo, sono stati sviluppati diversi algoritmi di calcolo, la maggior parte dei quali tendeva a simulare il fenomeno fisico osservato come indotto dalle variazioni di esercizio dell'infrastruttura con significativo onere computazionale. Più di recente, sono stati esplorati anche approcci diversi, come quelli che fanno uso di indicatori sintetici, che però hanno ancora il loro limite nel tempo di elaborazione e quindi negli elevati costi computazionali.

In questa attività di ricerca dottorale sono state sviluppate due diverse generazioni di modelli. Il primo è un modello "direttamente programmato" che è stato

sviluppato sulla base del metodo delle curve caratteristiche. Il concetto di programmazione diretta si basa sull'esecuzione di istruzioni precedentemente programmate e su un insieme di equazioni matematiche che seguono scenari probabili stabiliti.

Allo scopo sono stati definiti due indicatori: i) l'indicatore di rischio dell'idrante (Hydrant Risk Indicator, HRI), che rappresenta il rapporto tra la probabilità che l'idrante ennesimo sia parte attiva negli scenari di esercizio a maggior rischio di fallanza a causa dell'insorgenza di fenomeni di moto vario e il numero totale di volte in cui è parte attiva in una configurazione; e ii) il superamento di pressione relativa (Relative Pressure Exceedance, RPE), che fornisce la variazione della pressione durante l'occorrenza del moto vario rispetto al valore di pressione nominale della tubazione coinvolta. Essi possono aiutare i gestori a comprendere meglio il comportamento della rete rispetto alla perturbazione che può essere generata individuando gli idranti che possono indurre le perturbazioni più gravose e le tubazioni a maggiore rischio potenziale.

Sebbene conoscere gli idranti più rischiosi della rete sia un'informazione importante, la gestione di reti irrigue in tempo reale rimane tuttavia un compito difficile, per cui è emersa sin da subito la necessità di sviluppare un sistema di supporto alle decisioni (Real Time Decision Support System, RTDSS), in grado di elaborare in tempo reale tali informazioni e guidare le scelte gestionali di esercizio della rete.

A tal fine sono state simulate duemila configurazioni (scenari di esercizio) utilizzando il modello programmato direttamente dal primo step, sì da alimentare e addestrare un nuovo modello basato sul deep learning al fine di effettuare una previsione affidabile del valore di pressione massima che può generarsi in ciascuna sezione per effetto del moto vario. Il nuovo modello introdotto si è rivelato particolarmente efficace in quanto molto più veloce nel generare l'informazione cercata rispetto a quanto richiesto con gli approcci precedenti (il dato è stato prodotto istantaneamente con una accuratezza dell'85%).

Questo approccio è stato applicato a un reale sistema irriguo con esercizio a domanda situato nel sud dell'Italia; la rete si compone di 19 idranti distribuiti su una superficie di 57 ettari. Per tale rete, al fine di ridurre ulteriormente i tempi di elaborazione

con la tecnica dell'apprendimento profondo (deep learning), è stato sviluppato un altro modello che si avvale dell'autoencoding combinato con l'algoritmo t-SNE (t-distributed Stochastic Neighbor Embedding) per l'estrazione delle caratteristiche e la loro visualizzazione. Tale modello raggruppa le sezioni (sono 1017 nel caso in studio) in ragione del comportamento rispetto alla perturbazione, riducendo così il numero di addestramenti. In tal modo il codice di calcolo sviluppato ha individuato nove zone di comportamento analogo, e, conseguentemente, il modello di deep learning è stato addestrato solo su queste. Questi due modelli possono essere integrati per un sistema di supporto alle decisioni (DSS) utile a simulare il comportamento in moto vario nelle reti irrigue alla domanda e soprattutto consentirebbe di poter fornire, in tempo reale, informazioni indispensabili per operare scelte gestionali consapevoli e in sicurezza.

**Parole chiave:** reti di irrigazione in pressione; reti con esercizio a domanda; perturbazione; moto vario; metodo delle curve caratteristiche; reti neurali profonde; estrazione delle caratteristiche.



# INDEX

<b>1</b>	<b>CHAPTER 1. INTRODUCTION AND MOTIVATION</b> .....	1
1.1	BACKGROUND.....	1
1.2	OBJECTIVES AND CONTRIBUTIONS.....	6
<b>2</b>	<b>CHAPTER 2. OVERVIEW OF</b> .....	7
2.1	INTRODUCTION.....	7
2.2	UNSTEADY STATE FLOW CONDITIONS HYDRAULIC PRINCIPLES BACKGROUND....	8
2.2.1	The pressurised irrigation systems flow conditions .....	8
2.2.2	Causes of transient flow.....	8
2.2.3	Factors affecting the transient flow severity .....	8
2.2.4	Developing the unsteady state flow equations .....	9
2.3	ARTIFICIAL NEURAL NETWORKS BACKGROUND.....	23
2.3.1	Introduction to machine learning.....	23
2.3.2	Supervised and unsupervised Learning Algorithms .....	23
2.3.3	Deep neural networks .....	24
2.3.4	Learning process .....	26
2.3.5	Autoencoder module.....	31
2.3.6	t-distributed Stochastic Neighbor Embedding (t-SNE) .....	32
<b>3</b>	<b>CHAPTER 3. METHODOLOGY</b> .....	35
3.1	STUDY AREA.....	35
3.2	THE MOC BASED MODEL.....	37
3.2.1	The calculation process .....	37
3.2.2	The Hydrant Risk Indicator (HRI) .....	39
3.2.3	Relative pressure exceedance (RPE) .....	40
3.3	THE DEEP NEURAL NETWORK BASED MODEL.....	41
3.3.1	The functioning principle of the pressure classes forecasting model.	41

3.3.2	The model's architecture .....	44
3.3.3	The clustering model.....	44
3.4	DATA PRE-PROCESSING .....	46
3.4.1	One-Hot-Encoding data .....	46
3.4.2	Split into Train and Test Sets .....	47
3.5	PERFORMANCE EVALUATION.....	48
3.5.1	Confusion matrix.....	48
3.5.2	Learnig curve.....	49
<b>4</b>	<b>CHAPTER 4. RESULTS AND DISCUSSION</b> .....	<b>51</b>
4.1	RESULTS OF THE MOC BASED MODEL .....	51
4.1.1	The uniformity of the randomly generated configurations.....	51
4.1.2	Comparison of the model's output to the commercial product Hammer 51	
4.1.3	Computation of the RPE representation.....	53
4.1.4	Calculation of HRI .....	54
4.2	RESULTS OF THE DEEP NEURAL NETWORK BASED MODEL .....	56
4.2.1	Average accuracy using k-fold cross validation.....	57
4.2.2	Confusion matrix.....	58
4.2.3	Learning process .....	60
4.3	RESULTS OF THE CLUSTERING MODEL .....	62
4.3.1	Learning process .....	64
4.3.2	Clustering output.....	65
<b>5</b>	<b>CHAPTER 5. CONCLUSION</b> .....	<b>67</b>
	REFERENCES .....	73

# **CHAPTER 1. INTRODUCTION AND MOTIVATION**

## **1.1 Background**

The rising population and living standards are increasing the demand for agricultural goods, putting strains on water supplies and further on water-use decision-making mechanisms in irrigation systems to the test (Fernández García, et al., 2020). Modernization of irrigation networks has implied the replacement of previous open channel networks by pressurized ones. This change allows the implementation of pressurized irrigation systems (drip and sprinkler) which guarantee better services to the users and a higher distribution efficiency in comparison with the previous systems (Lamaddalena & Sagardoy, 2000). The optimum design and management of pressurized water distribution networks for irrigation have had the attention of the researchers, managers and decision-making agents (Fouial, et al., 2020; Rodríguez Díaz, et al., 2020).

Operating on-demand, systems are designed to deliver water at the flow rates and pressures required by on-farm irrigation systems, considering the time, duration, and frequency as defined by the farmers (Calejo, et al., 2008). In on-demand irrigation systems, a group of hydrants operating at the same time is known as a configuration. Changing from a configuration to another is the main origin of perturbation in such systems (Lamaddalena, et al., 2018).

Many authors (Clemmens, 2006; Playán & Mateos, 2006) have emphasized that improving the performance of existing irrigation schemes is a critical topic to decrease excessive water use and enhance system efficiency. To build up a reliable modernization strategy and improve the performance of the irrigation systems, a methodology named MASSCOTE (mapping system and services for canal operation techniques) has been developed by Food and Agriculture Organization (FAO) (Renault, et al., 2007). Aimed at a similar purpose for pressurized irrigation networks, a new



approach called MASSPRES (mapping system and services for pressurized irrigation networks) is being developed as a collaboration between FAO and CIHEAM-Bari (Centre International de Hautes Etudes Agronomiques Méditerranéennes-Bari). One of the most important aspects in the new approach is mapping the perturbation.

Unsteady state flow analysis is one of the most challenging flow problems both in the design and operation phases of water pipeline systems. Its control is essential to guarantee a safe functioning of the pressurized networks (Abuiziah, et al., 2013). The relationship between the irrigation network behavior and the origin of the perturbation is governed by differential equations. However, the correlation between the response and the factors affecting the perturbation is complex. Nevertheless, many studies have developed models based on different numerical methods. Among the others, (Triki, 2018) conducted research for developing a model to study mainly two scenarios, water-hammer up- and down-surge events. A 1-D unconventional water-hammer model inserting the Vitkovsky and the Kelvin-Voigt formulations for describing the flow behavior and solved by the method of characteristics (MOC). Another study was elaborated to investigate the effects of pump's moment of inertia on pipeline water hammer control using the MOC (Wan, et al., 2019).

Perturbation in pressurized irrigation systems is the unsteady state conditions, in which large pressure waves may occur in the system and could result a failure in the pump or one of its components, pipe ruptures or in the best scenario system's fatigue. Most of these networks are on-demand irrigation systems in which farmers have continuous access to water. This means that the flow in the network pipes may be significantly variable according to the farmers behavior. Thus, it is essential to simulate this variation by generating different operational scenarios changing the hydrants functioning modes (closed, opened, closing or opening) to ensure the safety and the good functioning of the system (Boulos, et al., 2005). The operational scenario was assigned as a configuration i.e., a configuration in this case is a set of various functioning modes for the different hydrants. Knowing that an irrigation hydrant is an outlet connecting the pressurized water distribution network to the on-farm irrigation systems. The perturbation is created by changing the functioning mode of some

hydrants representing the behavior of farmers. The four possible functioning modes of a hydrant are, remaining closed, remaining opened, in the closing and opening process.

Codes for simulating the perturbation in pressurized irrigation systems have come through two generations. The first includes directly programmed software using different available numerical methods such as the method of characteristics (MOC). The direct programming concept is based on the execution of previously programmed instructions and a set of mathematical equations that follow established probable scenarios. This was used and explained by (Lamaddalena, et al., 2018) and (Derardja, et al., 2019), in which some indicators were introduced. Different numerical methods including the MOC are explained in detail in numerous text books such as (Záruba, 1993; Sharp & Sharp, 1995).

Based on the latter mentioned method, a friendly user software was developed and new indicators have been introduced to represent the behavior of the pressurized irrigation networks to the perturbation propagated through the network pipes, knowing that different boundary conditions were considered. The hydrant risk indicator (HRI), that describes the degree of risk of each hydrant on the system by causing pressure waves propagating through the system pipes, and the relative pressure exceedance (RPE), that represents the variation of the pressure in the system with respect to the pipe nominal pressure. The latter is interpreted as a warning signal that a pipe in the system may collapse. This model is used as well as big data generator. This study was applied on a pressurized on-demand irrigation system located in south of Italy that consists of 19 hydrants and covers 57 hectares.

According to both the computer's characteristics and the size of the network to be analyzed, the computational time may vary. Using an average characteristics commercial computer of nowadays, the software takes 15 to 16 hours to simulate 2000 configurations for the irrigation network used in the present study. This software is considered to be time and computationally expensive which is expected according to the method followed (MOC) and the number of iterations to be handled. The output is a large dataset, due to a discretization through space (along the different network

pipes) and a second discretization through time (all over the simulation time) with a step that is defined by a specific equation taking into consideration the wave celerity and the minimum section length (Larock, et al., 1999).

However, to establish a consistent modernization strategy for improving the irrigation services to users and mainly to integrate them in decision-making tools, new powerful and flexible models able to give results on real time, needs to be introduced (Perea, et al., 2019).

The second generation of codes were introduced using deep neural networks (a subset of machine learning). The latest technique is addressed to the computation and time consumption issues of the first-generation software, consequently to be integrated in real-time decision-making systems. It is important to mention that, the directly programmed software was used as big data generators for training the model developed using deep neural networks.

In problems with a large dataset, machine learning algorithms can provide a better alternative solution to the directly programmed software in term of speed and accuracy. Although, in an initial phase, machine learning models may require significant time for being properly trained (Géron, 2019), it could provide instantaneous results with acceptable accuracy in usage phase. As a part of artificial intelligence, it is worth to mention, that machine learning is the study of algorithms that automatically learn through an experience. The aim of automated learning is to make forecasting or making decisions without even being explicitly programmed to do. Practically, it turns out to help the machine following different approaches to develop its own algorithm through the analysis of the available data (Jin, 2020). For developing a more powerful and flexible model there is a need for a high-level features extraction from data. This is one of the main advantages of deep learning algorithms (Raschka & Mirjalili, 2017; Deisenroth, et al., 2020), thus, it was selected for the present research. Deep Learning is a machine learning technique that constructs artificial neural networks to reproduce the structure and function of the human brain. It is a tool for processing big datasets, as its performance improves while analyzing larger data (Brownlee, 2016).

Introducing a new method for assessing the behavior of pressurized irrigation systems in case of unsteady flow conditions based on deep learning shortens drastically the time to instantaneous analysis. This offers a great potential to be used as a real-time decision-making tool and providing the manager with information about the risks of perturbations in the network.

The early mentioned studies were elaborated working on an irrigation network in south of Italy, that is of 3 km pipes length. The perturbation was simulated using a directly programmed software by generating 2000 configurations. The output was of 1017 sections resulted from the space discretization during the calculation process using the MOC. Each section comprises the behavior of the network as pressure values to the occurred perturbation. In the case of the deep neural network based model must be trained on each section. This represents a huge challenge specially for larger irrigation network as the training process of the deep neural network will be very time consuming. To bypass this problem, a technique called autoencoding has been introduced in a next step.

Autoencoders are artificial neural networks that learn to rebuild or copy the input to the output. For an initial sight the task is easy, but it gets its meaning and difficulty by adding some constraints according to the addressed issue. The neural network width is reduced until the so-called bottle neck, then increased to reach the initial number of neurons at the level of the output. The constraint in this case is the size limitation at the bottle neck layer. In other words, autoencoders are forced to learn dense representations of input data, often known as latent representations. As dense representations are of substantially lower dimension compared to the input data, autoencoders are useful for feature extraction and dimensionality reduction (Géron, 2019).

It is important to mention that an autoencoder is a stack of encoder and decoder. The encoder extracts the main features of the input until the latent representation. Subsequently, the decoder tries to rebuild the input starting from the extracted features (laten representation). In the present study an autoencoder was

trained and the encoder was saved extracting the main features representing the different sections of the irrigation network through. An algorithm known as t-SNE was built on the output of the encoder for visualizing data in two-dimensional map as clusters of similar behavior zones. The deep neural network for unsteady state pressure forecasting developed in a previous step will be trained for the resulted  $N_{\text{cluster}}$  number of clusters and not on each section of the network which will reduce considerably the training time. Thus, providing both designers and managers with adequate analysis on the hydraulic behavior of the system under unsteady flow conditions as well as the potential of integrating the developed models for a decision support system for real time decision making.

## ***1.2 Objectives and contributions***

Thus, the objective of the thesis is to introduce a new approach for mapping the perturbation in on-demand pressurized irrigation systems with the aim of building a reliable modernization strategy and to improve pressurized irrigation management. To establish this strategy powerful and flexible tools need to be developed and to be integrated in decision support systems. For this purpose, two models were developed. The first is directly programmed based on the method of characteristics along with introducing two new indicators. The second was build using a deep neural network that was trained on big data generated by the first model. The deep neural network based model may provide the manager with information about the risks of perturbations in the network in real time.

## ***CHAPTER 2. OVERVIEW OF THE NUMERICAL METHOD AND THE ARTIFICIAL NEURAL NETWORK USED FOR THE UNSTEADY STATE FLOW SIMULATION***

### ***2.1 Introduction***

Pressurized irrigation systems working on demand were the object of considerable attention in the sixties and seventies and a considerable number of them were designed and implemented in the Mediterranean basin mainly but also in other parts of the world (Lamaddalena & Sagardoy, 2000). Fluid transient analysis is one of the most challenging and complicated flow problems in the design and the operation of water pipeline systems. Transient control has become an essential requirement for ensuring safe operation of water pipeline systems (Abuiziah, et al., 2013).

Two main aspects are highlighted in this chapter. The first is the hydraulic principle of the unsteady state flow and the development of the equations ruling the phenomenon. Different boundary conditions were considered as well. This part is for the development of the first model based on the method of characteristics and the model is considered directly programmed.

In the second part, historical trend of machine learning and concepts about artificial neural networks related to the present issue were covered. In this case the model is not anymore directly programmed following predefined instructions, but learns by itself the hidden patterns behind the data and build its own algorithm.

## ***2.2 Unsteady state flow conditions hydraulic principles background***

### ***2.2.1 The pressurised irrigation systems flow conditions***

A steady flow is one in which the flow parameters (velocity, pressure) may differ from point to another but do not change with time. If at any point in the fluid, the flow parameters change with time, the flow is described as unsteady. In practice, there is always slight variations in velocity and pressure, but if the average values are constant, the flow is considered steady.

Due to different causes, the flow regime in a pressurized pipeline system can be suddenly changed and a series of pressure waves propagate along the pipes. Then it gradually depreciates because of energy losses due to friction forces. This transient event is a phenomenon called water hammer.

### ***2.2.2 Causes of transient flow***

The most common causes creating pressure waves propagating through the pressurized irrigation systems are (Bergant, et al., 2008):

- Pump start up or shut down, especially stop due to power failure;
- Valves operation;
- Air pockets in pipelines, especially at pump start;
- Changes in boundary pressures (e.g., losing overhead storage tank, adjustments in the water level at reservoirs, pressure changes in tanks, etc.);
- Rapid changes in demand conditions (e.g., hydrant flushing);
- Changes in transmission conditions (e.g., main break).

