

Article

The Digital Twin Realization of an Ejector for Multiphase Flows

Giovanni Mazzuto ^{*}, Filippo Emanuele Ciarapica, Marco Ortenzi  and Maurizio Bevilacqua 

Dipartimento Ingegneria Industriale e Scienze Matematiche, Faculty of Engineering, Università Politecnica delle Marche, 60131 Ancona, Italy; f.e.ciarapica@staff.univpm.it (F.E.C.); marcoorte0@gmail.com (M.O.); m.bevilacqua@staff.univpm.it (M.B.)

* Correspondence: g.mazzuto@staff.univpm.it

Abstract: Despite the extensive use of ejectors in the process industry, it is complex to predict suction and motive fluids mixture characteristics, especially with multiphase flows, even if, in most cases, mixture pressure control is necessary to satisfy process requirements or to avoid performance problems. The realization of an ejector model can allow the operators to overcome these difficulties to have real-time control of the system performance. In this context, this work proposes a framework for developing a Digital Twin of an ejector installed in an experimental plant able to predict the future state of an item and the impact of negative scenarios and faults diagnosis. ANNs have been identified as the most used tool for simulating the multiphase flow ejector. Nevertheless, the complexity in defining their structure and the computational effort to train and use them are not suitable for realizing standalone applications onboard the ejector. The proposed paper shows how Swarm Intelligence algorithms require a low computational complexity and overperform prediction error and computational effort. Specifically, the Grey Wolf optimizer proves to be the best one among those analyzed.

Keywords: ejector; multiphase flow; Swarm Intelligence algorithms



Citation: Mazzuto, G.; Ciarapica, F.E.; Ortenzi, M.; Bevilacqua, M. The Digital Twin Realization of an Ejector for Multiphase Flows. *Energies* **2021**, *14*, 5533. <https://doi.org/10.3390/en14175533>

Academic Editor: Ali Turan

Received: 15 July 2021

Accepted: 1 September 2021

Published: 4 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Ejectors are standard devices in the process industry for gas extraction or vapor from the specific system or container. An ejector can be considered as a compressor or vacuum pump, with no moving parts, relatively inexpensive and easy to maintain. It uses a high-pressure carrier fluid to drive a low-pressure suction fluid subject to an intermediate back pressure.

Despite the extensive use of ejectors, it is complex to predict mixture characteristics, mainly when operating with multiphase flows. Nevertheless, mixture pressure control is necessary to satisfy process requirements or avoid performance problems in most cases.

In this context, realizing a Digital Twin (DT) of the ejector can allow the operators to overcome these difficulties to have real-time control of the system performance. The notion of DT assumes a greater importance in a digital real-time image of a physical entity, system, or the like, which grants companies the performance improvement. The connection between the digital and physical systems is made easier thanks to the increased development of embedded sensors technology [1], signal processing algorithms [2], and wireless communications systems [3,4]. Since the DT provide a Realtime tracking of the physical world for monitoring, adjusting, and optimizing actual processes by various modeling techniques, such as simulation-, mathematics-, or data-based modeling.

In the literature, it is possible to find various techniques used to solve ejector modeling and theoretic and investigational analysis about motion in the multiphase fluids system. These techniques study the dynamic behavior of the ejector and the fluid-dynamic behavior of the liquids/gas passing through it to predict and optimize its operation considering several working conditions [5]. Predicting ejector performance was one of the first approaches to ejector modeling that was proposed [6]. The authors proposed a 1D analysis,

i.e., a mathematical model using a set of algebraic equations, differential equations, and conservation equations of thermodynamics, which characterizes the behavior of the ejector. This method has also been used by other authors such as [7]. In addition, one of the leading modeling/analysis techniques of ejector fluid dynamics behavior, still used today, is the Computational Fluid Dynamics analysis (CFD). For the purpose described above, various authors have used this technique [8–11]. CFD analysis can solve complicated equations, which characterize the motion of fluids, whose analytical resolution is instead feasible only when considering laminar (not turbulent) flows. This technique generally performs calculations/operations with a high computational cost. Reference [12] introduced the Shock Circle Model and showed that the latter is more straight forward than the one-dimensional modeling approaches and can foresee the ejectors performance in binding modes with more precision.

The mentioned methodologies are characterized by the fact that they use vector calculations and algebraic/differential equations that can be complex. Moreover, sometimes they imply, for their resolution, the knowledge of certain physical quantities specific to fluids, such as enthalpy or entropy, which in many cases are not measurable. Another critical aspect to consider is that the model carried out does often have to interface easily with other components of the plant (pump and tank) to create a DT of the overall plant. Such complexities would make the realization of the virtual model of the ejector very complex or sometimes impossible.

By analyzing the literature, there are no papers that propose a DT for managing an ejector. The proposed approach tries to overcome this gap by developing a DT that, throughout the evaluation of steady states, the analysis of past problems, and the forecast of potential tendencies, can offer complete support in the decision-making process to comprehend the ejector behavior. The proposed DT framework is tested in an experimental plant for multiphase ejector modeling.

It is essential to highlight that most simulation applications could solve only one specific design and offline optimization [13]. Thus, the application of DT features needed a significant change in the simulation approach [14]. In light of this, the application of proper simulation techniques brings DTs to life, thus becoming testable [15]. Hence, a Digital Twin Simulation Interface (DTSI) is essential for meeting all the requirements associated to the product or the simulation of the item life cycle. In particular, through customized interfaces, the interaction between the DT and the physical system can easily simulate different scenarios to identify proper actions to be applied to the second one.

In this paper, the Digital Twin is equipped with a simulation tool based on supervised machine learning techniques to overcome these problems and guarantee maximum interface ability and versatility. Through simulation tools, DT can help operators to understand physical objects as complex as an ejector system and predict the suction and motive fluids mixture characteristics.