### ***2.2.3 Factors affecting the transient flow severity***

The magnitude of the pressure waves depends on many factors, mainly summarized in (Bergant, et al., 2008):

- The initial flow velocity;

- Pipelines length (the longer the pipeline the stronger the hydraulic transients is);
- The geometry of the network and the topography of the area to be irrigated;
- The change of flow rate (The more rapidly the flow changes, the higher are the generated pressure waves amplitude);
- The elastic properties of both water and pipes.

## **2.2.4 Developing the unsteady state flow equations**

### **2.2.4.1 Established assumptions**

Possible mechanisms that may significantly affect pressure wave forms include unsteady friction, cavitation, a number of fluid–structure interaction effects, viscoelastic behavior of the pipe-wall material, leakages and blockages. These are usually not included in standard water hammer software packages and are often “hidden” in practical systems (Bergant, et al., 2008).

The following assumptions have been considered to develop the software code:

- Flow in the pipeline is considered to be one-dimensional with mean velocity and pressure values at each section (the velocity and the pressure; at a given instant time vary only in the direction of the flow and not across the cross-section);
- Unsteady friction losses are approximated to be equal to the losses for the steady-state losses;
- Pipes are full of water during all the transient flow and no water column separation phenomenon occurs;
- Free gas content of the liquid is small such that the wave speed can be regarded as a constant;
- The elasticity of the pipe wall and the liquid behave linearly;
- Structure-induced pressure changes are small compared to the water hammer pressure wave in the liquid.



### 2.2.4.2 Euler equation

The Euler equation is determined by applying Newton's second law to a small cylindrical control volume of fluid near the pipe centerline as illustrated in the Fig. 1. Disregard differences in fluid or flow parameters throughout the cross section, the resulting equation will apply to one-dimensional flow along the pipeline (Larock, et al., 1999).

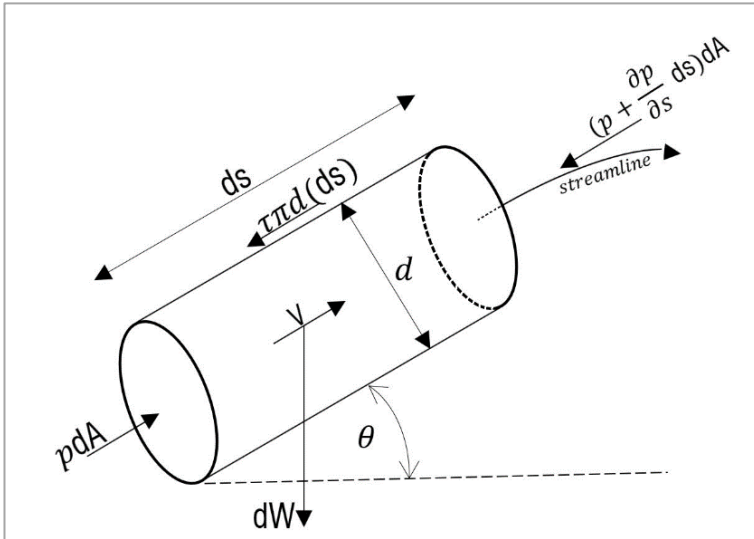


Fig. 1 - A cylindrical fluid element illustrating the different applied forces

Applying the Newton's second law along the streamline direction results:

$$\sum F_s = ma_s = m \frac{dV}{dt} \quad (2.1)$$

m: is the mass of the fluid in the cylindrical fluid element.

The term  $\frac{dV}{dt}$  is the total derivative of the fluid mean velocity. Projecting the different applied forces and representing the mass in terms of density and volume, Eq. 2.1 results:

$$pdA - (p + \frac{\partial p}{\partial s} ds)dA - dW \sin\theta - \tau\pi d(ds) = \frac{dW}{g} \cdot \frac{dV}{dt} \quad (2.2)$$

p: pressure head ( $N/m^2$ );

A: the area of the pipe's cross section ( $m^2$ );

s: distance along the pipe (m);

g: gravitational acceleration ( $m/s^2$ );

dW: unit weight (N);

$\theta$ : angle between the pipe and the horizontal plane;

$\tau$ : wall shear stress ( $N/m^2$ ).

Writing the wall shear stress as a function of mean velocity V, pipe diameter D, fluid density  $\rho$ , and viscosity  $\mu$ , and the equivalent sand-grain roughness r. Simplifications have been proceeded to eliminate the wall shear stress and write it with the form of frictional head loss. The new form of Euler equation of motion has been obtained

$$\frac{dV}{dt} + \frac{1}{\rho} \frac{\partial p}{\partial s} + g \frac{dz}{ds} + \frac{f}{2D} V|V| = 0 \quad (2.3)$$

z: elevation of pipe centerline (m);

f : Darcy-Weisbach friction factor;

D: pipe diameter (m).

### 2.2.4.3 Conservation of mass

As illustrated in Fig. 2, the mass conservation principle is applied on a controlled volume of length ds. The result of this application is as following:

$$\rho AV - [\rho AV + \frac{\partial}{\partial s}(\rho AV)ds] = \frac{\partial}{\partial s}(\rho A ds) \quad (2.4)$$

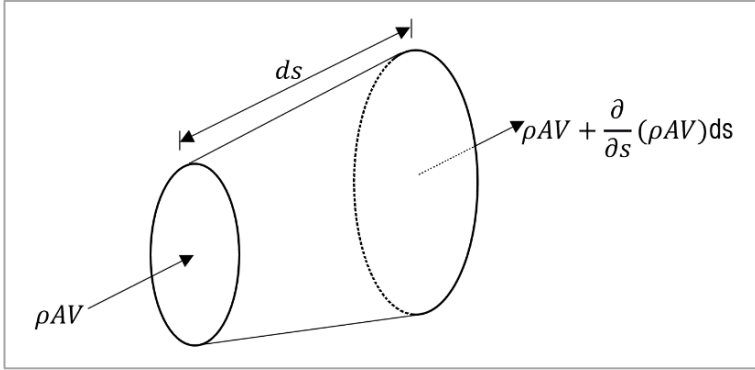


Fig. 2 - Applying the mass conservation principle on a control volume

Expanding the parentheses of Eq. 2.4 and equations for wave speed through thin-walled pipes with the different possible cases were considered (detailed explanation is available in (Ghidaoui, 2004)). The final form of the mass conservation equation is:

$$\frac{1}{\rho} \frac{dp}{dt} + a^2 \frac{\partial V}{\partial s} = 0 \quad (2.5)$$

(Wylie, et al., 1993) shows that the equation for wave speed can be conveniently expressed in the general form

$$\alpha = \sqrt{\frac{\frac{E_v}{\rho_v}}{1 + \frac{E_v D}{E_p e}}} \quad (2.6)$$

a: the wave speed (m/s);

$E_p$ : Young's elastic modulus ( $N/m^2$ );

$E_v$ : liquid elastic modulus ( $N/m^2$ );

$e$ : thickness of pipe wall ( $mm$ );

$\rho$ : fluid density ( $kg/m^3$ );

$D$ : pipe diameter ( $mm$ ).

Many numerical methods were developed to solve the set of partial differential equations (2.3 and 2.5), in which the method of characteristics.

#### **2.2.4.4 THE METHOD OF CHARACTERISTICS**

Throughout the history of water hammer analysis, several ingenious and effective ways for solving the Euler and mass conservation equations have been developed. Those approaches reflected the sophistication of numerical analytic skills at the time, as well as the practitioners' inventiveness. The advent of low-cost, high-performance desktop computers in recent years has resulted in the development of numerically highly accurate solution techniques for these equations capable of including a wide range of boundary and starting conditions. The method of characteristics is now the most general and extensively used methodology for solving these problems (Afshar & Rohani, 2008; Ghidaoui, et al., 2005). It's no surprise that this strategy is particularly compatible with numerical solutions provided by a digital computer. As a result, the method of characteristics is considered in this study.

It is worth mentioning that software like CEBELMAIL, SURGE, AFT Impulse or HAMMER use the method of characteristics as calculation engine.

The characteristic method makes it possible to replace the two partial differential equations 2.3 and 2.5 with a set of ordinary differential equations (Wichowski, 2006). The development of the method starts by presuming that the pair of Eq. 2.3 and 2.5 may be replaced by some linear combination of themselves. Using a Lagrange multiplier  $\lambda$  as a constant linear scale factor.

$$\lambda \left( \frac{dV}{dt} + \frac{1}{\rho} \frac{\partial p}{\partial s} + g \frac{dz}{ds} + \frac{f}{2D} V|V| \right) + \left( a^2 \frac{\partial V}{\partial s} + \frac{1}{\rho} \frac{dp}{dt} \right) = 0 \quad (2.7)$$

Splitting  $\frac{dV}{dt}$  and  $\frac{dp}{dt}$  into their component parts, and regrouping the terms, the equation gives:

$$\begin{aligned} \left( \lambda \frac{\partial V}{\partial t} + \lambda V \frac{\partial V}{\partial s} \right) + \frac{\lambda}{\rho} \frac{\partial p}{\partial s} + \lambda g \frac{dz}{ds} + \lambda \frac{f}{2D} V|V| + a^2 \frac{\partial V}{\partial s} + \left( \frac{1}{\rho} \frac{\partial p}{\partial t} \right. \\ \left. + \frac{V}{\rho} \frac{\partial p}{\partial s} \right) = 0 \end{aligned} \quad (2.8)$$

$$\begin{aligned} \left[ \lambda \frac{\partial V}{\partial t} + (\lambda V + a^2) \frac{\partial V}{\partial s} \right] + \left[ \frac{1}{\rho} \frac{\partial p}{\partial t} + \left( \frac{\lambda}{\rho} + \frac{V}{\rho} \right) \frac{\partial p}{\partial s} \right] + \lambda g \frac{dz}{ds} \\ + \lambda \frac{f}{2D} V|V| = 0 \end{aligned} \quad (2.9)$$

Notice that,

$$\left[ \lambda \frac{\partial V}{\partial t} + (\lambda V + a^2) \frac{\partial V}{\partial s} \right] = \lambda \frac{dV}{dt} \quad \text{if} \quad \lambda \frac{ds}{dt} = \lambda V + a^2 \quad (2.10)$$

And

$$\left[ \frac{1}{\rho} \frac{\partial p}{\partial t} + \left( \frac{\lambda}{\rho} + \frac{V}{\rho} \right) \frac{\partial p}{\partial s} \right] = \frac{1}{\rho} \frac{dp}{dt} \quad \text{if} \quad \frac{1}{\rho} \frac{ds}{dt} = \frac{\lambda}{\rho} + \frac{V}{\rho} \quad (2.11)$$

Thus, we require for  $\frac{ds}{dt}$  that

$$\frac{ds}{dt} = V + \frac{a^2}{\lambda} \quad \text{and} \quad \frac{ds}{dt} = V + \lambda \quad (2.12)$$

Equating these two expressions to eliminate  $\frac{ds}{dt}$  and then solving for  $\lambda$  leads to:

$$\lambda = \pm a \quad (2.13)$$

With  $\lambda$  again equals the wave speed, we find that the equations for the characteristics are:

$$\frac{ds}{dt} = V + a \quad \text{and} \quad \frac{ds}{dt} = V - a \quad (2.14)$$

The resulting equations will be expressed in terms of piezometric head  $H(m)$  using  $p = \rho g(H - z)$ . These equations are typically described in different hydraulic textbooks discussing the water hammer phenomenon (Chaudhry, 1979; Mambretti, 2013).

It is important to mention that the slope of the characteristic curves on the space-time planes is a function of  $V(s, t)$ . This is introduced in the numerical solution procedure as explained hereafter.

Finally, the final set of equations is as following

$$C^+ : \frac{dV}{dt} + \frac{g}{a} \frac{dH}{dt} - \frac{g}{a} V \frac{dz}{ds} + \frac{f}{2D} V|V| = 0 \quad \text{only when} \quad \frac{ds}{dt} = V + a \quad (2.15)$$

$$C^- : \frac{dV}{dt} - \frac{g}{a} \frac{dH}{dt} + \frac{g}{a} V \frac{dz}{ds} + \frac{f}{2D} V|V| = 0 \quad \text{only when} \quad \frac{ds}{dt} = V - a \quad (2.16)$$

The equations  $\frac{ds}{dt} = V + a$  and  $\frac{ds}{dt} = V - a$  are the characteristics of Eq. 2.15 and 2.16, respectively.

The integration of  $(\frac{ds}{dt} = V + a)$  gives  $(t = \frac{1}{V+a} * s + constant)$  that is represented by the curve  $C^+$ . Similarly, for  $(\frac{ds}{dt} = V - a)$ ,  $(t = -\frac{1}{a-V} * s + constant)$  is determined and represented by the curve  $C^-$ , Fig. 3.

The characteristic curves can be approximated to straight lines over each single  $\Delta t$  interval. In fact, (i)  $\Delta t$  may be made as small as one wishes, and (ii) usually  $a \gg V$ , causing  $\frac{ds}{dt}$  to be nearly constant (Wylie, et al., 1993; Chaudhry, 2014). We seek to find the values of  $V$  and  $H$  at the point  $P_n$ . They are calculated basing on  $V$  and  $H$  at the points  $C$ ,  $L$  and  $R$  of the previous time following the characteristic curves  $C^+$  and  $C^-$ .  $V$  and  $H$  at  $P_n$  become the known values for the subsequent time calculation, Fig. 3.

For this situation, the finite difference approximations to Eqs. 2.15 and 2.16 become

$$C^+ : \frac{V_{Pn} - V_L}{\Delta t} + \frac{g}{a} \frac{H_{Pn} - H_L}{\Delta t} - \frac{g}{a} V_L \frac{dz}{ds} + \frac{f}{2D} V_L |V_L| = 0 \quad (2.17)$$

$$C^- : \frac{V_{Pn} - V_R}{\Delta t} - \frac{g}{a} \frac{H_{Pn} - H_R}{\Delta t} + \frac{g}{a} V_R \frac{dz}{ds} + \frac{f}{2D} V_R |V_R| = 0 \quad (2.18)$$

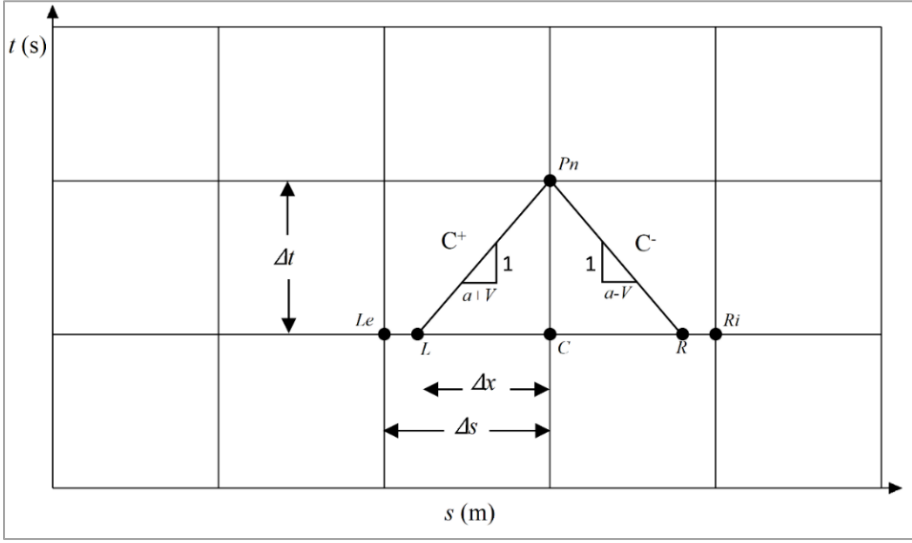


Fig. 3 - Interpolation of H and V values on the  $\Delta s$ - $\Delta t$  grid

The last two equations include six unknown terms  $V_{Pn}$ ,  $H_{Pn}$ ,  $V_L$ ,  $H_L$ ,  $V_R$  and  $H_R$ . In the earlier time, values of H and V are known only at the points C, Le and Ri. Using linear interpolation as shown in Fig. 3,  $V_L$ ,  $H_L$ ,  $V_R$  and  $H_R$  are to be expressed as function of  $V_C$ ,  $H_C$ ,  $V_{Le}$ ,  $H_{Le}$ ,  $V_{Ri}$  and  $H_{Ri}$ .

More in detail, along the  $C^+$  characteristic we assume:

$$\frac{\Delta x}{\Delta s} = \frac{V_L - V_C}{V_{Le} - V_C} = \frac{H_L - H_C}{H_{Le} - H_C} \quad (2.19)$$

solving the above equations for  $V_L$  and  $H_L$ , we obtain

$$V_L = V_C + a \frac{\Delta t}{\Delta s} (V_{Le} - V_C) \quad (2.20)$$

$$H_L = H_C + a \frac{\Delta t}{\Delta s} (H_{Le} - H_C) \quad (2.21)$$



An analogous approach can be applied along the  $C^-$  characteristic.

$$V_R = V_C + a \frac{\Delta t}{\Delta S} (V_{Ri} - V_C) \quad (2.22)$$

$$H_R = H_C + a \frac{\Delta t}{\Delta S} (H_{Ri} - H_C) \quad (2.23)$$

This leads to solve Eqs. 2.17 and 2.18 simultaneously for  $V_{Pn}$  and  $H_{Pn}$  as following:

$$V_{Pn} = \frac{1}{2} [(V_L + V_R) + \frac{g}{a} (H_L - H_R) + \frac{g}{a} \Delta t (V_L - V_R) \sin \theta - \frac{f \Delta t}{2D} (V_L |V_L| + V_R |V_R|)]$$

$$H_{Pn} = \frac{1}{2} [\frac{a}{g} (V_L - V_R) + (H_L + H_R) + \Delta t (V_L + V_R) \sin \theta - \frac{a f \Delta t}{g 2D} (V_L |V_L| - V_R |V_R|)]$$

Usually, the slope term ( $\frac{dz}{ds} = \sin \theta$ ) is small and may be neglected (Chaudhry, 2014).

The complexity of irrigation systems is the non-uniformity of pipe materials and pipe sizes, which requires a pipe discretization where each elementary section has constant geometrical and physical properties. Each elementary section is divided into an integer number of elements  $NS_i$ , with length  $\Delta S_i$ , whose value is calculated, to have the same  $\Delta t$  in all the system (Lamaddalena & Pereira, 2007).

A steady-state simulation is executed for each configuration of hydrants simultaneously operating. The obtained results (H and V) will constitute the initial conditions for running the transient simulation. The computer code calculates the water hammer process until the simulation time reaches a predefined observation time

( $T_{max}$ ), generally assumed large enough to reach again the new steady-state flow conditions.

#### **2.2.4.5 Boundary conditions equations**

The boundary conditions at each end of the pipes describe externally imposed conditions on  $V$  and  $H$ . The application of the differential equations assumes the boundary conditions described hereafter. The variables  $V$  and  $H$  are indexed with  $P_i$  corresponding to the points, one on each side of the boundary section, which are nearly superposed as it is shown in Fig. 4. For all the other parameters, only the number of the pipes is used as an index to prevent any complication in naming. In both cases of upstream and downstream end boundaries of the systems, only one point exists following  $C^-$  and  $C^+$ , respectively.

##### **2.2.4.5.1 At the reservoir level (upstream end of the network)**

If a reservoir with constant pressure head  $H_0$  is located upstream the network, then:

$$H_{P_1} = H_0 \quad (2.26)$$

By substitution into Eq. 2.18,  $V_{P_1}$  is obtained.

$$V_{P_1} = V_2 + \frac{g}{a}(H_0 - H_2) - \frac{f\Delta t}{2D}V_2|V_2| \quad (2.27)$$

In which  $V_2$  and  $H_2$  are the mean velocity and piezometric head at the  $\Delta s$  downstream the reservoir and  $\Delta t$  earlier time.

##### **2.2.4.5.2 At valves level**

Being located at the downstream end of the pipes, the valve closure is assumed to induce a linear flow velocity variation at the cross-section according to the following equation:

$$V_{P_1} = V_0 * \left(1 - \frac{t}{T_c}\right) \quad (2.28)$$

where  $V_0$  (m/s) is the initial flow velocity and  $T_c$  (s) is the valve closure.

$$H_{P_1} = H_2 - \frac{a}{g}(V_{P_1} - H_2) - \frac{a f \Delta t}{g 2D} V_2 |V_2| \quad (2.29)$$

In which  $V_2$  and  $H_2$  are the mean velocity and piezometric head at the  $\Delta s$  upstream the valve and  $\Delta t$  earlier time.

For any value of  $V_{P_{N+1}}$ , including zero.

### 2.2.4.5.3 Internal boundary conditions

Junctions with two and three pipes are considered:

- Two-pipe junction:

A two-pipe junction is shown in Fig. 2 (a).

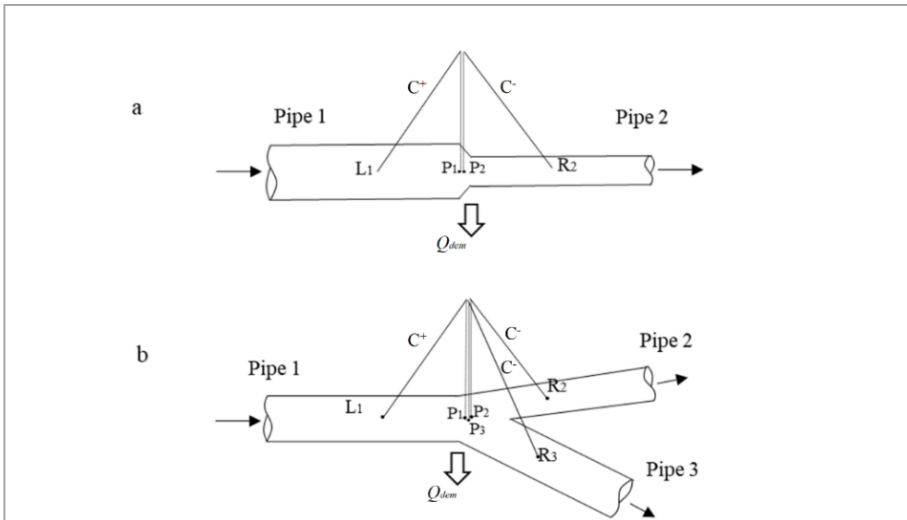


Fig. 4 - Boundary conditions at a typical series pipe junction

In the case of no external demand, the values of the four unknowns can be found by solving the set of equations below:

- following  $C^+$ :

$$V_{P_1} = (V_{L_1} + \frac{g}{a_1} H_{L_1} - \frac{f_1 \Delta t}{2D_1} V_{L_1} |V_{L_1}|) - (\frac{g}{a_1}) H_{P_1} \quad (2.30)$$

- following  $C^-$ :

$$V_{P_2} = (V_{R_2} - \frac{g}{a_2} H_{R_2} - \frac{f_2 \Delta t}{2D_2} V_{R_2} |V_{R_2}|) + (\frac{g}{a_2}) H_{P_2} \quad (2.31)$$

- the conservation of mass equation:

$$V_{P_1} A_1 = V_{P_2} A_2 \quad (2.32)$$

- the energy equation at the points  $P_1$  and  $P_2$ , neglecting the difference in velocity heads and any local losses:

$$H_{P_1} = H_{P_2} \quad (2.33)$$

Solving the above system of equations, the head value  $H$  at the junction can be calculated as follows:

$$H_{P_1} = H_{P_2} = \frac{C_3 A_1 - C_1 A_2}{C_2 A_2 + C_4 A_1} \quad (2.34)$$

$C_1$ ,  $C_2$ ,  $C_3$  and  $C_4$  are function of the known values obtained from the earlier time. By means of back-substitution, also the flow velocity can be found.

In the case of a series of two pipes with an external constant demand  $Q_{dem}$  ( $m^3/s$ ) (that is delivered by one hydrant), a similar system of equations can be used, modifying only Eq. 2.32 as follows:

$$V_{P_1} A_1 = V_{P_2} A_2 + Q_{dem} \quad (2.35)$$

that makes Eq. 2.34 to become:

$$H_{P_1} = H_{P_2} = \frac{C_3A_1 - C_1A_2 - Q_{dem}}{C_2A_2 + C_4A_1} \quad (2.36)$$

- Three- pipe junction

A three-pipe junction is shown in Fig. 2 (b).

In the case of a pipe junction with one inflow and two outflows, the following equations are used to find the six unknowns:

$$\text{Pipe 1, } C^+: \quad V_{P_1} = C_1 - C_2H_{P_1} \quad (2.37)$$

$$\text{Pipe 2, } C^-: \quad V_{P_2} = C_3 + C_4H_{P_2} \quad (2.38)$$

$$\text{Pipe 3, } C^-: \quad V_{P_3} = C_5 + C_6H_{P_3} \quad (2.39)$$

$$\text{Conservation of mass:} \quad V_{P_1}A_1 = V_{P_2}A_2 + V_{P_3}A_3 \quad (2.40)$$

$$\text{Energy balance, neglecting local losses: } H_{P_1} = H_{P_2} = H_{P_3} \quad (2.41)$$

Solving the previous set of equations leads to

$$H_{P_1} = H_{P_2} = H_{P_3} = \frac{C_1A_1 - C_3A_2 - C_5A_3}{C_2A_1 + C_4A_2 + C_6A_3} \quad (2.42)$$

In the case of a three-pipe junction with an outlet, in the previous set of equations, only the Eq. 2.40 has to be modified as follows:

$$V_{P_1}A_1 = V_{P_2}A_2 + V_{P_3}A_3 + Q_{dem} \quad (2.43)$$

while the Eq. 2.42 becomes:

$$H_{P_1} = H_{P_2} = H_{P_3} = \frac{C_1A_1 - C_3A_2 - C_5A_3 - Q_{dem}}{C_2A_1 + C_4A_2 + C_6A_3} \quad (2.44)$$

## **2.3 Artificial neural networks background**

### **2.3.1 Introduction to machine learning**

As a part of artificial intelligence, it is worth to mention that machine learning (ML) is the study of algorithms that automatically learn through an experience. As defined by a computer scientist “A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.” (Mitchell, 1997). The experience can be gotten from an available data known as "training data" in supervised learning or as feedback during the learning process like in the case of reinforcement learning. The aim of automated learning is to make predictions or decisions without even being explicitly programmed to do. Practically, it turns out to help the machine following different approaches to develop its own algorithm through the analysis of the available data (Jin, 2020).

According to the criteria taken into consideration, many approaches of machine learning are available. If we consider feedback as a criterion for division, mainly we get three categories. Supervised, unsupervised and reinforcement learning (Reddy, et al., 2018).

### **2.3.2 Supervised and unsupervised Learning Algorithms**

Supervised learning algorithms are algorithms that learn from given input and the associated output to develop a way to produce more correct predictions for new (unseen) data. This prediction should be with the minimum possible error. Supervised learning was used to develop the model for pressure classes forecasting.

In the other hand, unsupervised learning algorithms are those that deal only with input and there is no expected output. There is no objective test for distinguishing whether a value is a feature or a target provided by a supervisor. The great example is the one of clustering data into groups having similar behaviour that was used for the model based on autoencoding.

### **2.3.3 *Deep neural networks***

Artificial Neural Networks (ANN) are a subfield of machine learning. It is a computational representation of the human brain made up of a series of layers of artificial neurons as shown in Fig. 5. For a closer idea, it is a mathematical function created by combining multiple smaller functions. The core of deep learning is the availability of powerful computers and large data to train deep neural networks. For this reason, deep learning models have grown in size over time with the main advantage of learning more complex information. With the term deep, we mean the number of layers is higher than the early used neural networks. Fig. 6 illustrates the comparison between older learning algorithms and deep learning performance with respect of data amount.

It is important to consider the architecture determination that refers to the structure of the neural network. It consists of the number of layers, the number of neurons for each layer and how the layers are connected to each other. These architectural considerations are known as the depth and the width of the layers, respectively. The best architecture selection may be reached via experimentation oriented by the performance of the model on the validation set.

The data is fed to the neural network through the input layer and being processed along the hidden layers. The output of the model is presented at the level of the so-called output layer.

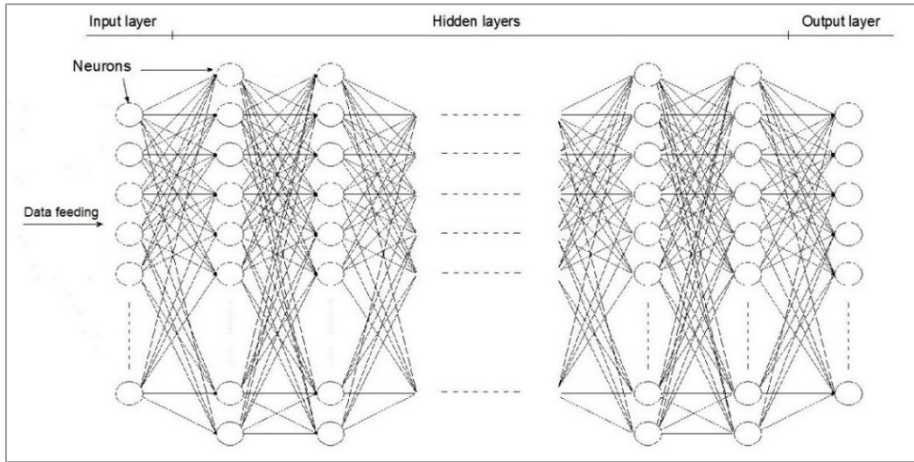


Fig. 5 - Multilayer perceptron neural network

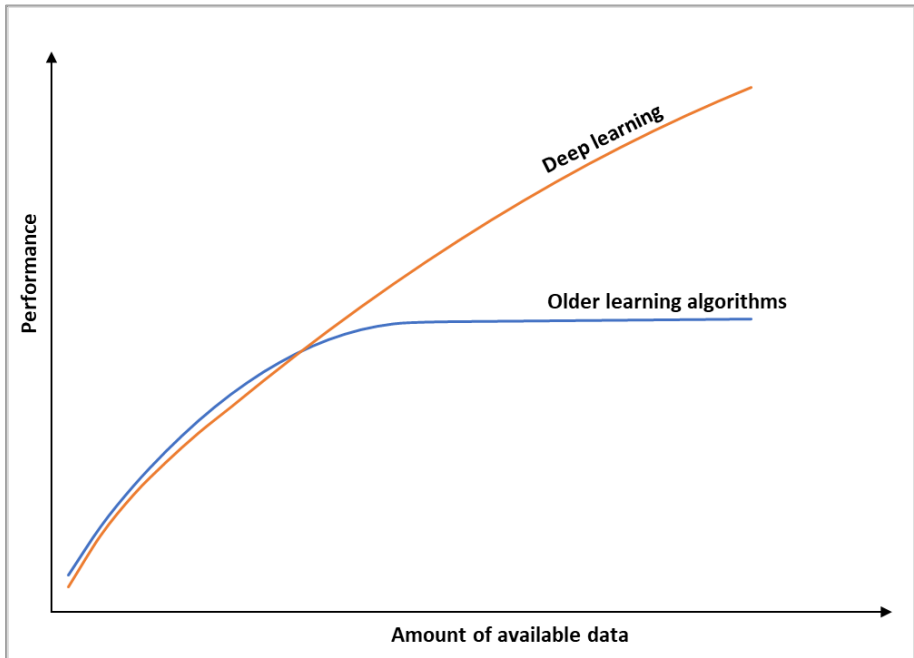


Fig. 6 - Performance of the two techniques with the amount of data

Many books and articles cut through mathematical talk around machine learning algorithms and explain detailed concepts (Subramanian, s.d.; Yan, 2021). In this work, we will focus on the application of the appropriate methods and techniques



to reach our objectives. In the studied problem, machine learning is great for these two reasons:

- The current executions need a lot of accurate tuning or lengthy series of rules, in which the machine learning algorithm seeks simplifying the code to perform better than the conventional approach.
- Getting clear idea about complex issues that deal with big data.

#### **2.3.4 Learning process**

Building a deep neural network involves two main set of components, hyperparameters and parameters. Hyperparameters are elements that are set before training the model and that are not to be updated during the learning phase such as the number of layers, activation function, the batch size and the number of epochs. In the other hand, parameters are the elements that get updated as a part of the model learning such as weights.

Training a neural network is an iterative process progressing with updating weights that improves the model's forecasting accuracy for each iteration. It is performed through a prefixed number of iterations called epochs that is the number of passages through all the dataset. As a part of the repetitive process, the error in the forecast made by the model respect to the real pressure class value in the current state is to be estimated using a loss function. Based on the estimated loss the weights are updated to reduce the error for the next evaluations.

During the training phase, the model focuses on finding the best combination or set of weights to make the model forecasting performance as high as possible for the studied problem. Weights are parameters within the neural network that converts input data through the hidden layers. The inputs get multiplied by the assigned weights and added to a bias (Eq. 2.45) then passed through an activation function (Eq. 2.46) to the next layer as input or observed if it is the last layer (Fig. 7).

$$S_1^L = \sum W_{1,i}^L X_i^{L-1} + b \quad (2.45)$$

$$X_1^L = f(S_1^L) \tag{2.46}$$

- $W_{1,i}^L$  : weighs from the nodes  $i$  of the previous layer (L-1) to the node 1 of the layer L.
- $X_i^{L-1}$  : outputs of the nodes  $i$  from the previous layer (L-1).
- $b$ : the bias parameter of the affine transformation. In the absence of any input, the transformation's output is based to  $b$ .
- $S_1^L$  : the output of the transformation that is the input for the activation function ( $f$ ).
- $X_1^L$  : the output of the node 1 of the layer L.

### **2.3.4.1 Activation function**

Activation functions play an important role by adding non-linearity into neural networks, permitting the neural networks to learn powerful operations and find out complicated patterns. The selected activation function for the different layers is ReLu (Rectified Linear Units), while for the output layer is Softmax to represent the output as probabilities. ReLu is defined as following:

$$f(S) = \max \{0, S\} \tag{2.47}$$

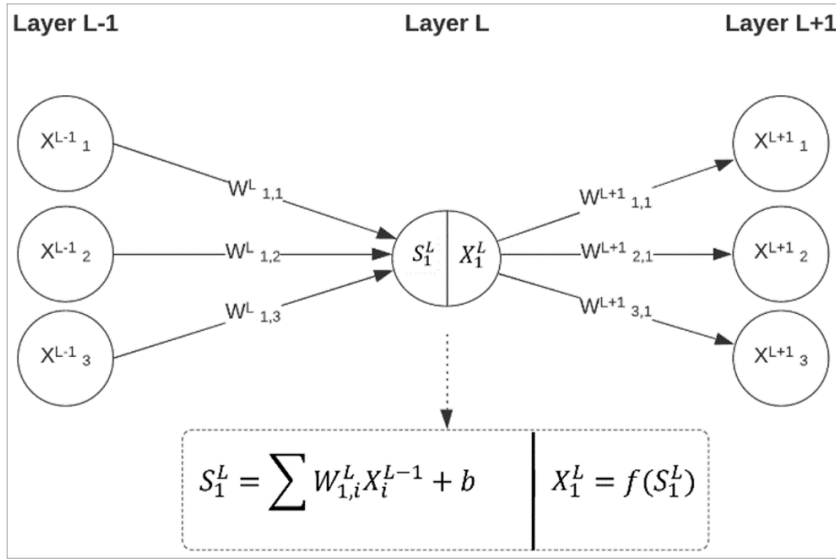


Fig. 7 - A scheme mapping input to output for an artificial neuron

The Softmax activation function calculates the relative probabilities using the values from the output layer  $x$  to determine the final probability value.

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j^N \exp(x_j)} \quad (2.48)$$

$N$  is the total number of output layer neurons.

### 2.3.4.2 Loss function

The function used to assess the associated weights in the context of an optimization method is referred to as the objective function. We may strive to maximize or decrease the objective function, which means we are looking for a set of weights with the greatest or lowest score. Typically, while using neural networks, we want to reduce error. As a result, the objective function is also known as a cost function or a loss function, and the value computed by the loss function is simply referred to as loss.

There are various loss functions to select from, it can be challenging to identify which one to use. According to the studied problem, the selection of the loss function may become easier. For the model using deep neural networks for pressure classes

forecasting, the problem is of multi-class classification type. A logarithmic loss function called “categorical\_crossentropy” was selected in this case. This latter is defined as following:

$$L(y - \hat{y}) = \sum_i^N y_i * \log \hat{y}_i \quad (2.49)$$

In which  $\hat{y}$  is the predicted value and  $y$  is the expected output,  $\hat{y}_i$  is the  $i^{th}$  value in the neural network output and  $y_i$  is the corresponding target value.  $N$  is the number of the model’s output values.

### 2.3.4.3 Optimization function

The process of the losses’ minimization is elaborated through the called optimization function. This is elaborated by adapting the attributes of the neural network such weights and learning rate to reduce the losses. For this reason, it is essential to choose the appropriate optimization function.

Adam, for adaptive moments, is an adaptive learning rate optimization algorithm that was selected in this study is one of the most famous and used optimization functions (Kingma & Ba, 2014). Some parameters are to be selected first as following:

- $S_s$  for the step size, suggested equal to 0.001;
- $E_1$  and  $E_2$  for exponential decay rates for moment estimates that they vary in the interval  $[0, 1)$ , in the present study are suggested equal to 0.9 and 0.999 respectively;
- $N_s$  for numerical stabilization constant, suggested equal to  $10^{-8}$ ;
- Initialize 1<sup>st</sup> and 2<sup>nd</sup> moment variables ( $s, r$ ) and the time step ( $t$ ) to 0.