The simulation model is one of the essential tools of the DT, and it must allow operators to examine the relationships established among the various variables of the problem. It must have low computational complexity and a low need for computational resources to minimize the system latency. For this reason, not only the Artificial Neural Networks (ANNs) but also the Swarm Intelligence (SI) algorithms have been decided to be used to evaluate as simulation tools. The implementation of SI algorithms is not complex, and the cost associated with these types of projects is not high, conversely to the ANNs. Moreover, the ANNs are essentially a “Black-Box”, which means that the networks do not allow to examine the relationships established among the various variables of the problem in the individual stages of elaboration but only provide “the obtained solution”. Finally, unlike the SI algorithms, ANNs present a high computational complexity. They need more computational resources. The proposed paper aims to demonstrate how emerging AI techniques, such as Swarm Intelligent algorithms, can allow developers to realize a valid tool for advanced system monitoring. In particular, they can be used onboard of

the considered item with no strict requirements in terms of computational power and device features.

The proposed study is organized as follows: Section 2.1 shows the multiphase system and ejector modeling literature review. A description of used techniques is briefly reported in terms of advantages, disadvantages, and application fields. Then, Section 2.2 describes the Digital Twin framework used for the multiphase ejector installed in an experimental plant. In Section 3, the research approach for developing the simulation tool and resulting models are described. Finally, in Section 4, the discussion and conclusions are presented.

2. Materials and Methods

2.1. Literature Review

The multiphase flow investigation and the ejector performance are one of the most relevant topics in many research fields: from oil and gas industries to refrigeration applications and gathering information about fluid and well properties such as the amount of gas and oil in the mixture, their pressure, and temperature [16].

According to [17], simulation is used to predict the pressure loss and phase velocities for stationary multiphase flow in wells and pipelines. The majority of the analyzed papers used CFD to predict experimental results since simulations save time and money accurately. For example, a three-dimensional CFD model was compared by [18] with the experimental data in different working conditions. Reference [19] provided a numerical research of an ejector for steam based on CFD to distinguish correct experimental conditions allowing one to perform a cycle of reliable operations on an ejector. In particular, the structure of the flow and the combination process within the ejector were evaluated considering a solar-driven air conditioning system. New models have been developed for predicting ejector performance at a critical point and breaking point, based on constant pressure mixing and constant pressure disturbance. Reference [20] developed ejector models to predict the performance of the ejector at a critical point and at a breaking point considering constant pressure mixing and disturbance hypotheses obtaining an ejector model for the entire operating range through a method of analysis of the effect of the change (EOC) to identify the efficiencies of ejector components and influences on ejector shape and accuracy. Reference [21], through CFD and orthogonal test, carried out an optimization analysis of structure parameters of stem ejector for the waste heat recovery and, also, to see how those single-factors affect the ejector using a single-factor analysis. Reference [22] studied the deviations from Darcy's law by focusing on high-speed non-Darcy flow and low-speed pre-Darcy flow, using an ejector to move air from the pores by sucking the porous medium. Reference [23] numerically evaluated how the inlet and outlet nozzles diameters and the divergent section variations affect the steam ejector performance and functioning (under different flow pressure conditions). Reference [24] presented a theoretic assessment of the impact of Venturi devices in gas wells by forming simulations in a steady state to adapt to experimental and theoretic pressure profiles. They used Aspen Plus[®], for the reproduction of the creation of gases in wells, and then, through CFD simulations, they reproduced the multiphase mixture flow in a venturi machine referring to ANSYS CFX[®].

By focusing on the ejector and multiphase flow modeling, it is possible to highlight if ANNs have always been used for this purpose. Indeed, [25] provided an ANN model to foresee the pressure loss considering a Venturi scrubbers. The authors designed three independent ANNs, comparing the results with those calculated using other models. This analysis showed that the results of the ANNs were more similar to the experimental data. References [26,27] realized an ANN to calculate the liquid holdup in a horizontal two-phase flow. Reference [28] used Artificial Neural Networks to perform a thermodynamic analysis of the ejection-absorption cycle of thermal systems. In particular, the authors have calculated the energy losses in the system, avoiding the classical thermodynamic analysis that uses complex differential equations and complex simulation programs. Reference [29] developed an ANN to estimate the output pressure from an ejector, given various input states. The authors then proposed an alternative method of ejector modeling compared

to traditional models (thermodynamic and CFD models). This research also lays the foundation for the control strategies for constructing an ejector-based refrigeration system using machine learning methods. Reference [30] used an ANN for the identification of the flow pattern through the natural logarithmic normalization.

Reference [31] have modeled a hybrid air conditioning system to an ejector using artificial neural networks to predict the performance of the latter. The authors used MLP, RBF, and SVM neural networks, comparing the results and identifying the best performing in the MLP network regarding the forecasting error. In comparison with previous studies, the aim of the proposed approach is the prediction and the performance control of an ejector not only through a simulation model but by trying to realize a real DT of the physical ejector. Several simulation models are tested for the DT. In addition to the well-known ANNs, the SI algorithms are tested as they provide a clear view of relationships established among the various variables of the problem in every individual stage of elaboration. The results of the literature analysis on SI algorithms are very different, and they underline the absence of SI algorithms for the ejector modeling. However, they are used in many research fields, from the biomedical one [32] to decision-making process [33], from data mining [34] to vehicle routing [35,36]. Nevertheless, the low computational complexity of SI algorithms and their low need for computational resources make them appropriate for minimizing the system latency.

2.2. The DT Model

Figure 1 shows the reference model of the proposed DT. It allows the virtual creation of a physical process, offering a static and dynamic analysis tool. Moreover, it is possible to understand how to spread information among all the available digital objects to increase the actors' safety level.

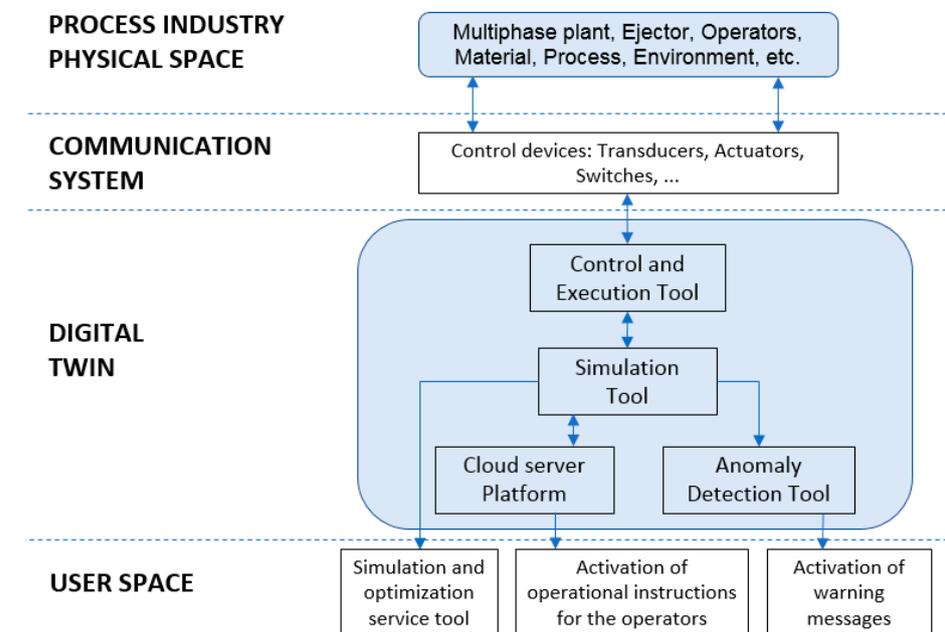


Figure 1. The reference model of the DT.

The main features of the proposed DT system refer to the controlling plant, the use of simulation tools for better plant management through scenarios analysis, anomalies detection, and predictive maintenance. For these reasons, the required platform needs to be fast and computationally not complex, so that it is possible to connect more users and manage many actions.

The reference model consists of four main levels as shown in Figure 1. In the last layer (User Space), there are the model outputs.

2.2.1. The Physical Space of an Industrial Process

The plant test used is an experimental plant located inside the Department of Industrial Engineering and Mathematical Science (DIISM) of the Università Politecnica delle Marche (Ancona, Italy).

The plant simulates a classic situation in extraction processes, such as exploiting the pressure of a reservoir whose pressure is higher than the transport pressure to create a suction on a reservoir whose pressure is not high enough for transport on the line. In the actual case, the treated fluids are crude oil and natural gas, while water and air are used in the experimental plant. Indeed, in the oil, chemical, and nuclear industries, there is frequently the problem of transporting two-phase gas–liquid mixtures in a single pipeline. Particularly in the oil industry, it is very often the case that oil pressure in a field is not sufficient to bring it to the surface. In such circumstances, the most obvious solution is installing appropriate pumps positioned at the surface and the bottom of the oil well. An undoubtedly more efficient solution is to exploit a nearby well with higher pressure than the transport pressure. To mix two phases at different pressures and to impress the necessary transport energy, the use of gas–liquid ejectors is conceivable. The lack of moving parts, the consequent extreme simplicity of the apparatus, the low maintenance and installation costs, and the absence of sealing problems make the gas–liquid ejector a machine with many possibilities of applications in the transport of both hydrocarbon mixtures and corrosive, toxic, or radioactive gases. Figures 2 and 3 show the 3D picture and the functional scheme of the mentioned plant.

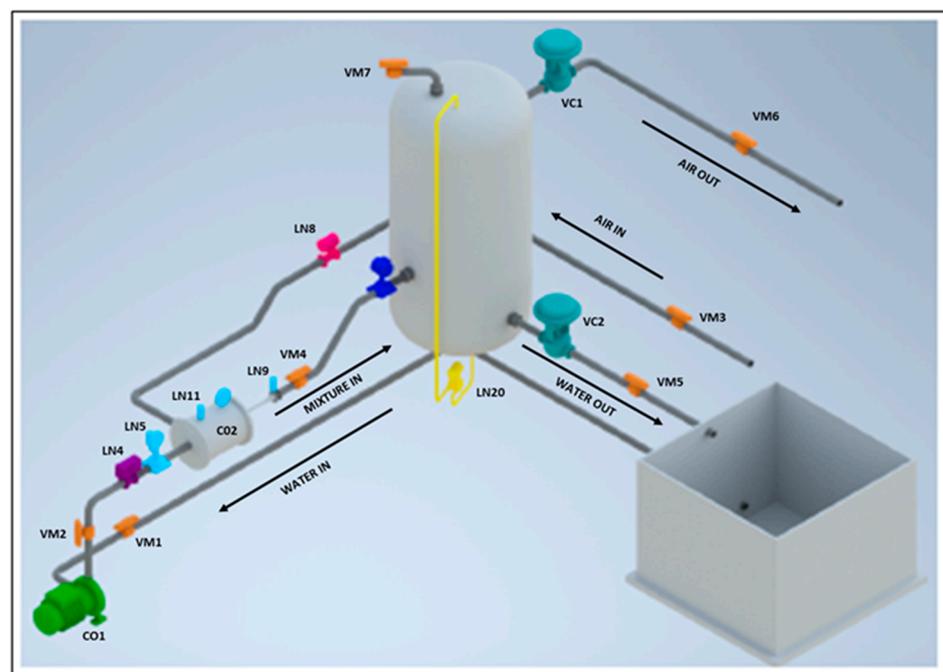


Figure 2. 3D scheme of the DIISM plant.