The optimization function starts by computing the gradient  $g$  as following:

$$g = \frac{1}{Nm} \nabla_{\theta} \sum_i^{Nm} L(f(x^i; \theta), y^i) \quad (2.50)$$

In which  $Nm$  is the number of examples in a minibatch from the training set,  $L$  is the loss function,  $f(x^i; \theta)$  is the predicted output of the input  $x^i$ ,  $y^i$  is the corresponding target output, and finally  $\theta$  that is an initial parameter getting updated as explained here after.

After that the biased first and second moments get updated each time  $t$ :

$$s = E_1 * s + (1 - E_1)g \quad (2.51)$$

$$r = E_2 * r + (1 - E_2)g \odot g \quad (2.52)$$

Both moments get updated is a second step:

$$\hat{s} = \frac{s}{1 - E_1^t} \quad (2.53)$$

$$\hat{r} = \frac{r}{1 - E_2^t} \quad (2.54)$$

In a final step the update value gets calculated and updated as following:

$$\Delta\theta = -Ss \frac{\hat{s}}{\sqrt{\hat{r}} + Ns} \quad (2.55)$$

$$\theta = \theta + \Delta\theta \quad (2.56)$$

The algorithm keeps repeating the above process until a stopping selected criterion.

### 2.3.5 Autoencoder module

Autoencoders are neural networks that learn to build an output to be similar to the input with the smallest possible error. Autoencoders are unsupervised learning algorithms, more precisely, they are considered to be self-supervised because they create their own labels from the training data. Mainly, autoencoders compress the input data into a lower-dimension layer (called code or latent space), afterward recreate the output from the compressed layer. The compression and recreation phases are respectively the encoder and the decoder. The committed error between the original and the reconstructed data is controlled by a loss function. It is worth to mention that autoencoders are lossy because the output is never built exactly as the input, in fact, the output is close but degraded. Various dimensionality reduction algorithms exist, the big advantage of using autoencoders is capacity of handling large datasets (Goodfellow, et al., 2016).

The output of the encoder is the vector  $r$  that is the representation of the input  $x$  (Fig. 8). This first part can be presented by the encoder function  $f$  in which  $r = f(x)$ . The second part seeks to rebuild the input  $x'$  from the vector  $r$  represented by the decoder function  $g$  in which  $x' = g(r)$ . The learning process of the autoencoder is basically minimizing the loss function  $L(x, g(f(x)))$  that penalizes  $g(f(x))$  for not being similar to the input  $x$ .

As autoencoders are simply artificial neural networks, the learning process is as mentioned previously. The selection of the different functions may vary according to the case.

The autoencoder tries to rebuild its own input that are pressure values. The most convenient loss function is the mean squared error (MSE). It is derived by averaging the squared differences between predicted and real values. The outcome is always positive, regardless of the sign of the predicted and real values. The loss value is to be reduced, thus, it may be employed in a maximal optimization process by setting the score to a negative number. The loss function is defined as following:

$$L(y - \hat{y}) = \frac{1}{N} \sum_{i=0}^N (y_i - \hat{y}_i)^2 \quad (2.57)$$

As mentioned earlier,  $\hat{y}$  is the predicted value and  $y$  is the expected output,  $\hat{y}_i$  is the  $i^{th}$  value in the neural network output and  $y_i$  is the corresponding target value.  $N$  is the number of the model's output values.

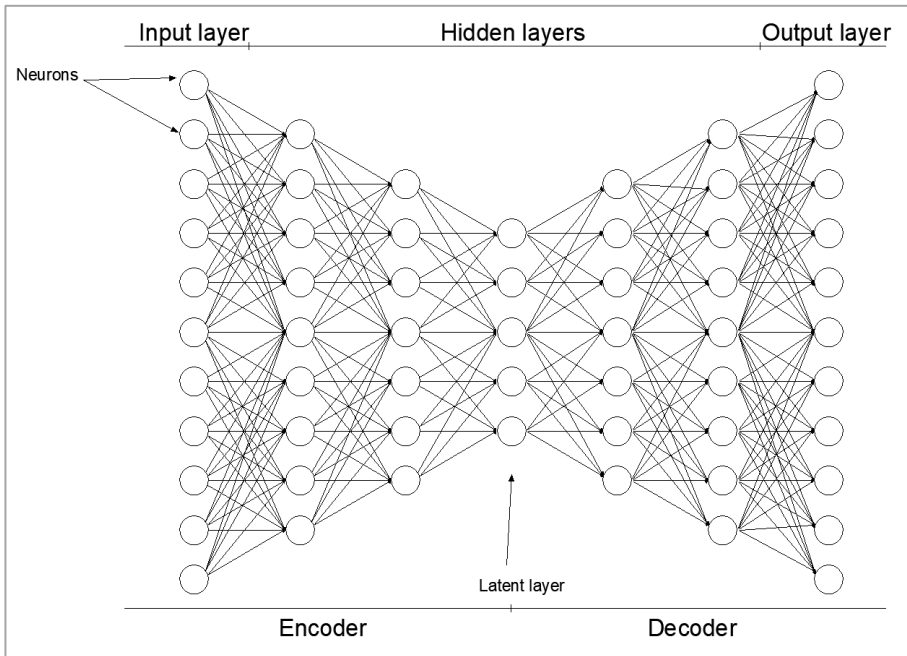


Fig. 8 - Undercomplete autoencoder architecture

### 2.3.6 *t*-distributed Stochastic Neighbor Embedding (*t*-SNE)

*t*-SNE is an unsupervised, randomized algorithm that applies non-linear dimensionality reduction technique focusing on keeping similar data points close together in lower dimensional space (2 or 3D). In the present study, the data is reduced to a reasonable level using the encoder saved from the previous step. Afterwards, *t*-SNE is used for 2D visualization. It is great at conserving as much relevant information

as possible from the available high-dimensional data, as well as increasing the interpretability of the data in the lower dimension. As mentioned before, t-SNE could be used directly for dimensionality reduction but for large datasets it is advantageous to use an encoder first.

t-SNE algorithm consists of two parts. Firstly, the algorithm starts by converting the high-dimensional Euclidean distances between data points into conditional probabilities. In other words, finding the pairwise similarity between nearby points in the high dimensional dataset. The similarity of the points  $x_j$  (neighbours) to the point  $x_i$  is calculated through the conditional probability  $p_{j|i}$ , whereas, similar  $x_j$  to  $x_i$  is, higher is  $p_{j|i}$ . For  $i \neq j$  the conditional probability is defined as following:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)} \quad (2.58)$$

Whereas,  $\sigma_i$  is the Gaussian variance that is centred on the point  $x_i$ ,  $p_{i|i} = 0$  and  $\sum_j p_{j|i} = 1$  for all  $i$ .

To symmetrize the conditional probabilities the average of the two probabilities is considered as following:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N_h} \quad (2.59)$$

$N_h$  is the number of elements in the high-dimensional dataset.

For the second part, Student t-distribution is used with one freedom degree as the heavy-tailed distribution in the low-dimensional map. As a result, the joint probabilities  $q_{ij}$  are defined as following:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}} \quad (2.60)$$



t-SNE seeks a low-dimensional data representation minimizing the mismatch between  $p_{ij}$  and  $q_{ij}$ . The joint probabilities  $p_{ij}$  and  $q_{ij}$  are equal if the points  $y_i$  and  $y_j$  perfectly model the similarity between the points  $x_i$  and  $x_j$ .

For measuring the accuracy of similarity modelling, the gradient of the Kullback-Leibler divergence between P and Q is used (P and Q are respectively the conditional probability distribution and the Student-t based joint probability distribution over all other datapoints). The gradient  $\frac{\delta C}{\delta y}$  is given by:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1} \quad (2.61)$$

C is the cost function.

$$y^{(t)} = y^{(t-1)} + \eta \frac{\delta C}{\delta y} + \alpha(t)(y^{(t-1)} - y^{(t-2)}) \quad (2.62)$$

Where  $y^{(t)}$  and  $\alpha(t)$  are respectively is the solution and the momentum at the iteration t and  $\eta$  is the learning rate.

## **CHAPTER 3. METHODOLOGY**

### **3.1 Study area**

The developed model was tested on a network selected from Sinistra Ofanto irrigation scheme in the province of Foggia that is managed by the Consortium of Capitanata in the Puglia Region - Italy. The scheme is divided into 7 districts (numbered from 4 to 10) which are subdivided into sectors. The area is irrigated by reservoirs supplied from the Capacciotti dam through a conveyance pipe as shown Fig. 9. The irrigation sectors are ramified distribution networks and serve the farmers through outlets (called hydrants), mostly designed for a minimum pressure head of 20 m and a discharge of 10 l/s (Lamaddalena, et al., 2004). The study was performed for the sector 25 in district 4 that is an on-demand drip irrigation system. Both district 4 and the sector 25 are illustrated in Fig. 10 and 11 respectively.

The cropping patterns are similar amongst the different irrigation districts (Vineyards 63.4 %, olive trees 20.3 %, orchards 3.6 % and horticulture 12.7 %). The climate in this area is semi-arid to sub-humid and defined as 'Maritime-Mediterranean' with an annual average precipitation of about 500 mm which is poorly distributed (Er-Rami, et al., 2021).

The sector consists of 19 hydrants served by gravity from a reservoir with a piezometric elevation of 128 m a.s.l. The nominal pressure of all network pipes is equal to 10 bars (the maximum pressure tolerable by the pipes). The conveyance conduits are steel pipes while the sectors are equipped with PVC pipes.

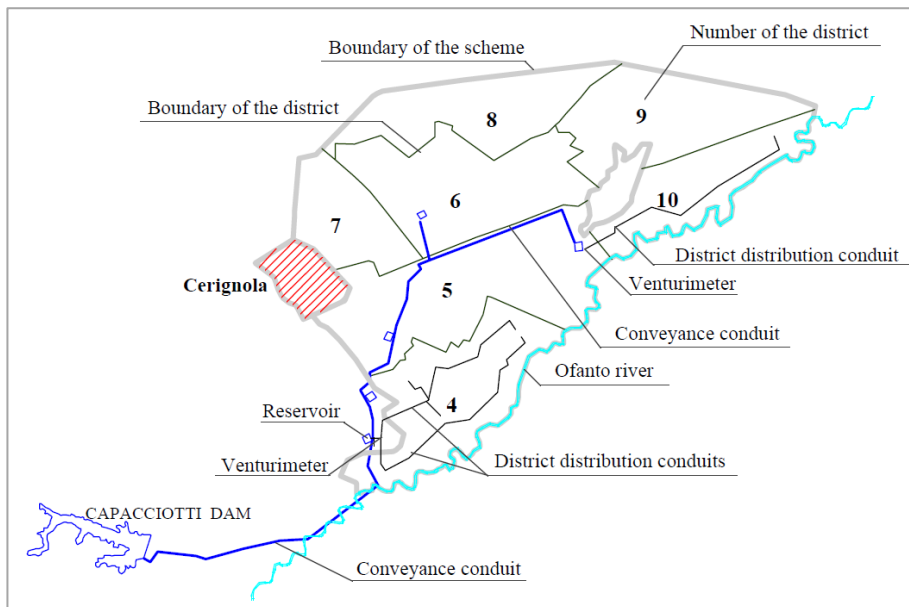


Fig. 9 - The "Sinistra Ofanto" irrigation scheme

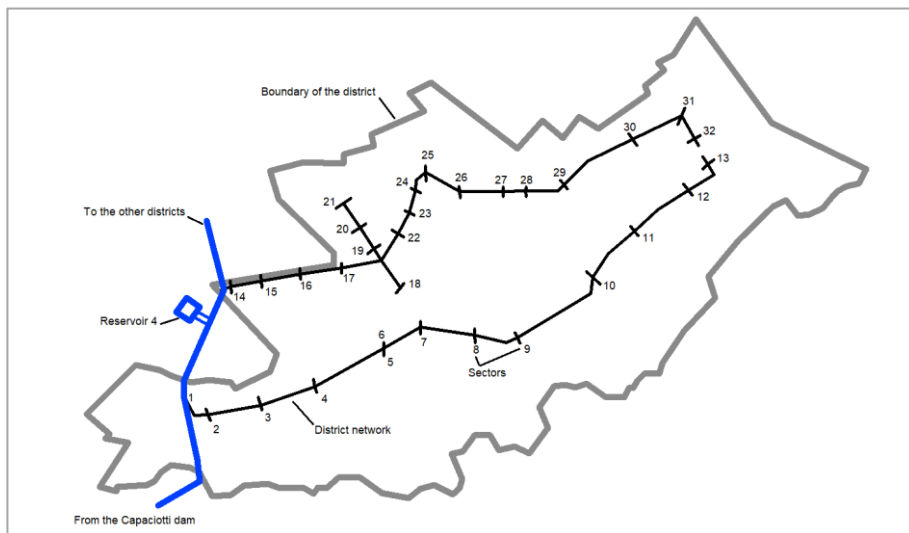


Fig. 10 - Layout of the district 4 network

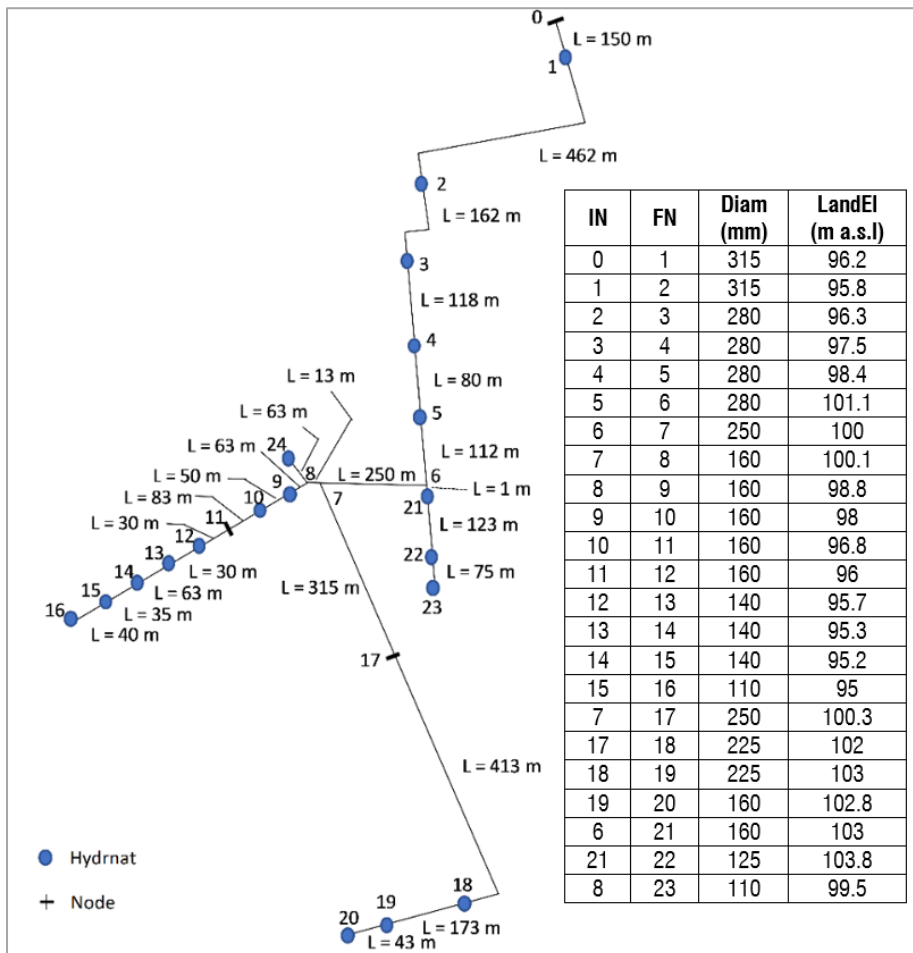


Fig. 11 - Layout of the network (sector 25)

IN is the initial node of each section, FN is the final node, LandEI is the land elevation and Diam is the pipes diameter. The land elevation is referred to the final node.

### 3.2 The MOC based model

#### 3.2.1 The calculation process

In a first step, the model generates the set number of configurations. It provides the possibility to set different possible scenarios including the various functioning

modes for the hydrant as well as the valves closure arrangement. The generated configurations are uniformly distributed for the representability purposes.

In order to define the initial conditions for the unsteady flow analysis, a steady state simulation was executed for each configuration. The piezometric elevation (H) and the velocity (V) are defined for each section of the pressurized irrigation system.

The perturbation is generated through the hydrants' manipulation. Starting with the pre-computed H and V from the steady state conditions, calculations of the new values  $H_{P_n}$  and  $V_{P_n}$  are carried out for each grid point with an increment of  $\Delta t$ . Therefore, new values of H and V are obtained and are to replace the previous ones. The process continues for the pre-fixed simulation time. The software selects the maximum and the minimum pressure occurred at each section through the time of simulation (selection through time). The output at this level is an array of  $N_c$  configurations rows and  $X_p$  maximum and minimum pressure values for each grid node (section). This output will be used for training the deep neural network model.

A second selection through the pipe sections for  $P_{max}$  and  $P_{min}$  is performed (selection through space). The analysis results are tabulated as maximum and minimum pressure head occurred for each pipe that will be the basis of the calculation of one of the introduced indicators (Hydrant Risk Indicator, HRI).

As it was mentioned before, in this study  $T_{max}$  has been chosen equal to 30 (sec), for which the variation of propagated pressure waves magnitude is no more significant. For larger network the time of simulation may be higher, the same may happen in the case of systems of certain geo-planimetric conditions. The calculation process is summarized in the flow chart illustrated in Fig. 12.