In particular, the pump (CO1) connected to the open tank (CO4) takes a flow of water and sends it to the ejector (CO2) at a specific pressure. In the ejector, the transformation from pressure energy into kinetic energy of the liquid takes place. The resulting depression draws airflow from the outside at atmospheric pressure. In the final part of the ejector, the two fluids are mixed and the pressure of the mixture is recovered from the current value in the ejector chamber. The diverging cone at the ejector outlet, made of transparent material (perspex), guarantees further pressure recovery.

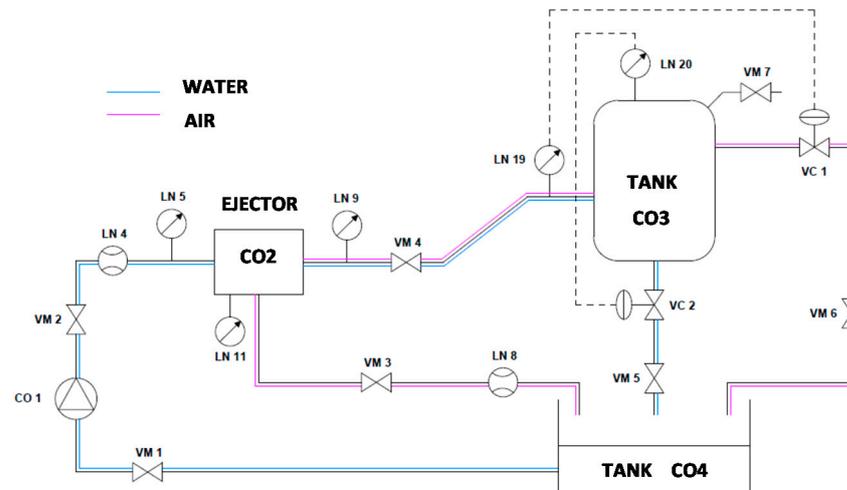


Figure 3. Plant functional scheme.

The two-phase mixture, formed at the ejector outlet, flows into a pressure vessel (CO3), which acts as a vertical separator of the liquid and gaseous components. Through two solenoid valves (VC1 and VC2), it is possible to control the water flow and the airflow at the outlet, respectively; this system regulates the pressure and the liquid level inside the tank.

2.2.2. The Ejector Functioning in Brief

The ejector carries out the primary process of this experimental plant. An ejector is a machine without moving parts, used both as a compressor and as a pump to lower the pressure of a fluid through the supply of fluid (of similar or different nature). Because of their versatility, ejectors can be used for several applications, where requirements such as “Constructive simplicity”, “Compactness”, “Reliability”, and “Safety” prevail.

Many different applications exploit the basic principle. A fluid with high momentum, meeting one with low momentum, transmits the momentum from the first to the second as an inelastic shock—the principle on which the operation of the Venturi tube is based.

2.2.3. The Communication System

This level is reserved to the data transfer between DT and the plant. The physical equipment is supervised and perceived by specific devices for gathering data and device control connected to cameras, sensors, actuators, and composite devices. This system connects the visual system parts to the digital ones for the synchronization and vice versa.

The plant analyzed in this paper is equipped with a series of sensors (pressure, flow, and level) that monitor the process. Table 1 describes the sensors and components equipped on the plant. The 780 L vertical tank is examined and supplied with a 10 bar safety valve (VS1). In the current configuration, a pump capable of providing a maximum pressure of 5.5 bar is used. The PVC pipes can be connected with bonded fittings, and they can resist pressures up to 6 bar. However, as a safety measure, a 4.5 bar pressure has been fixed as the maximum pressure limit.

Table 1. Sensors and components equipped on the plant.

Component ID	Description
CO1	Monobloc centrifugal pump Vergani 32–201
CO2	Ejector
CO3	Vertical separator Elbi 780-I
CO4	Open tank
LN4	Flow sensor Foxboro Magnetic Flowtransmitter
LN5	Absolute pressure sensor Setra 280E
LN11	Absolute pressure sensor Setra 280E
LN9	Absolute pressure sensor Setra 280E
LN19	Gauge pressure sensor Foxboro 841GM-CI1
LN20	Differential pressure sensor Foxboro IDP-10
LN8	Foxboro Vortez volume flow sensor DN 50
VC1	Pneumatic solenoid valve ECKARDT MB6713
VC2	Pneumatic solenoid valve ECKARDT MB6713
VM1	2-way shut-off valve FIP DN 50 PN 16
VM2	2-way shut-off valve FIP DN 50 PN 16
VM3	2-way shut-off valve FIP DN 50 PN 16
VM4	2-way shut-off valve FIP DN 50 PN 16
VM5	2-way shut-off valve FIP DN 50 PN 16
VM6	2-way shut-off valve FIP DN 50 PN 16
VM7	2-way shut-off valve FIP DN 50 PN 16

2.2.4. The DT Layer

The third layer consists of:

The Control-Execution Tool—this allows for the connection between the physical and virtual spaces via sensors, transducers, etc., and it allows for the management and control of the process plant. An Arduino Mega 2560 hardware platform, which integrates an ATmega2560 microcontroller, manages the sensors readings. The Arduino platform controls the solenoid valves VC1 and VC2 to maintain predefined pressure and liquid level values in the tank.

Simulation and Anomaly Detection Tools—the simulation tool allows the company to make a digital ejector model. It can work online or offline: in the first case, the inputs come from the plant sensors, and, in the second case, they are introduced by the user. In the offline mode, the item virtual representation allows managers to investigate what-if scenarios without a physical realization, thus avoiding potential risk situations for operators. For example, the tool can be used to run a new installation in a virtual way to identify risks for the operators before effectively activating the installation or identifying the risk connected to a simulated maintenance operation. In the online mode, it obtains measures from sensors and changes the virtual object parameters value if the asset changes its condition. Thus, it allows users to compare the data obtained by the simulation system. Sensors detect conditions for activating warning advice if the discrepancy is greater than the thresholds. This is the purpose of the anomaly detector.

A detailed analysis of the simulation tool developed in this work has been included in the Results section.

The Platform of the Cloud Server—the platform captures readings by the plant sensors. Consequently, a standard architecture of the server is not enough because of the huge amount acquired by the environment. The same applies to traditional relational databases, which cannot tolerate an excessive number of simultaneous access requests for reading and writing. For this reason, the platform is designed specifically for the sensors readings' acquisition, organizing, and visualization using cloud solutions.

2.2.5. The User Space

The last layer is the User, and it refers to a specific device, human, or system. The proposed framework offers different types of services:

- Using virtual sensor data, energy costs or performance factors, optimization tools can be stimulated to run a large amount of “what-if” simulations to assess the readiness or the required adjustments for the considered set. They allow users to optimize or control system operations during the operation to diminish risk, decrease energy utilization, and increase the efficiency of the system.
- A cloud service platform connects the DT model to operators by displaying sensor readings and analyzing results using cloud products. It provides extensive data analysis, extraction, and value-added services for businesses, such as the activation of operational instructions for the system management in terms of maintenance and safety.
- Activation of warning messages. If the ML application foresees risk conditions, wearable systems can warn operators of anomalies. In addition, corrective measures can be carried out.

3. Results

The core of the DT framework proposed in this work is the simulation tool of the ejector. For this reason, this section focuses on explaining what steps have been taken to develop a supervised simulation model based on machine learning algorithms. The Glossary Section summarizes and describes the terminologies and abbreviations adopted in the text. All the realized algorithms have been implemented in a Matlab 2020 environment for research purposes and in python to be uploaded on a Raspberry device positioned onboard the ejector.

As mentioned above, it was chosen to opt for a supervised method because we needed a low-computational-cost tool. The model must easily interface with other Digital Twins of the plant (pump and tank).

The most-used supervised simulation models are Artificial Neural Networks (ANNs). In this work, the Swarm Intelligence (SI) algorithms were also tried because operators need to examine the relationships established among the various variables of the problem and not only the model result through a “black box”.

The development of both types of algorithms (ANNs and SI) for the simulation models followed the following steps:

- Preprocessing—this consists of the preparation of the dataset for analysis. The pre-processing phase incorporates all the steps for the “preparation of the dataset”. The feature engineering step plays a fundamental role in how the data and the analysis previously made are used to create new entries within the dataset that allows the use of the machine learning system with as much information as possible.
- Split Data—it is an analytical step to understand how best to train the machine learning system. There are two parts to machine learning systems: the first is “training”, which trains the system. Then, the system is ready to perform what has been learned and test whether the training performed in the previous phase was successful; this is conducted through the “score” or “test”.
- Choice of Models and Comparison Method—the training needs input from the theoretical model to be used and trained. It is not necessary to choose in advance what the best model is. It is possible to train more models and choose the best performing one after the results are obtained.
- Practical Comparison between the Models—the various algorithms are evaluated to choose the one that performs best in the model designed and developed. In general, the performance of the model in its generality is evaluated. Thus, the comparison of the involved algorithms is performed in terms of computational time in the proposed paper. All models are evaluated concerning the prediction accuracy, referring to the Variance Inflation Factor (VIF) and other indicators [37].

3.1. Dataset

The data available collected experimentally represent the operating conditions under which the plant has worked over the years. In particular, taking into consideration the plant characteristics, the pressure and flow rate of the motive fluid (see Figures 4 and 5), P_{liq} and Q_{liq} , vary from a minimum of 1.9 bar to a maximum of 10.38 bar and from a minimum of 1.44 m³/h to a maximum of 282.39 m³/h. The constraint pressure imposed on the tank, P_{serb} , varies from a minimum of 1 bar (atmospheric pressure) to a maximum of 3.8 bar. All possible ejector configurations are defined concerning the size of the fluid outlet nozzle diameter, D_1 , and the diffuser, D_2 (from a minimum of 5.2 mm to a maximum of 62 mm). These parameters have a considerable influence on the determination of the flow rate of air sucked by the ejector, Q_{gas} , at atmospheric pressure ($P_{gas} = P_{atm}$). The pressure inside the diffuser, P_{diff} , varies from a minimum of 0.96 bar to a maximum of 1.96 bar. Table 2 shows an extract of the data collected for the ejector configuration currently mounted on the system ($D_1 = 11$ mm and $D_2 = 29$ mm).

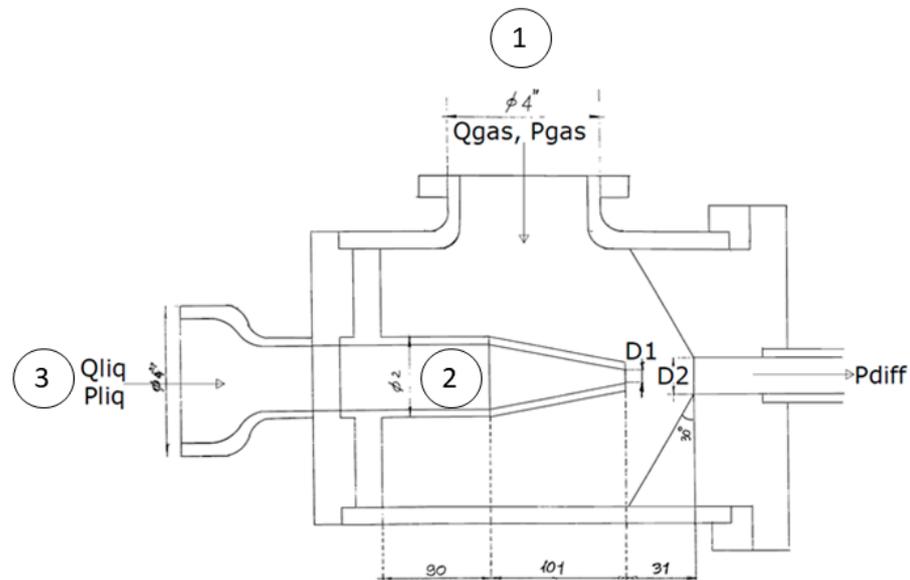


Figure 4. The ejector scheme. (1) Identifies the suction mouth of the fluid sucked in; (2) the converging nozzle, where the pressure is reduced, and the speed of the motor fluid is increased by the Venturi effect; and (3) the inlet port of the motive fluid.