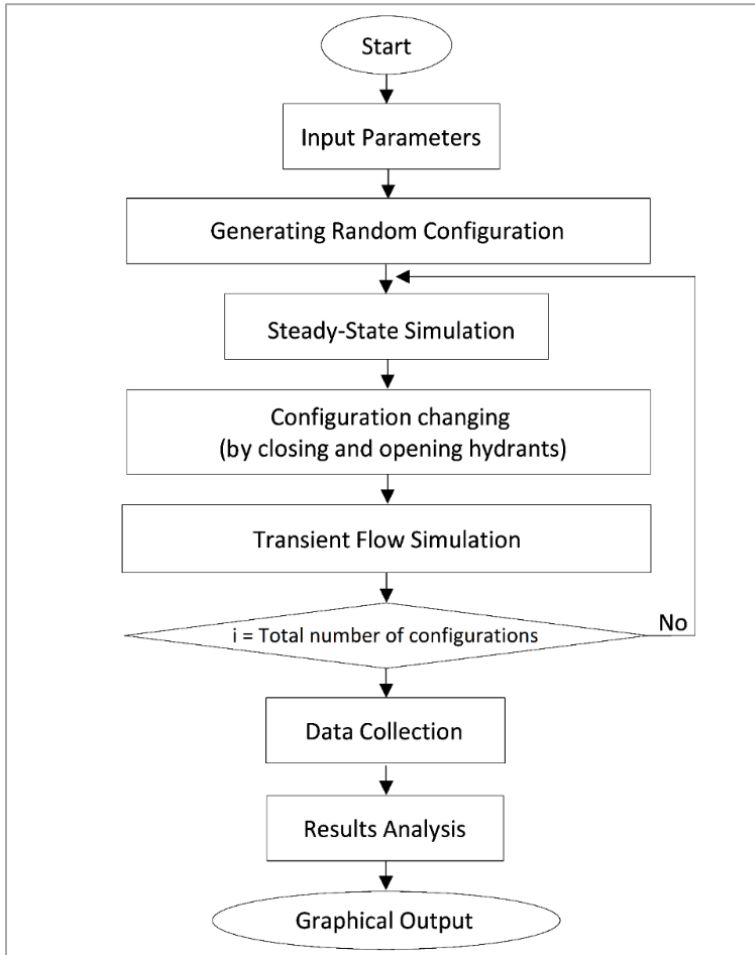


Fig. 12 - Computer Code Flowchart.

### 3.2.2 The Hydrant Risk Indicator (HRI)

By analyzing the impacts of different configurations on the hydraulic behavior of the irrigation system, the Hydrant Risk Indicator evaluates the sensitivity of the network in terms of pressure to each hydrant manipulation. It is defined as the ratio between the participation of the hydrant in the riskiest configurations and its total number of participations.

A chosen percentage of the riskiest configurations will yield an upper and lower pressure envelope. The upper envelope represents the maximum pressure magnitude

recorded through all the pipes, while the lower envelope represents the minimum recorded pressure magnitude values.

HRI reflects the potential risk created by each hydrant. Hydrants significantly impacting the overall performance in terms of pressure are expected to appear more frequently in the risky configurations.

$$HRI = \frac{RPN}{TPN} \quad (3.1)$$

TPN is the Total Participation Number, and RPN is the Risky Participation Number. A hydrant will be considered when it is being closed or opened, which is the main reason of the perturbation.

Knowing that the maximum and the minimum pressures are separately treated and presented

- $HRI_{P_{min}}$  indicates the ability of each hydrant to create a negative wave ( $P_{min}$ ), (RPN takes into consideration only the opening mode).
- and  $HRI_{P_{max}}$  indicates the ability of each hydrant to create a positive wave ( $P_{max}$ ), (RPN takes into consideration the closing mode).

It is worth mentioning that the total number of configurations has to be taken to ensure that the indicator achieves the stabilization stage as explained hereafter.

### **3.2.3 Relative pressure exceedance (RPE)**

It numerically represents the pressure variation and the created risk with respect to the nominal pressure. The RPE was introduced to help both the designer and manager analyze the irrigation systems operating on-demand and illustrate the weak points of the system where any pipe damage may occur. RPE is defined as following:

$$RPE = 100 * \frac{P_{max} - NP}{NP} \quad (3.2)$$

RPE is the relative pressure exceedance (%);  $P_{\max}$  (bar), the maximum pressure recorded throughout the simulation time at each section; NP (bar), the nominal pressure. In the present study.

The RPE is presented as 10% equiprobability curves, where each curve represents a probability of occurrence.

### ***3.3 The deep neural network based model***

#### ***3.3.1 The functioning principle of the pressure classes forecasting model***

Assessing the perturbation in pressurized irrigation networks with directly programmed software is time consuming, in such situation, integrating the software for real-time decision-making is not possible. Therefore, a novel framework is proposed to tremendously reduce the time consumption and warrant being integrated as a part for mapping the perturbation in a whole real-time decision-making system for best irrigation networks management.

The work was performed through two main steps. The first was elaborated by slightly modifying the MOC based model developed in (Lamaddalena, et al., 2018) and (Derardja, et al., 2019) to get the output and the enclosed information as an input for the new model. This input is presented as a table of which columns represent the different features. Features include each hydrant functioning mode (closing, opening, closed and opened), the valve closing time and the hydrant closing arrangement type (the table is illustrated in Fig. 13 as the input of the deep neural network). The combination of the mentioned features is referred as a configuration. Along with each configuration, the output of the MOC based model will be fed as an expected output from the deep learning model to investigate the correlation existence between the features and real output (deep learning training phase). The MOC based model output is the maximum recorded pressure at each section of the irrigation network. In a further step, the pressure is converted into classes of 1, 2 or 3 bars step instead of decimal values.



The different rows of the input table represent the number of simulated configurations. Some data types can be categorical (for example the valve closing arrangement) thus specific encoding should be taken into consideration (giving a code for each mode).

The second step is to build and train a deep learning model. It learns paths and relations between the input and the output of the MOC based model. The model learns to assess the behavior of irrigation networks to the perturbation occurrence without being directly programmed.

It is worth to mention that models developed using deep learning, with the previously mentioned input and characteristics are valid only (tailored) for the studied irrigation system. In other words, for a new system, the model should be trained on the system’s data that are specific to it.

Several programming languages are suitable as an environment for developing deep learning models in which each language offers a stronghold on a specific concept. As it is one of the most popular and the fastest growing programming languages (Srinath, 2017), Python was used for this study.

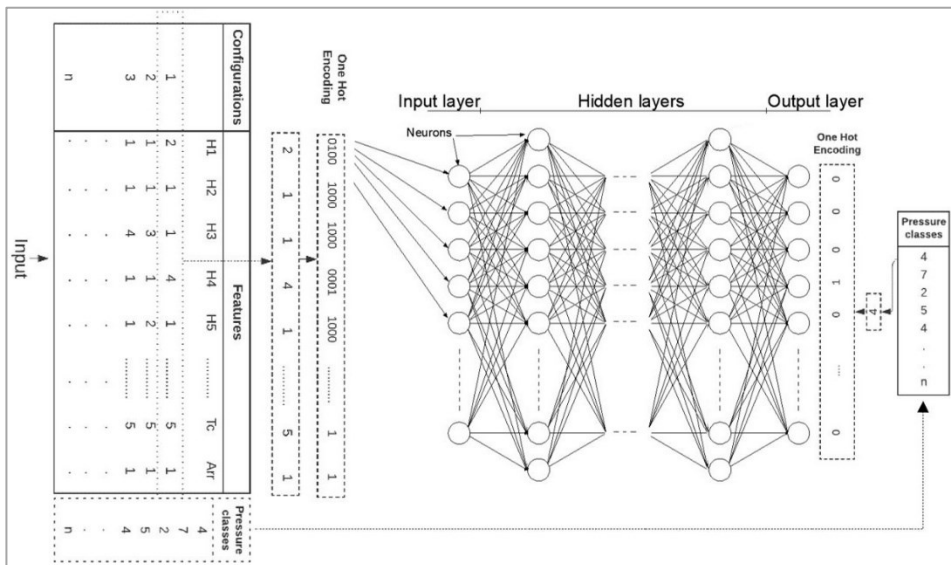


Fig. 13 - Multilayer perceptron neural network.

The previously mentioned steps are illustrated by the flow chart in Fig. 14. After getting trained, the new model forecasts instantaneously the pressure classes for new possible scenarios, called configurations.

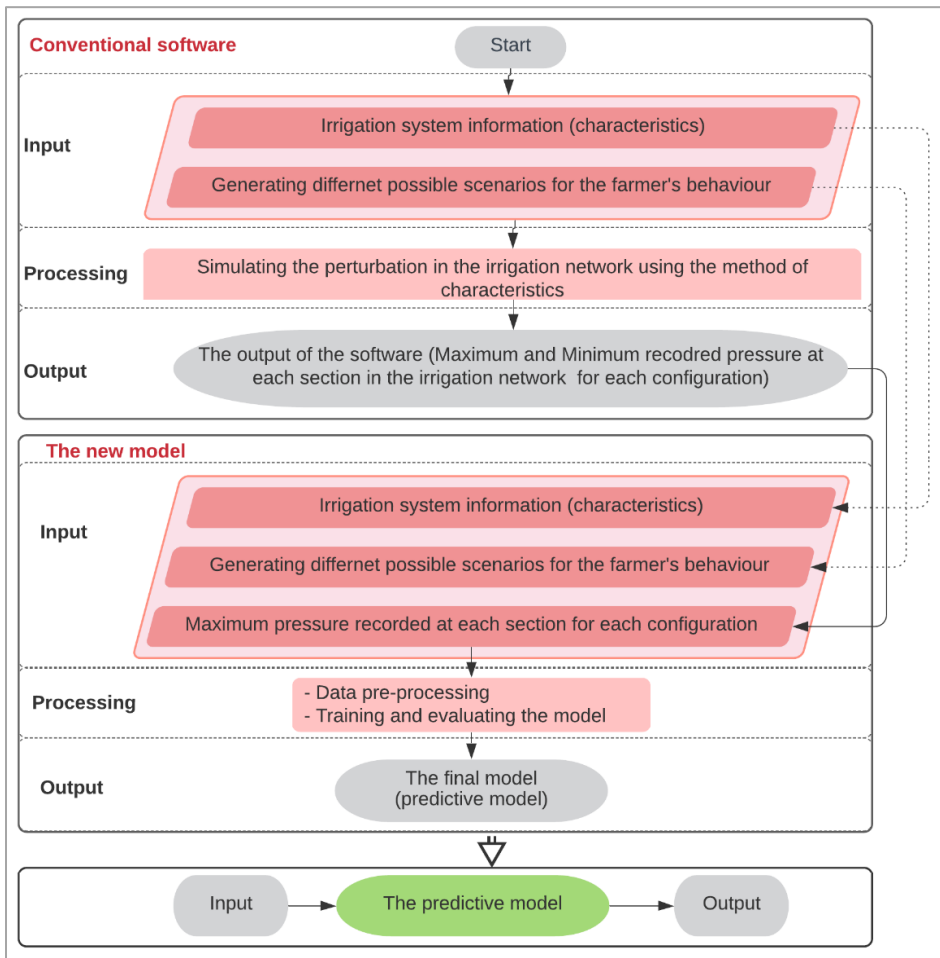


Fig. 14 - Flow chart of the model phases

### **3.3.2 *The model's architecture***

A multilayer perceptron network (MLP) was used in the present study. The performance of the model is affected by many parameters such as the architecture of the neural network, in which the number of layers and neurons of each layer, the activation, the loss and the optimization functions.

The model is fed through an input layer, whereas the number of its neurons is the number of the data features after being encoded (using one-hot-encoding). In the present case, the considered features are the hydrants functioning modes (4 modes), the closing time (considered equal for all hydrants) and the closing arrangement type (one type was chosen). The result of the encoding phase that is the number of neurons in the input layer is 78 (19 hydrants \* 4 functioning modes + 1 closing time + 1 closing arrangement), Fig. 13. The output later contains a number of neurons equal to the number classes covering the reached pressure that depends on the chosen step. In the simulated 2000 configurations, the pressure ranges from 4 to more than 15 bars, if a step of 2 bars is considered, the number of pressure classes will be 6 to cover this range. Regarding the hidden layers, the number of neurons is varying between 150 to 500 neurons.

Increasing the number of layers or neurons by layer increases the adaptivity so the model can learn more detailed information. But it can lead in some cases to the called over-fitting, means the model learns well from the training set but turns to not be capable to generalize on new data (test set). In fact, the choice depends on the complexity of the problem itself, the two variables are set by trial and error aiming to get the best performance (Goodfellow, et al., 2016). The latter mentioned process is known as the hyperparameter tuning.

### **3.3.3 *The clustering model***

In the framework of introducing machine learning into the pressurized irrigation systems perturbation assessment, the deep neural network based model developed in the previous step for maximum pressure forecasting must be trained on each section of the irrigation network. The sections length, in the order of 3 meters, is determined in

the data preparation phase using a directly programmed software based on the MOC. The selected irrigation network (sector 25) is of about 3000 meters, thus, 1017 sections. Training the previously mentioned model on such number is not practical and requires a long time at the training phase. Therefore, extracting the called main features or the representative sections of the irrigation network is the solution key for this issue. The different sections will be presented as clusters of similar behavior. In other words, sections belonging to the same cluster have a similar behavior to the occurred perturbation.

The core part of this step of the work aims to develop a model for features extraction of the irrigation networks' different sections. The data input to be analyzed include the pressure at all the network sections, that are 3 meters distanced, each have  $N_{\text{conf}}$  pressure values ( $N_{\text{conf}}$  is the number of configuration). Going through the different configurations, the model extracts the main features and presents sections of similar behaviour to the occurred perturbation as clusters.

Before training the autoencoder, it is necessary to establish a set of hyperparameters, mainly four. Firstly, the latent layer size that is the number of neurons in the middle layer, smaller size results more compression of the input. The depth of the neural network or the number of layers. The width that is the number of neurons by layer maintaining the reduction of the neurons number until the latent space then increasing the number until the output layer. The autoencoder architecture generally is symmetric respect to the latent layer but is not a requirement, whereas the output layer should be as the same size as the input one. Lastly, the loss function that leads the update of the different parameters during the learning phase.

The model is built up of an encoder (the saved first part of a trained autoencoder) with a t-SNE algorithm for 2D visualisation. It is important to mention that the model developed in this step is to be introduced in between the first two ones developed earlier in this thesis as shown in Fig. 15. The model developed in the second step will be trained only  $N_{\text{cluster}}$  instead of  $N_{\text{sec}}$ .

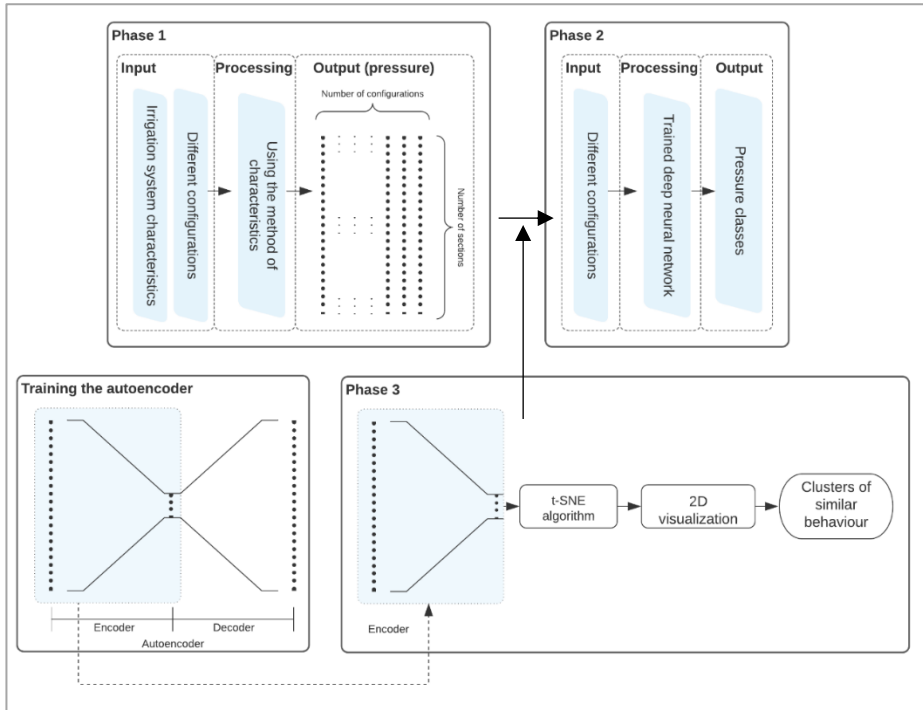


Fig. 15 - Flow chart of the three phases

### 3.4 Data Pre-processing

Almost all data need to be pre-processed before being fed to the deep learning algorithms. Data quality is crucial when it comes to deep learning models performance, thus, different techniques could be used according to various requirements.

#### 3.4.1 One-Hot-Encoding data

The One-Hot-Encoding approach was used to encode the introduced variables in a way that the model does not attempt to identify a link or a relation between them. The variables are supplied in binary form, and the active class is fired (given a value of 1), while the others are set to 0. In the case of hydrants functioning modes, the variable takes the values 1, 2, 3 or 4 in which 1 is represented as [1000], 2 as [0100] etc.

### **3.4.2 *Split into Train and Test Sets***

After the data has been properly fed into the model, it must be divided into training and test sets (as an example, 80 percent of the data is used to train the model, while the remaining 20 percent is used to test it by comparing the model's forecasts to the expected output). The splitting size depends mainly on the size of the original data itself. It is worth noting that this technique is best suited for big datasets with solid evidence that both the training and test sets are representative of the problem; otherwise, the downside of this technique may appear. It is about having the so-called high variance; which indicates that altering the selected training set causes a considerable change in forecasting accuracy.

The class values balance is a crucial information while dealing with classification challenges. For imbalanced classes distribution, the observations appear in some classes more than others. This is a prevalent problem in practice, and it must be addressed appropriately throughout the data preparation process. Referring to the present case study, more observations appear in classes for pressure ranging between 6 and 14 bars.

One of the widely used techniques for handling this challenge is the k-fold cross validation. It splits the original data into k-folds (k parts), trains the algorithm on k-1 folds and then evaluates it on the held back one. This procedure is to be performed k times, each holding back a different fold and it allows each fold to serve as a test set. As a summary of the performance evaluation, this approach reports the mean across the various splits. It results a more accurate estimation of the developed model real performance. The number of folds (k) should be accurately chosen. In the present study, the model was simulated for a list of k values (2 to 13 folds) that will be discussed in more detail in a further section. The higher that k is, the more computing expensive the model becomes.

## **3.5 Performance evaluation**

Deep learning algorithms are evaluated using metrics. It is important to choose them appropriately to get the real model's performance that affects the choices and the importance weight of the different characteristics. In the present work, the developed model for forecasting the pressure classes is of classification problems type, thus the classification accuracy was used. It is the number of correct forecasting as a ratio of all forecasting made.

### **3.5.1 Confusion matrix**

The accuracy is one of the most common evaluation metrics for classification problems, it is also the most misused. The so-called confusion matrix is at the heart of the performance evaluation of multiclass classification algorithms. The confusion matrix is defined as the matrix that contains the ratio of expected to actual class instances. It enables the creation of a wide range of performance measurements for multiclass classification, typically the accuracy (Markoulidakis, et al., 2021). In other words, It provides information about an overall correctly forecasted classes as well as incorrect forecasting with errors being made. Presenting forecasting on the horizontal axis and the real outcomes on the vertical one. The correct forecasting fall on the diagonal of the matrix ( $C_{i,i}$ ) and the incorrectly over and under forecasted values on the right and the left of the diagonal, respectively.