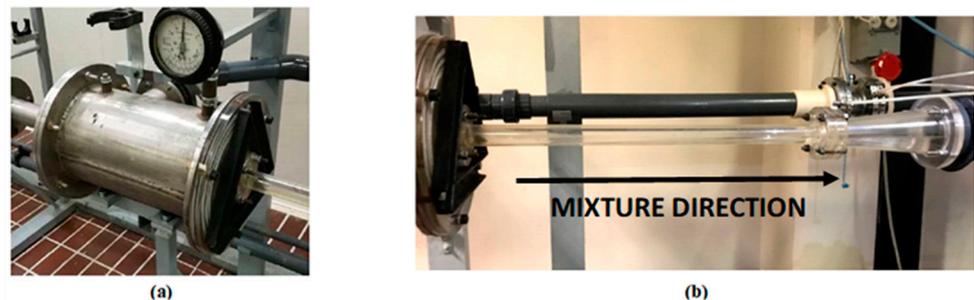


Figure 5. The ejector components analyzed in this work. (a) Shows the mixing chamber, which allows the mixing and exchange of energy and momentum between motive fluid and sucked fluid; (b) the diverging diffuser, which slows down the fluid mixture and contributes to a partial pressure recovery of the fluid current.

Table 2. Extract of operating data for current ejector configuration (D1 = 11 mm and D2 = 29 mm).

PresLiq (bar)	P _{diff} (bar)	PresSerb (bar)	Q _{liq} (m ³ /h)	Q _{gas} (m ³ /h)
4	1.26	1.3	8.45	11.957
4.5	1.26	1.3	9.29	17.16
5	1.26	1.3	10.09	22.05
5.5	1.26	1.3	10.83	26.63
6	1.26	1.3	11.53	30.89
6.5	1.26	1.3	12.18	34.83
3.5	1.36	1.4	7.56	3.16
4	1.36	1.4	8.45	8.69
4.5	1.36	1.4	9.29	13.90
5	1.36	1.4	10.09	18.79
5.5	1.36	1.4	10.83	23.37
6	1.36	1.4	11.53	27.62
6.5	1.36	1.4	12.18	31.57
4	1.46	1.5	8.45	5.43

To identify the best model to be used for the digital twin realization, the available dataset has been normalized and divided into two classes: one for the model training and one for its testing. The same dataset has been used for both the ANN and SI evaluation to compare them more significantly.

3.2. The ANNs Model

An ANN is a set of artificial neurons that model those in a human brain. Each connection transmits a signal to the others like areal synapses. The “signal” is a real number, and the output is computed by functions that reprocess those inputs. A weight characterizes all neurons and connections and changes as learning progresses, increasing or decreasing the strength of the signal in the connection under consideration. In addition, neurons may have a threshold that inferiorly limits the transmitted signal. The learning rate describes the number of corrective steps that must be taken to fix errors in the observation. Specifically, a high learning rate shortens the training time with lower final accuracy, while a lower learning rate takes longer but is characterized by better accuracy. To avoid network oscillations, the improvements use an adaptive learning rate to increase or decrease as appropriate. The momentum concept allows the balance between the gradient and the last change to be weighted so that the weight adjustment depends to some extent on the last change. In this work, the momentum and batch size parameters have been set to equal 0.9 and 150, respectively. The momentum has been fixed according to [38] since it is considered to be optimal. For evaluating the other parameters, a systematic trial has been carried out to evaluate the best configuration. Specifically, the number of network neurons increases from 1 to 40; the hidden layers range between 1 and 2 (since the dataset is not so complex); the epochs numbers range from 1 to 1000; and, finally, the learning rate is evaluated to be 0.01, 0.05, and 0.1. The activation function between the various layers is logistic, while the optimization algorithm used is a gradient descent.

Once the systematic experimentation has been performed, the best configurations are selected for each adapted learning rate (0.01–0.05–0.1), at 1–10 neurons, 10–20 neurons, and 20–40 neurons. At this stage, the best configuration is defined by evaluating the prediction accuracy on the test dataset.

Comparing the observed variable with what was obtained from the neural model with 1 hidden layer and 10 neurons (Figure 6) shows that the interpolation error is contained. The model realized correctly follows the air intake flow rate measured by the system. In particular, the mean quadratic error (MSE) obtained from the training dataset is equal to 8.25×10^{-11} (comparable to zero, given the machine error committed using the Matlab software).