		Predicted Classes			
		C <sub>1</sub>	C <sub>2</sub>	...	C <sub>N</sub>
Actual classes	C <sub>1</sub>	C <sub>1,1</sub>	C <sub>1,2</sub>	...	C <sub>1,N</sub>
	C <sub>2</sub>	C <sub>2,1</sub>	TP	...	C <sub>2,N</sub>
	...	...	...	...	...
	C <sub>N</sub>	C <sub>N,1</sub>	C <sub>N,2</sub>	...	C <sub>N,N</sub>

Fig. 16 – Confusion matrix for multiclass classification problems

### 3.5.2 *Learnig curve*

Along with the training process, the accuracy will be graphed as a function of epochs number, nevertheless the value is associated with the model's final accuracy. If the model is learning well, the graph shows the correct forecast out of the total ones improving over time.

The powerful deep learning algorithms may learn too well until the point of over fitting the training data. It is for this reason such algorithms need to be used carefully. Learning curves are commonly used as a diagnostic tool for incrementally learning models (optimize the internal parameters over time). Training the model and testing it can be evaluated by plotting the accuracy as a function of epochs number (the model passes through all the data in one epoch). After each epoch, the performance of the model is updated for both the training and the testing.

The learning process stops either automatically using for example the called early stopping in keras or by plotting the learning curve versus the number of epochs then selecting the good fit.





## CHAPTER 4. Results and discussion

### 4.1 Results of the MOC based model

#### 4.1.1 The uniformity of the randomly generated configurations

The strong variation of the discharges flowing into the network due to the variation in the demand is the first provoker of the perturbation in the pressurized irrigation systems. That variation is presented through the different configurations. With a view to having a good representation, the software uses a random number generator to run different configurations following a uniform distribution function (having the same possibility of getting one operating mode for each hydrant), see Fig. 17. The reported results refer to the opening/closing of two hydrants, as this situation occurs with higher probability compared to the simultaneous opening/closing of 3, 4 or 5 hydrants, and stronger waves with respect to the opening/closing of one hydrant do occur.

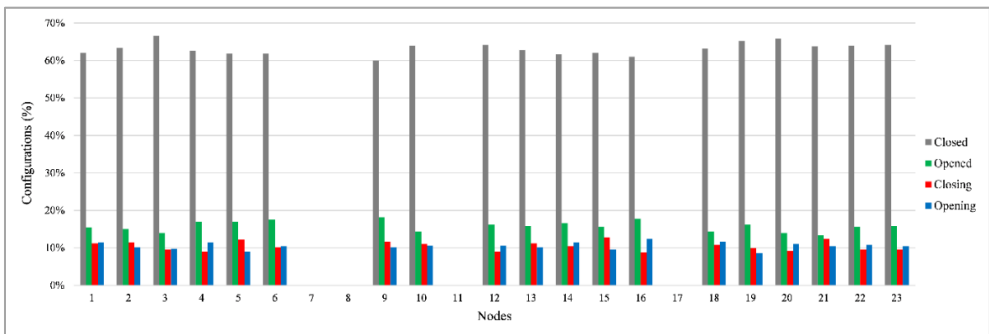


Fig. 17 - Uniformity distribution of the random number generator (500 hydrants configurations).

#### 4.1.2 Comparison of the model's output to the commercial product Hammer

A validated commercial software, named HAMMER (Bentley), was used to simulate 100 configurations. The same configurations were simulated using the developed model during this study. The output of both models are pressure profiles

through all the networks pipes Reservoir to Node 16. The root mean squared error between the output of the two models is illustrated in Fig. 18. The chart shows an increasing error from the upstream end of the irrigation network (reservoir) to the downstream end of the profiles (node 16). This is because of the increasing sensitivity of the network to the occurring perturbation at the level of the downstream pipes, thus, the pressure values presented as envelopes become larger. It is worth to mention that the error at the level of the reservoir is slightly higher than the following sections because of the set boundary conditions parameters between the two models. In addition to the boundary conditions, the output that is not perfectly the same may be affected by other factors such as, the unknown time step used in the commercial model while the present developed one is approximately of 0.003, hydrants were connected through 1 m pipes to the network pipes while is possible to connect them directly to the network in the present developed model.

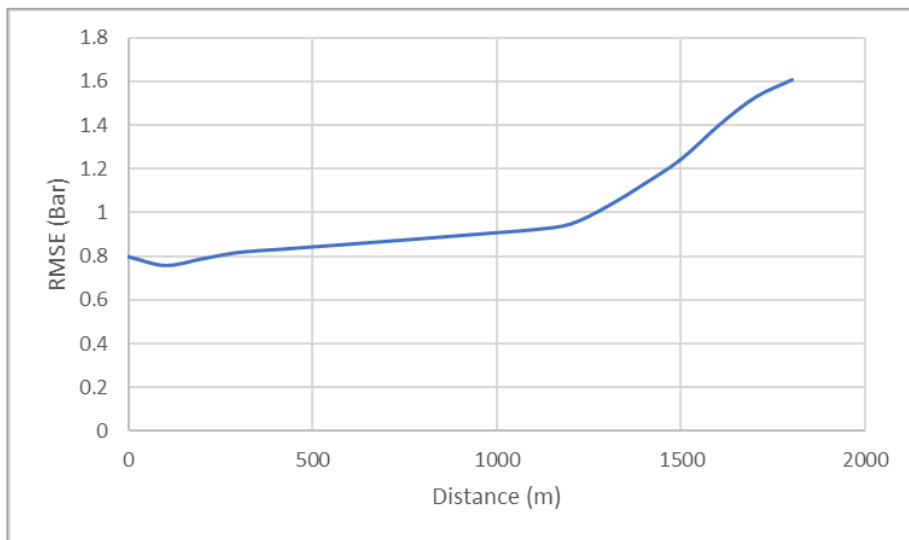


Fig. 18 - Root Mean Square Error along with the profile Res-Node 16

### **4.1.3 Computation of the RPE representation**

In this study, the hydrant closing time  $T_c = 0$  was considered, as it represents the riskiest case that may happen. The irrigation system consists of four layout profiles; only the profile Res-Node16 is presented in Fig. 19. After the perturbation, the maximum pressure waves were recorded along the pipes and presented as 10% equiprobability curves. The maximum pressure variation for the different steady-state conditions at node 16 is around 0.35 bar and increases when moving from the upstream end of the studied profile to downstream.

It is worth mentioning that the code imposes a constraint of not having the water column separation even with the occurrence of low pressure, in line with the assumption mentioned above that the pipes are assumed to be full and remain full during the transient flow occurrence, which enables the application of the differential equations.

RPE provides a very clear idea about the pipes under risk ( $P_{\max}$  influence). The pipes are considered to be safe when RPE values are negative, which means that the maximum occurred pressure does not exceed the nominal pressure. As the value of zero means that the transient pressure is equal to the nominal pressure, from that value onwards the pipes start being under risk.

In Fig. 19, the pipes for the main line (Res-Node 8) are in the safe range (RPE < 0). At the level of the node 8, which is the entrance of the branch "8-16", the RPE starts to take positive values with less than 10 % probability of occurrence. The zone corresponding to the hydrants from 11 to 13 is potentially subject to failure with 10 % probability of occurrence. The more distant the section from the upstream end, the greater the risk of pipe failure. The failure reaches its maximum occurrence probability of 40% (100%-60%) at the downstream end of the layout profiles (hydrant 16).

In parallel with the probability of occurrence and the corresponding zones, it is important to mention the role of the RPE that provides an overview of the exceedance severity.

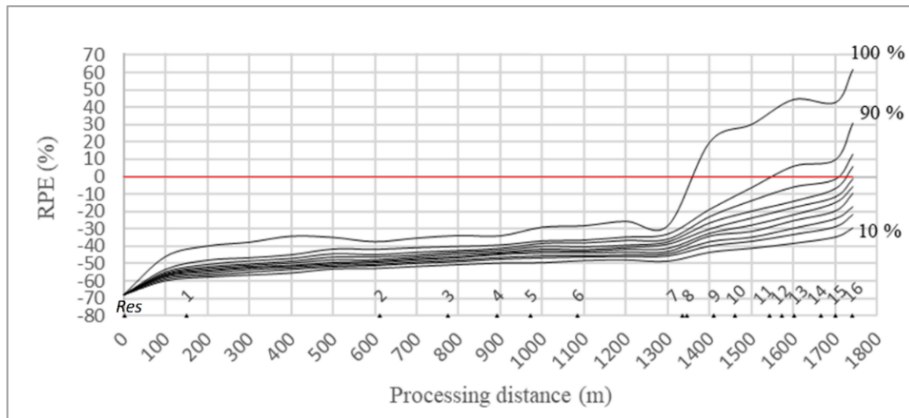


Fig. 19 - Relative Pressure Exceedance 'RPE' for 500 configurations.

#### 4.1.4 Calculation of HRI

To assess the network response to the different perturbations of configurations, 500 different configurations were generated and analyzed. The number of configurations was run not only to satisfy the uniformity test, but also for the stability of the indicator. In fact, the HRI graph starts to take its final shape at around 350 configurations, where only the real risky hydrants will appear. Once the values are stabilized, the increase of the number of configurations does not significantly affect the results. See Fig. 20, 21 and 22.

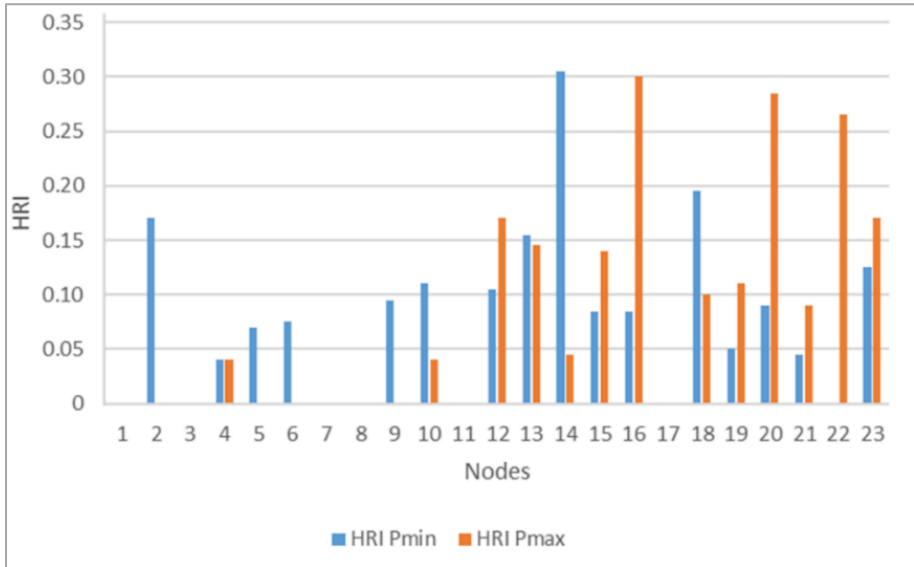


Fig. 20 - Hydrant Risk Indicator 'HRI' for 50 configurations.

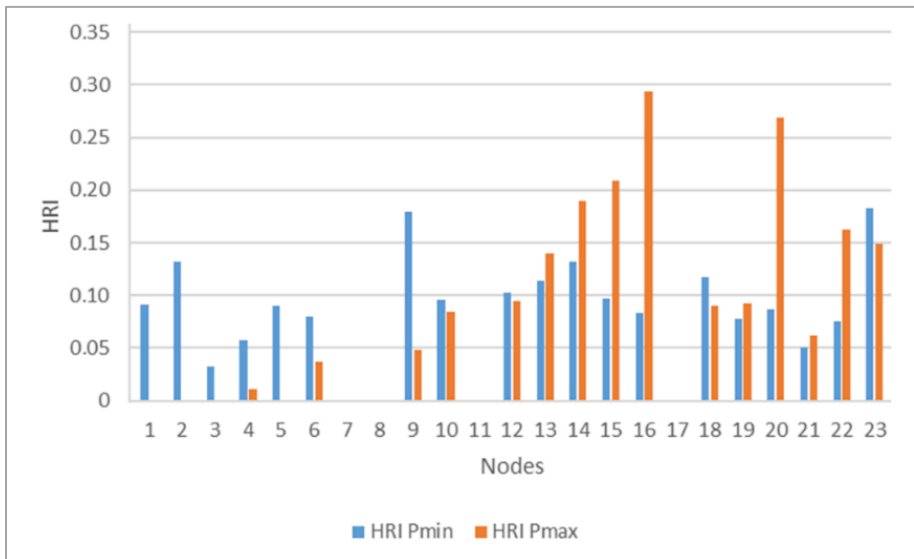


Fig. 21 - Hydrant Risk Indicator 'HRI' for 350 configurations.

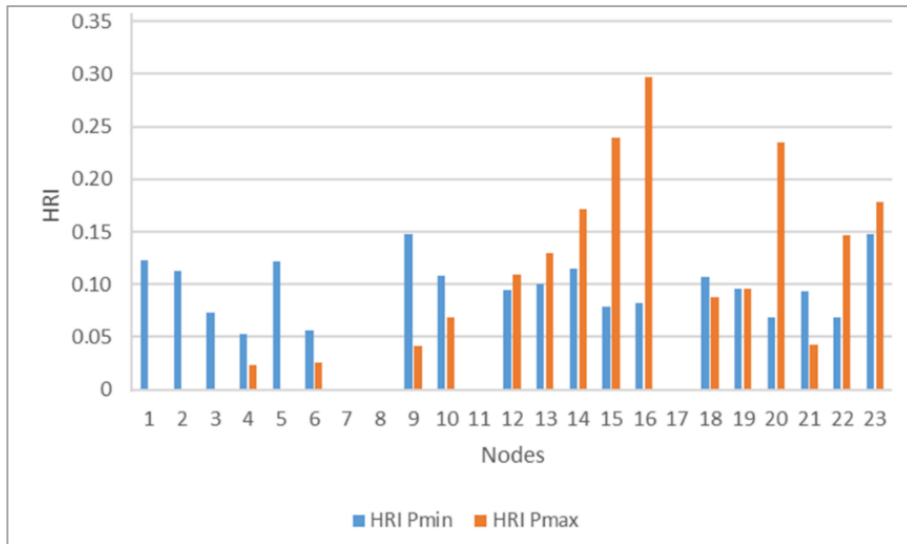


Fig. 22 - Hydrant Risk Indicator 'HRI' for 500 configurations.

By running 500 configurations for the allocated system where all nodes are represented, the contribution of the riskiest hydrants could be easily identified by noticing the extreme irregularities of the generated graphs Fig. 20. In the presented case, the hydrants 14, 15, 16 and 20 could be identified as risky hydrants that could generate positive pressure waves.

The term “Risky Configuration” could be precisely defined, in this case, as the configuration causing exceedance of the allowed domain of variation in pressure set by the manager according to certain criteria, mainly the system infrastructure.

In these extreme cases, “Risky Hydrants” have a high probability to cause either positive or negative waves. The impacts of such cases could cause serious problems.

## 4.2 Results of the deep neural network based model

Using deep learning, a model to forecast the class of the maximum pressure in the irrigation network was developed. The forecasted pressure values are represented as classes. Being the model, accuracy is strongly related to the targeted step, 2 bar

step was chosen, obtaining a model accuracy of 85 %, that is sufficient to provide a clear idea about the risk severity. In fact, the accuracy in this case can be considered even higher that will be explained hereafter with the called confusion matrix. Changing the class step is possible and affects the forecasting accuracy positively if the step is larger.

An Artificial Neural Network (ANN) of 11 layers (1 input, 9 hidden and 1 output layers) was built. The output layer contains a number of neurons equal to the number classes, that depends on the chosen step, covering the reached pressures. Two thousand configurations were simulated and the pressure ranges from 4 to more than 15 bars, so if a 2 bar step is considered, 6 pressure classes are enough to cover this range. For the hidden layers, the number of neurons is between 150 to 500.

The efficient Adam gradient descent optimization algorithm and the logarithmic loss function “categorical crossentropy” were used. The rectified linear unit (ReLU) activation function was selected for hidden layers. Being this work related to a multiclass classification problem, a softmax activation function was used at the level of the output layer.

#### **4.2.1 Average accuracy using k-fold cross validation**

As mentioned earlier, the deep learning algorithms are evaluated using the called metrics. Since dealing with multiclass classification, the accuracy as a performance metrics was set. The study was elaborated through two sections, the first is to have the average accuracy, in which the k-fold cross validation technique was used. Different values of k were run (2 to 13 folds). It is clearly illustrated in Fig. 23 that the average accuracy is increasing with the number of folds until  $k = 5$ , after that, it keeps fluctuating between 84 to 85%. The highest average accuracy was reached for 8 folds.

Having sparse values for accuracy (min and max) is because splitting the data into many folds increases the probability of having a non-representative training and/or the testing examples. On the other hand, calculating the average of the many different



splits increases the overall accuracy by compensation and it is one of the used techniques to avoid the previously mentioned “high variance”.

For an overall performance evaluation, 5-folds splitting is the best choice and the one selected since it is less computationally expensive and with a good accuracy respect to other cases. As selected in the previous step, 5 folds is best representing the study. It means that the model splits the data into 80 and 20 % for training and test sets respectively, and that this procedure is repeated 5 times, holding back a different test set (20%) each time.

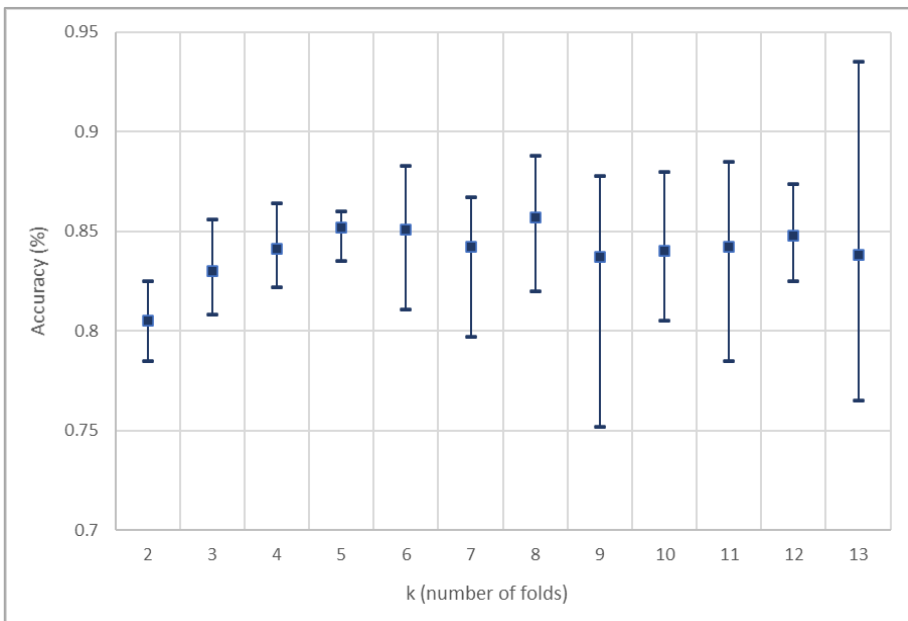


Fig. 23 - Average, minimum and maximum accuracy in function of k cross-validation values.