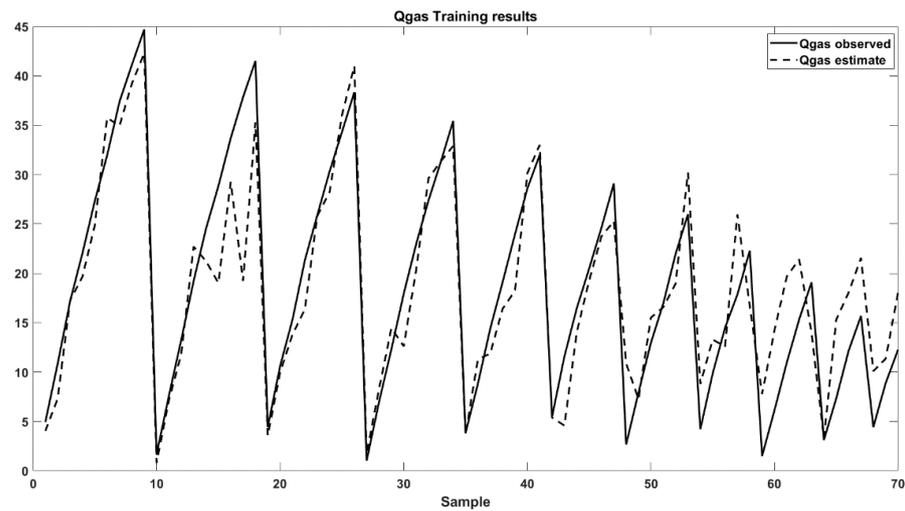


Figure 6. Comparison of observed and estimated intake air flow rate with NN at 1 hidden level and 10 neurons.

Using then, a percentage of about 15% of the original dataset both to perform a first testing and model validation, we obtain for the first case an MSE equal to 3.29×10^{-9} and in the second case equal to 1.09×10^{-9} . The correlation between the observed and estimated signal is about 0.9123 (p -value equal to 3.2546×10^{-28} at a significance level set at 5%). Given the R-value of the Pearson index and considering that the p -value is much lower than the level of significance, there is a strong correlation.

The multilayer perceptron (MLP) neural network with two hidden levels, the number of the input and output neurons, activation function, and activation algorithm that are concerned, retains the same characteristics as the one-level hidden network.

The same considerations of the hidden-layer model can be made for the two-hidden-layers neural network. Comparing the observed variable with the one obtained by the neural model (Figure 7) shows how the interpolation error is contained and how the realized model correctly follows the air intake flow rate measured by the system. In particular, the mean quadratic error (MSE) obtained from the training dataset is equal to 1.49×10^{-10} ; therefore, given the machine error committed using the Matlab software, it is comparable to zero.

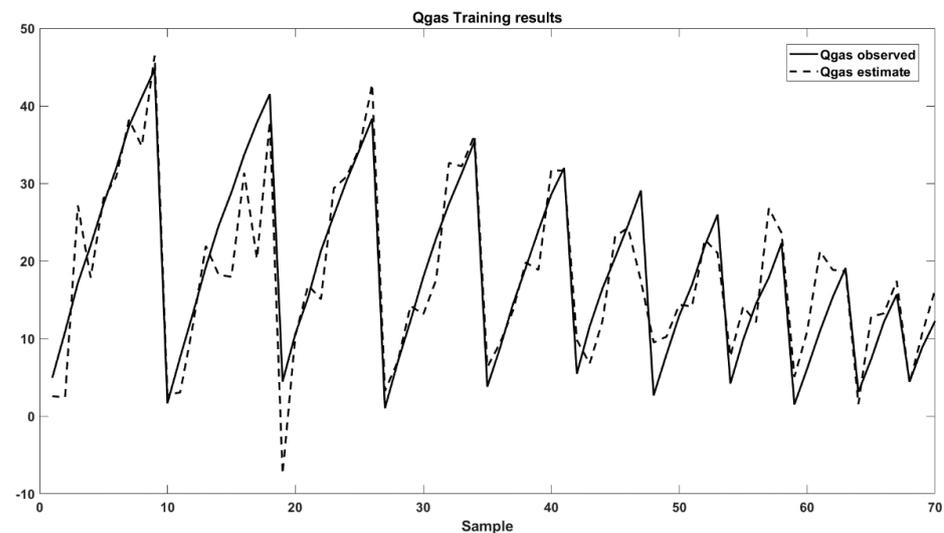


Figure 7. Comparison of observed and estimated intake air flow rate with NN at 2 hidden levels and 10 neurons.

Comparing the results and the performance obtained with one-hidden-layer neural network model (ten neurons) and two hidden layers shows that the improvements do not differ. This implies that, with the same dataset, using a more complex network does not improve the performance of the simpler network.

3.3. The SI Model

Among the bioinspired algorithms, a particular class of algorithms has been developed, taking inspiration from the intelligence of the swarm. The algorithms in this class are swarm intelligence, also called Si-based. In particular, the SI techniques aim to determine the optimal solution of a given problem, exploiting the global behavior of a “swarm” of homogeneous agents. While each agent can be considered “non-intelligent”, the whole system of several agents shows a self-organizing behavior guaranteeing a sort of collective intelligence. Each agent shares information and experiences, managing, in this way, to solve even very complex tasks. The Swarm Intelligence group is vast and includes procedures that take into consideration the collective behavior of insects, such as ants, bees, and fireflies, and other animals such as flocks of birds, shoals of fish, and wolves. The main algorithms belonging to this class are Particle Swarm Optimization (PSO), Artificial Bee Colony algorithm (ABC), and Ant Colony Optimization (ACO). These are agent-based optimization techniques specific to problems in which the objective function can be decomposed into independent partial functions. Each agent maintains a hypothesis that is tested iteratively by evaluating a randomly chosen partial objective function.

In this study, eight SI algorithms were tested to identify the ejector model, and several functions were implemented concerning the parameters. Figure 8 briefly describes the general framework used by SI algorithms.