#### 4.2.2 Confusion matrix

In the present study, the confusion matrix makes results less confusing by breaking down the different forecasting, correct, wrongly over or underestimated class and how far it is from the correct one. Three cases were simulated by changing the pressure class step (1, 2 and 3 bars) and the average accuracy is around 75, 84 and 90 % respectively.

The correct forecasts are on the diagonal (darker color, Fig. 24). Taking the example of 3 bar pressure step confusion matrix, the class 2 on the vertical axis represents the actual or the real class. On the horizontal axis 91 % are correctly forecasted as class 2, 3% as class 1 and 6% as class 3. It is clear that wider the class is, higher the probability for the model to get the correct forecasting. In the opposite side, converging to precise forecasting reduces the accuracy which is the case for 1 bar pressure step.

The class step is selected depending on the sensitivity of the pressurized irrigation system. Choosing a smaller step helps to get a good intuition about the model's real accuracy and its distribution. For instance, a pressure of 10.90 belongs to the class [10:11] and forecasting it as belonging to the class [11:12] does not expose the pressurized pipes to a high risk. A step of 2 bars is considered to be appropriate for irrigation systems.



Fig. 24 - Confusion matrices for 1, 2 and 3 bars steps classification.

In most of practical study cases, data is not uniformly distributed. In the present case study, the pressure reaches values higher than 16 bar only a few times, hence miss predicting has a significant impact on the percentage of overall correct forecasting. The distribution in the test set (real classification) is depicted in Fig. 25. As previously stated, the first and last classes contain less data than the other.

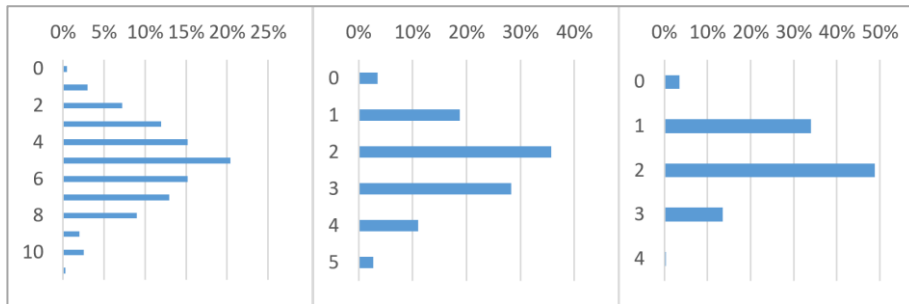


Fig. 25 - The data distribution in the test set (real classes).

The results of the model may vary slightly due to the stochastic nature of the used algorithms, the evaluation procedure and/or differences in numerical precision. The model should be run several times for comparing the average outcome.

### 4.2.3 Learning process

In this section, we seek evaluating the model's behavior during the learning process. Thus, three cases are presented here after (for 1, 2 and 3 bar class step). It shows experience on the horizontal axis and learning development on the vertical axis. The used metric (accuracy) to evaluate model learning should be maximized. The higher the accuracy, the better the model learns. If reached, a value of 1 indicates that the training dataset was learned perfectly and no mistakes were made. Meanwhile, the validation curve should converge to a certain acceptable accuracy. The behavior of the developed model can be diagnosed with the help of the curves' shape and the learning dynamics.

In the three graphs (Fig. 26, 27 and 28), the curve for training is rising (increasing accuracy) with respect to the number of epochs tending to 1 or 100%, whereas for validation curves it converges to 79, 85 and 90 % for 1, 2 and 3 bar steps respectively, taking into account that, as mentioned before, results may vary slightly due to the nature of the algorithm. The curves stabilize at different number of epochs, in which 1 bar step needs approximately 170 epochs to learn the information from data; this is obvious since we seek narrow classes forecasting (intervals). In the case of 2 and 3 bar steps, the model stabilizes earlier since the information is easier to get for larger intervals (approximately 55 and 30 epochs respectively).

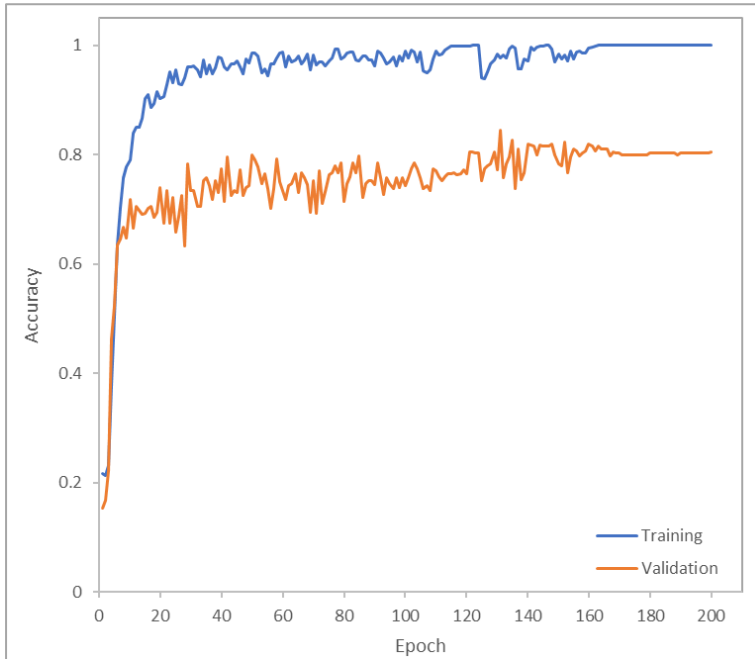


Fig. 26 - Learning curve for 1 bar pressure step classification

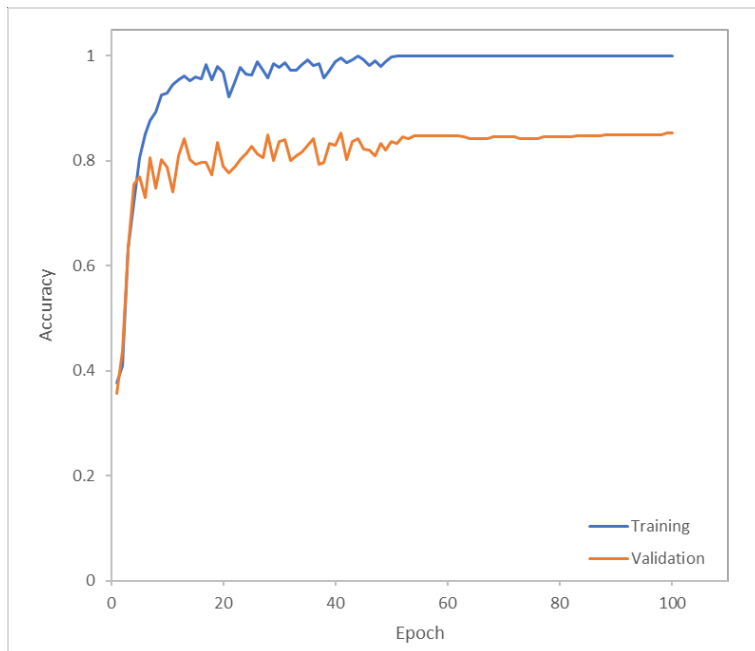


Fig. 27 - Learning curve for 2 bar pressure step classification

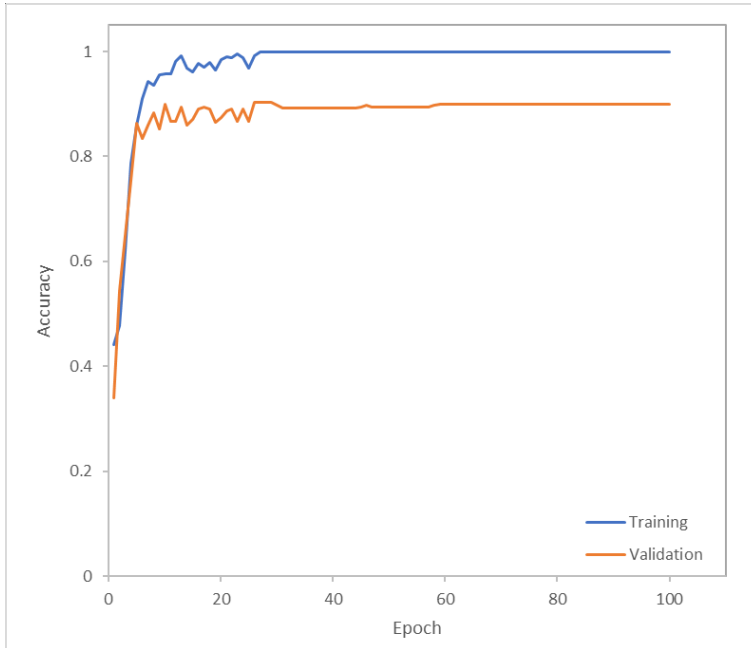


Fig. 28 - Learning curve for 3 bar pressure step classification

### ***4.3 Results of the clustering model***

An autoencoder was built and trained to extract the main features and learn to rebuild the initial data from these features. The used data is of 1017 network sections for 2000 different configurations, the model was trained and tested on 80 and 20% of the configurations respectively. The model consists of an artificial neural network of 7 layers (input, 5 hidden and an output layer). The input layer is of 1017 neurons that is equal to the number of pressurized network sections, the number of neurons decreases until the called bottle neck with 20 neurons for extracting the main features of the data. The number of neurons by layer increases back symmetrically to reach 1017 at the output layer, the autoencoder is illustrated in Fig. 29.

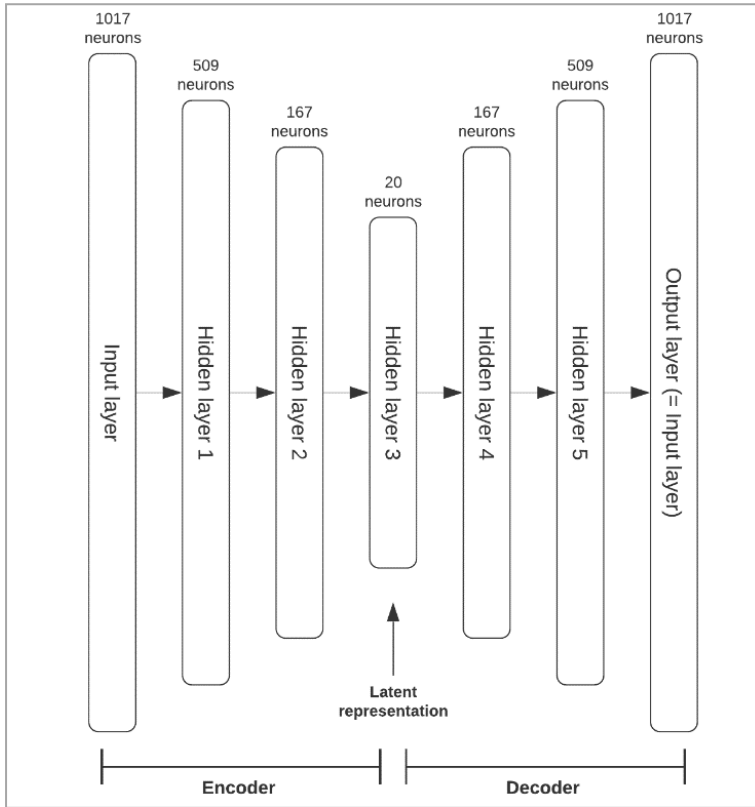


Fig. 29 - The autoencoder neural network structure

The efficient Adam gradient descent optimization algorithm and the Mean Squared Error (MSE) loss function were used. The rectified linear unit (ReLU) activation function was selected for layers.

The first part, the encoder, is saved and called after to be combined with t-SNE as shown in Fig. 30. The model ends by clustering the different sections of the network to groups of similar behavior zones to the perturbation.

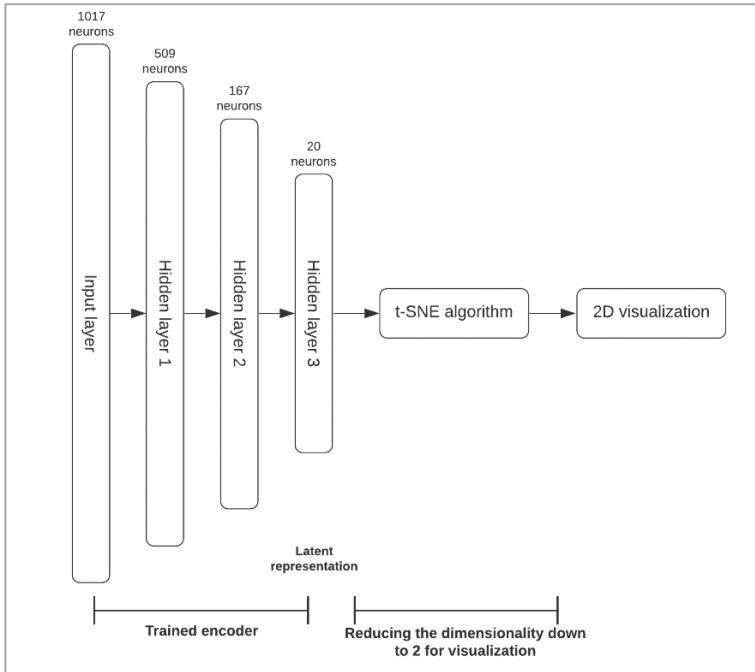


Fig. 30 - A combined encoder and t-SNE algorithm model.

### 4.3.1 Learning process

To assess the committed error of the autoencoder in rebuilding the input starting from the extracted features along the learning process, two curves (training and validation curve) were plotted to visualize the decreasing error (Fig. 31). As mentioned before, the used loss function is the mean squared error, the committed error is decreasing along with the number of epochs that ends with converging curves to 0.003 after 45 epochs.

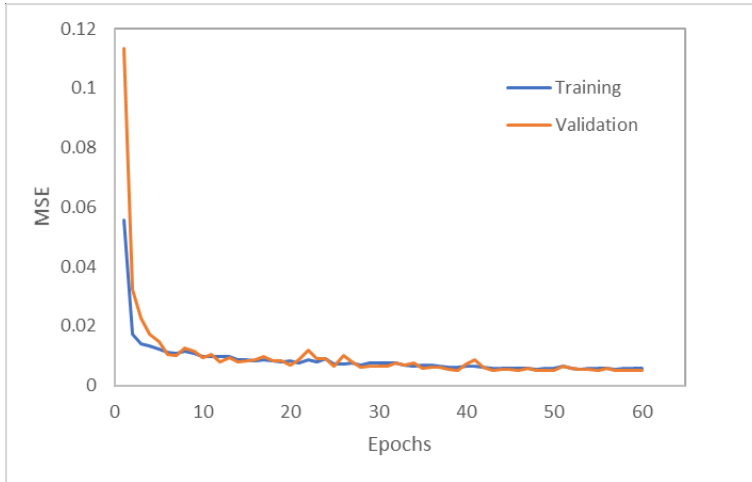


Fig. 31 - Loss function curves for the training and test sets during model training

### 4.3.2 Clustering output

As explained here before, clustering is often used as a data analysis technique for discovering interesting patterns in data, such as groups of customers based on their behavior.

Fig. 32 shows the results obtained on the patterns obtained from the different sections' response to the perturbation occurred through the network. Each point on the graph represents a replicate pattern, where the multidimensional data has been reduced to two dimensions. The output of the encoder that represents the main features extracted from the data is presented in 2 dimensions by the use of t-SNE. The graph shows nine visible clusters, however, they can be considered more or less according to how much similar clusters are. The number of clusters is the number of zones that have similar behavior to the perturbation and that are representing all the network sections. Thus, the deep neural network for pressure classes forecasting will be trained only on this number of clusters instead of being trained on all the sections (that are 1017 in the present case study). Larger the network, more noticeable the reduction in time consumption as the number of sections will be much higher.



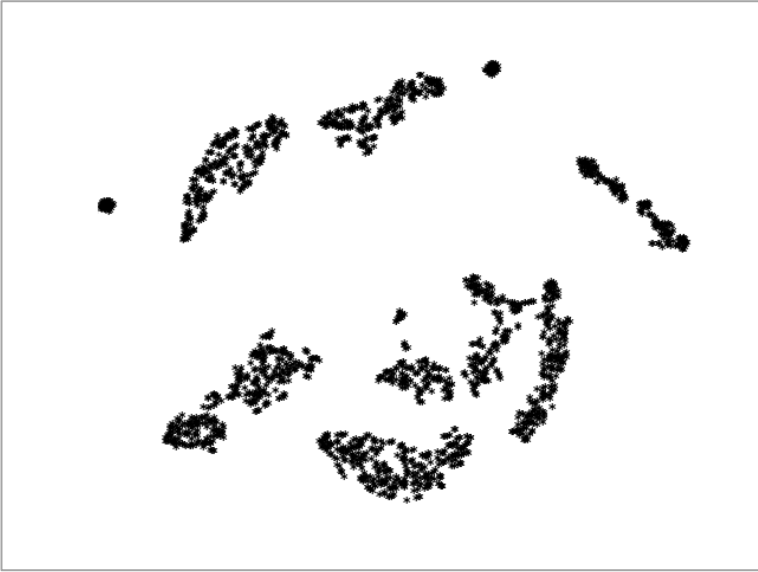


Fig. 32 - Two dimensions t-SNE representation of the encoder output

## ***CHAPTER 5. CONCLUSION***

The performance of on-demand pressurized irrigation systems is highly affected by the unsteady flow, which highlights the importance of identifying appropriate indicators to be integrated in designing and managing pressurized system.

To that end, a user-friendly computer code capable of simulating the real operating conditions of a pressurized irrigation system and, consequently, the unsteady flow, through the random opening and closing of hydrants was developed. The code makes it possible to investigate and quantify the generated effects through two simple indicators developed in the framework of this study (Hydrant Risk Indicator 'HRI' and Relative Pressure Exceedance 'RPE').

Such informative indicators could significantly contribute to more efficient operation management of on-demand pressurized systems by avoiding highly risky probabilistic configurations. Moreover, they could be embedded in the designing phase allowing for better interpretation of the impacts of different design alternatives.

Management of pressurized irrigation systems in which the assessment of the possible perturbation needs powerful and new techniques, principally for real time decision making. In fact, even if many software codes were available with different computational potential and applicability the analysis of the aforementioned phenomenon is computationally expensive and requires powerful computers and extended simulation times.

As a result, a novel framework is presented to drastically minimize time consumption and merit inclusion as part of a real-time decision-making system for the optimum irrigation network management. The model requires initially enough time to be trained, but it rapidly becomes cost-effective.

In the second part of this work, using supervised learning algorithms, a model was successfully built and trained to provide accurate forecasting pressure classes.

Given a set of features (hydrant different functioning modes, closing time etc.) and a target (the actual pressure class), the model finds a function approximator for classification that is a sequence of non-linear layers linking the input with the output layer.

According to the required pressure step, classes are forecasted with accuracy of around 75, 84 and 90 % 1, 2 and 3 bar steps respectively. Indeed, the results are even better as the mistaken forecasting in most cases are one class higher or lower, because in many cases the absolute pressure was on the classes edges.

The developed model using deep neural networks in the present work represents the first attempt to use deep learning for assessing the perturbation in pressurized irrigation systems working on demand. The introduced technique and the developed model could be a first step in mapping the perturbation in such systems and provide a powerful tool for managers and decision makers towards a more consistent control in particular for real time decision making. Furthermore, the model provides more flexibility and a great potential for being trained and learn more features from the farmers behavior feedback by implementing necessary equipment.

Many parameters affect the behavior of the pressurized irrigation network to the often-occurring perturbation. For instance, the location of each section having its own specification such as the location respect to the source of perturbation, the pipe diameter, the type of material, the thickness etc. For this, different behavior to the perturbation may occur at different sections of the network. As each section of the irrigation network represents a different behavior to the perturbation, the deep neural network based model developed in the previous step should be trained on each of them that will be time consuming. Therefore, a second model was developed using autoencoding combined with t-SNE algorithm for clustering the irrigation network sections into groups of similar behavior and reducing the number of trainings for the previous model.

The first part of the clustering model is called autoencoder that is an artificial neural network, it is trained to rebuild its input after being reduced to what is considered

main features. Being trained for 60 epochs using the mean squared error as loss function, the error made by the model is reduced and stabilized at 0.003. Subsequently, the first part of the autoencoder (called encoder) is registered and used for extracting the main features of the data. The t-SNE algorithm clusters the extracted features into groups that represent similar behavior zones of the irrigation network to the perturbation. Nine zones are distinguished, on which the deep learning model will be trained instead of all the sections. It is important to mention that the training will be only once, thus the trained model will be saved and combined for a decision support system.



## Acknowledgements

I would like to express my deepest gratitude to my supervisors **Prof. Umberto Fratino** and **Dr. Eng. Nicola Lamaddalena** for their continuous support and help with advices, suggestions, encouragement and professional guidance during all the work phases enabled the completion of the present work. I could not have imagined having better mentors for my Ph.D study.

I wish to extend my special thanks to **Prof. Juan Antonio Rodríguez Díaz** for giving me the opportunity to join his team and for his precious advices. I would like to take this opportunity to thank as well all the team, specially, **Dr. Rafael González Perea**.

Besides them, I would like to thank my thesis reviewing committee, **Prof. Emilio Camacho Poyato** and **Prof. André Daccache** for their insightful comments and encouragement.

I am grateful to **Polytechnic University of Bari** for providing the opportunity to resume my academic attainments through the PhD scholarships program for the international students.

I owe a deep sense of gratitude to **CIHEAM Bari** for an open and productive environment for the research. Special thanks to the Land and Water Resources Management department staff, **Dr. Fadhila**, **Dr. Roula** and all my colleagues and friends. I dedicate warm thanks to my brothers and not friends **Wahid** and **Ahmed**.

Last but not least, I want to thank my mother **Zeineb** and my father **Idris**, without you none of this would indeed be possible. My supporters, my brother **Zakaria**, my dear sisters, my wife and my daughter who were always next to me.



## References

- Abuiziah, I., Ahmed, O. & Driss, O., 2013. Simulating Flow Transients in Conveying Pipeline Systems by Rigid Column and Full Elastic Methods: Pump Combined with Air Chamber. *International Journal of Mechanical, Industrial Science and Engineering*, pp. 2391-2397.
- Afshar, M. & Rohani, M., 2008. Water hammer simulation by implicit method of characteristic. *International Journal of Pressure vessels and piping*, 85(12), pp. 851-859.
- Bergant, A. et al., 2008. Parameters affecting water-hammer wave attenuation, shape and timing—Part 1: Mathematical tools. *Journal of Hydraulic Research*, pp. 373-381.
- Bergant, A. et al., 2008. Parameters affecting water-hammer wave attenuation, shape and timing—Part 2: Case studies. *Journal of Hydraulic Research*, 46(3), pp. 382-391.
- Boulos, P. F., Karney, B. W., Wood, D. J. & Lingireddy, S., 2005. Hydraulic transient guidelines for protecting water distribution systems. *Journal-American Water Works Association*, pp. 111-124.
- Brownlee, J., 2016. *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*. s.l.:Machine Learning Mastery.
- Calejo, M., Lamaddalena, N., Teixeira, J. & Pereira, L. S., 2008. Performance analysis of pressurized irrigation systems operating on-demand using flow-driven simulation models. *agricultural water management*, pp. 154-162.
- Chaudhry, M. H., 1979. *Applied hydraulic transients*. Springer.
- Chaudhry, M. H., 2014. Transient-flow equations. In: *Applied hydraulic transients*. s.l.:Springer, pp. 35-64.
- Clemmens, A. J., 2006. Improving irrigated agriculture performance through an understanding of the water delivery process. *Irrigation and Drainage: The journal of the International Commission on Irrigation and Drainage*, pp. 223-234.



Deisenroth, M. P., Faisal, A. A. & Ong, C. S., 2020. *Mathematics for machine learning*. s.l.:Cambridge University Press.

Derardja, B., Lamaddalena, N. & Fratino, U., 2019. Perturbation indicators for on-demand pressurized irrigation systems. *Water*, p. 558.

Er-Rami, M. et al., 2021. Analysis of irrigation system performance based on an integrated approach with Sentinel-2 satellite images. *Journal of Agricultural Engineering*.

Fernández García, I. et al., 2020. Trends and challenges in irrigation scheduling in the semi-arid area of Spain. *Water*, p. 785.

Fouial, A., Lamaddalena, N. & Rodriguez Diaz, J. A., 2020. Generating hydrants' configurations for efficient analysis and management of Pressurized Irrigation distribution systems. *Water*, p. 204.

Géron, A., 2019. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. s.l.:O'Reilly Media.

Ghidaoui, M. S., 2004. On the fundamental equations of water hammer. *Urban Water Journal*, 1(2), pp. 71-83.

Ghidaoui, M. S., Zhao, M., McInnis, D. A. & Axworthy, D. H., 2005. A review of water hammer theory and practice. *Appl. Mech. Rev.*, 58(1), pp. 49-76.

Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep learning*. s.l.:MIT press.

Jin, W., 2020. *Research on Machine Learning and Its Algorithms and Development*. s.l., IOP Publishing, p. 012003.

Kingma, D. P. & Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lamaddalena, N. et al., 2004. Participatory water management in Italy: case study of the Consortium "Bonifica della Capitanata". *Options Méditerranéennes Series B*, pp. 159-169.

Lamaddalena, N., Khadra, R., Derardja, B. & Fratino, U., 2018. A new indicator for unsteady flow analysis in pressurized irrigation systems. *Water Resources Management*, pp. 3219-3232.

Lamaddalena, N. & Pereira, L. S., 2007. Pressure-driven modeling for performance analysis of irrigation systems operating on demand. *Agricultural water management*, pp. 36-44.

Lamaddalena, N. & Sagardoy, J., 2000. *Performance Analysis of On-Demand Pressurized Irrigation Systems*. s.l.:Food & Agriculture Org..

Larock, B. E., Jeppson, R. W. & Watters, G. Z., 1999. *Hydraulics of pipeline systems*. s.l.:CRC press.

Mambretti, S., 2013. *Water hammer simulations*. s.l.:Wit Press.

Markoulidakis, I. et al., 2021. Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*, 9(4), p. 81.

Mitchell, T. M., 1997. *Machine learning*.

Perea, R. G., Poyato, E. C., Montesinos, P. & Díaz, J. R., 2019. Prediction of irrigation event occurrence at farm level using optimal decision trees. *Computers and electronics in agriculture*, pp. 173-180.

Playán, E. & Mateos, L., 2006. Modernization and optimization of irrigation systems to increase water productivity. *Agricultural water management*, pp. 100-116.

Raschka, S. & Mirjalili, V., 2017. *Python Machine Learning: Machine Learning and Deep Learning with Python. Scikit-Learn, and TensorFlow. Second edition ed.*

Reddy, Y., Viswanath, P. & Reddy, B. E., 2018. Semi-supervised learning: A brief review. *Int. J. Eng. Technol*, p. 81.

Renault, D., Facon, T. & Wahaj, R., 2007. *Modernizing Irrigation Management: The MASSCOTE Approach--Mapping System and Services for Canal Operation Techniques*. s.l.:Food & Agriculture Org..

Rodríguez Díaz, J. A., Perea, R. G. & Moreno, M. Á., 2020. Modelling and Management of Irrigation System. *Water*, 12(3), p. 697.

Sharp, B. & Sharp, D., 1995. *Water hammer: practical solutions*. s.l.:Elsevier.

Srinath, K., 2017. Python—the fastest growing programming language. *International Research Journal of Engineering and Technology (IRJET)*, 4(12), pp. 354-357.

Subramanian, V., n.d. *Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch*. 2018: Packt Publishing Ltd.

Triki, A., 2018. Further investigation on water-hammer control inline strategy in water-supply systems. *Journal of Water Supply: Research and Technology—AQUA*, 67(1), pp. 30-43.

Wan, W., Zhang, B. & Chen, X., 2019. Investigation on water hammer control of centrifugal pumps in water supply pipeline systems. *Energies*, 12(1), p. 108.

Wichowski, R., 2006. Hydraulic transients analysis in pipe networks by the method of characteristics (MOC). *Archives of Hydro-Engineering and Environmental Mechanics*, 53(3), pp. 267-291.

Wylie, E. B., Streeter, V. L. & Suo, L., 1993. *Fluid transients in systems*. s.l.:Prentice Hall Englewood Cliffs, NJ.

Yan, W., 2021. *Computational methods for deep learning*. s.l.:Springer.

Záruba, J., 1993. *Water hammer in pipe-line systems*. s.l.:Elsevier.

# Bilal Derardja

---

Date of birth: 08/01/1989 | **Nationality:** Algerian | **Gender:** Male |

Tel: (+39) 3512586731 | [bilal.derardja@poliba.it](mailto:bilal.derardja@poliba.it) | [b.derardja.dz@gmail.com](mailto:b.derardja.dz@gmail.com) |



## ● WORK EXPERIENCE

---

01/02/2022 – CURRENT – Italy

**WATER RESOURCES MANAGEMENT CONSULTANT – CIHEAM BARI - MEDITERRANEAN AGRONOMIC INSTITUTE OF BARI**

---

Models' developer and researcher in water resources management related issues.

02/05/2018 – 31/10/2018 – Italy

**ASSISTANT RESEARCHER – POLYTECHNIC UNIVERSITY OF BARI**

---

Indexes for unsteady flow in hydraulic pressurized systems

01/01/2017 – 31/12/2017 – Italy

**ASSISTANT RESEARCHER – MEDITERRANEAN AGRONOMIC INSTITUTE OF BARI**

---

- Pursue the research activity on the topic “Definition of perturbation indicators as a part of the MASSPRES program”, the activity is part of “FAO-Africa” project.
- Facing an over designed irrigation system problem in Egypt, I have developed a software for running different possible scenarios providing the manager a decision tool for best irrigation scheduling.
- Worked in several regional water related projects in the Mediterranean countries (e.g. Egypt, Jordan and Tunisia) related to on demand irrigation systems, water conjunctive use and irrigation modernization assessment.

## ● PROJECTS

---

2018 – 2020 (participation)

**A new methodology to modernize pressurized irrigation systems: MAPPING SYSTEM AND SERVICES FOR PRESSURIZED IRRIGATION SYSTEMS - MASSPRES**

---

MASSPRESS is a step-wise approach for the mapping of the behaviour of pressurized irrigation systems and the assessment of their hydraulic performance at network and hydrant level. The collaboration was between the Food and Agriculture Organization of the United Nations (“FAO”) and International Centre for Advanced Mediterranean Agronomic Studies – Mediterranean Agronomic Institute, Bari (CIHEAM Bari).

I participated in developing the unsteady state flow conditions model, as a part of the COPAM model for assessing the performance of the irrigation systems. I participated as well in writing a chapter in the FAO paper as an output of the elaborated work.

02/2020 – 02/2020 (participation)

**Organization of a high level technical workshop on “Can Water Productivity Improvements Save Us from Global Water Scarcity?” at the CIHEAM Bari Institute**

---

Within the framework of the WASAG Working Group on Agricultural Water Use (led by the International Water Management Institute, IWMI) IWMI organized a workshop together with the CIHEAM - Mediterranean Agronomic Institute of Bari (CIHEAM Bari) and the CGIAR Research Program on Water, Land and Ecosystems (WLE), hosted at the CIHEAM Bari.

2018 – 2018 (participation)

**Support Sustainable Water Management and Irrigation Modernization for Newly Reclaimed Areas Project (TCP/EGY/3604) and Implementing the 2030 Agenda for water efficiency/productivity and water sustainability in NENA countries.**

I participated in developing a model for assessing the hydraulic performance of an oversized irrigation system in Egypt for the rehabilitation.

---

2017 – 2017 (participation)

## Strengthening Agricultural Water Efficiency and Productivity on the African and Global Level

---

The project was funded by the Swiss Agency for Development and Cooperation (SDC). The project was of collaboration between the Food and Agriculture Organization of the United Nations (“FAO”) and the Mediterranean Agronomic Institute of Bari MAIB aiming mainly at:

- Reducing vulnerability in Jordan in the context of water scarcity and increasing food/energy demand;
- Enhancing capacity for increased water use efficiency in small-scale irrigation in Burkina Faso, Morocco and Uganda.

## EDUCATION AND TRAINING

---

11/2018 – 01/2022 – Italy

**PHD IN RISK AND ENVIRONMENTAL, TERRITORIAL AND BUILDING DEVELOPMENT** – Polytechnic University of Bari, Bari (Italy)

---

04/10/2014 – 13/10/2016 – Italy

**MASTER OF SCIENCE IN LAND AND WATER RESOURCES MANAGEMENT** – Mediterranean Agronomic Institute of Bari

---

09/2009 – 06/2014 – Algeria

**HYDRAULIC ENGINEERING** – The National Highest School of Hydraulics (Ecole Nationale Supérieure d'Hydraulique), Blida

---

09/2013 – 09/2013

**TRAINING AT ALGERIE ENGINEERING (BUREAU OF STUDIES), BATNA (ALGERIA)**

---

07/2012 – 07/2012

**TRAINING AT HYDRAULIC SUBDIVISION OF N'GAOUS, BATNA (ALGERIA)**

---

## LANGUAGE SKILLS

---

Mother tongue(s): **ARABIC** | **THAMAZIGHTH**

Other language(s):

	UNDERSTANDING		SPEAKING		WRITING
	Listening	Reading	Spoken production	Spoken interaction	
<b>FRENCH</b>	C1	C1	B2	B2	B2

---

Levels: A1 and A2: Basic user; B1 and B2: Independent user; C1 and C2: Proficient user

## ● PUBLICATIONS

---

### Monograph

---

- Derardja, B 2014, 'L'approvisionnement en eau potable de la ville d'Arris à partir du transfert Koudiat Lemdouar-Batna', Engineering thesis, National High School of Hydraulics, Blida (Algeria).
- Derardja, B et al. 2015, 'Irrigation Project, San Ferdinando Scheme', First level Master of Science, Mediterranean Agronomic Institute of Bari (Italy).
- Derardja, B 2016, 'Perturbation indicators for the pressurized irrigation systems', Second level Master of Science, Mediterranean Agronomic Institute of Bari (Italy).
- Derardja, B et al., 2022. 'Innovative Approaches for Mapping the Pressurized Irrigation Systems Performances Under Unsteady Flow Conditions', PhD dissertation, Polytechnic University of Bari.

### Publications

---

- Lamaddalena, N., Khadra, R., Derardja, B., & Fratino, U. (2018). A new indicator for unsteady flow analysis in pressurized irrigation systems. *Water Resources Management*, 32(9), 3219-3232.
- Derardja, B., Lamaddalena, N., & Fratino, U. (2019). Perturbation indicators for on-demand pressurized irrigation systems. *Water*, 11(3), 558.
- Alobid, M., Derardja, B., & Szűcs, I. (2021). Food Gap Optimization for Sustainability Concerns, the Case of Egypt. *Sustainability*, 13(5), 2999.
- Alobid, M., Derardja, B., and Szűcs, I. (2022). Economic Analysis of an Optimized Irrigation System: Case of Sant' Arcangelo, Southern Italy. *European Online Journal of Natural and Social Sciences*, 11(1), 134-155.

### Submitted

---

- A research article titled "A rough artificial intelligence model for forecasting unsteady state pressure values in irrigation systems", written by Bilal Derardja, Umberto Fratino, Nicola Lamaddalena, R. González Perea and J.A. Rodríguez Díaz. The article was submitted to "Computers and Electronics in Agriculture" on December 05<sup>th</sup>, 2021.
- A research article titled "The effects of Land Consolidation Projects of Pressurized Irrigation System Performance and the Cost: A Case Study from Turkey", written by Firat Arslan, Aymen Sawassi, Bilal Derardja, Hasan Degirmenci and Nicola Lamaddalena. The Article was submitted to "Advances in Water Resources" on January 31<sup>th</sup>, 2022.

## ● JOB RELATED SKILLS

---

Using different programming languages, four models have been developed:

---

- Water-hammer simulator, in the framework of the master of science thesis for better understanding the unsteady flow phenomenon in the pressurized irrigation systems. Three new indicators have been set up.
- Steady-State simulator for analysing a specific conditions over designed irrigation system in Egypt.
- Using Octave, a mathematical model was build to find the optimum land reallocation and production distribution for minimizing the food gap (case study of Egypt).
- An intelligent model that simulates the perturbation in pressurized systems using machine learning.