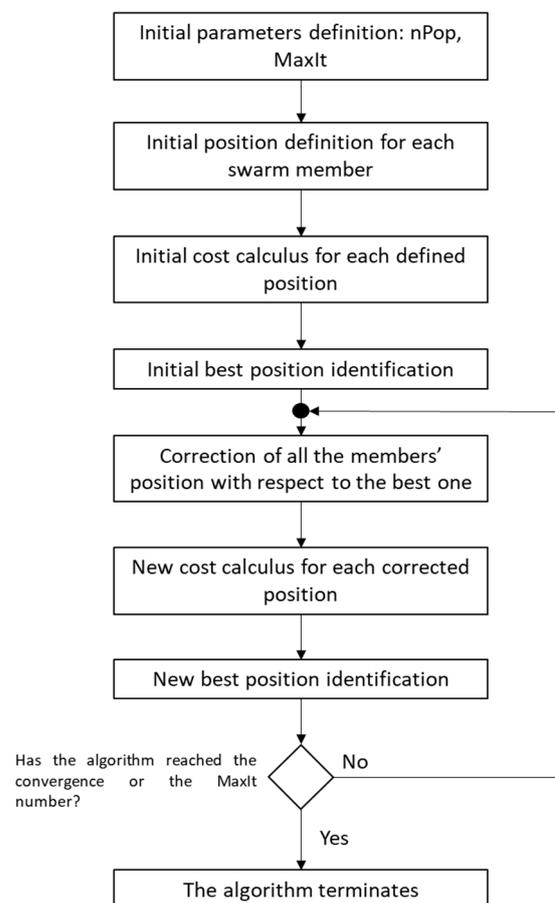


Figure 8. The General Swarm Intelligence algorithm scheme.

The user defines the swarm population size (nPop) and the maximum iterations number (MaxIt) at the initial step. At this point, for each swarm member, an initial position and its relative positional cost are defined according to a specific function decided by the user. In this paper, the Root Mean Squared Error was chosen since it describes the data concentration around the best fit line.

The position with the minimum cost value is assumed to be the temporary best solution. The iterative phase starts until the maximum number of iterations is reached, or the error between the solutions of two consecutive iterations is equal to a fixed threshold. According to the specific SI algorithm adopted, each position is corrected with respect to the best one during this phase. Specifically, it is possible to refer to Table 3 to analyze how each algorithm defines the new position for each swarm member.

Table 3. References for a detailed study.

Algorithm	Reference	Algorithm	Reference
Chicken Swarm	[39]	Particle Swarm Algorithm	[40]
Bat Algorithm	[41]	Artificial Bee Colony	[42]
Cuckoo Search	[43]	Firefly Algorithm	[44]
Grey Wolf Optimizer	[45]	Artificial Fish-Swarm Algorithm	[46]
Ant Colony Optimization	[47]	Krillheard	[48]

Table 4 shows an extract of the analyzed scenarios (experiment number, intercept present or not, function type, population size, and maximum iteration number). Table A1, in Appendix A, shows all the functions, f , considered in the scenarios analysis.

Table 4. Extract of the analyzed scenarios (n —experiment number, C —intercept present or not, f —function type, nP —population size, and MI —maximum iteration number).

n	C	f	nP	MI	n	C	f	nP	MI	n	C	f	nP	MI
1	0	1	30	30	49	0	1	30	60	97	0	1	30	90
2	1	1	30	30	50	1	1	30	60	98	1	1	30	90
3	0	2	30	30	51	0	2	30	60	99	0	2	30	90
4	1	2	30	30	52	1	2	30	60	100	1	2	30	90
5	0	3	30	30	53	0	3	30	60	101	0	3	30	90
6	1	3	30	30	54	1	3	30	60	102	1	3	30	90
7	0	4	30	30	55	0	4	30	60	103	0	4	30	90
8	1	4	30	30	56	1	4	30	60	104	1	4	30	90
...								

The essential parameters for using the SI algorithms and the regression function are the number of maximum iterations to perform (MaxIt) and the agents number to use for the solution determination (nPop). Specifically, several simulations were carried out to test the various algorithms examined, each of which considers a particular function, a value of iterations between 30 and 90, and, finally, a swarm size between 30 and 90, for 144 simulative scenarios.

The different scenarios aim to identify the best parametric combination so that the solution is the best possible in terms of error for the observed signal but at the same time involves an acceptable computational effort and is evaluated in terms of calculation time.

In particular, the choice of the agents number in the swarm and the iterations number is fundamental to the computational balance since the total iterations number up to the end of the algorithm is equal to ($nPop \times MaxIt$). The demonstration of the above is reported, although only in some cases, in Table 5.

Table 5. Extract of scenario results for a set of algorithms used (nP—population size, MI—maximum iteration number, and R2a—R2 adjusted).

Alg	nP	MI	R2a	VIF	Time [s]	Alg	nP	MI	R2a	VIF	Time [s]
GW	30	30	0.87	7.91	2.66	BC	30	30	0.88	8.29	5.07
GW	60	30	0.87	8.14	5.39	BC	60	30	0.86	7.33	9.96
GW	90	30	0.02	1.05	7.84	BC	90	30	0.03	1.07	15.13
GW	30	60	0.17	1.24	5.39	BC	30	60	0.21	1.30	10.44
GW	60	60	0.15	1.21	10.61	BC	60	60	0.17	1.24	20.67
GW	90	60	0.02	1.05	17.36	BC	90	60	−1.66	0.39	33.04
GW	30	90	0.33	1.53	8.97	BC	30	90	0.34	1.56	16.98
GW	60	90	0.46	1.92	17.55	BC	60	90	0.44	1.84	33.19
GW	90	90	0.32	1.51	26.70	BC	90	90	−9.24	0.10	50.04
BaC	30	30	0.02	1.05	2.67	PS	30	30	0.86	7.56	2.70
BaC	60	30	0.14	1.19	5.26	PS	60	30	0.87	8.15	5.37
BaC	90	30	−0.03	1.00	7.84	PS	90	30	0.05	1.08	7.97
BaC	30	60	−0.06	0.97	5.44	PS	30	60	0.20	1.28	5.47
BaC	60	60	0.17	1.24	10.67	PS	60	60	−0.15	0.89	10.89
BaC	90	60	0.08	1.12	17.70	PS	90	60	0.11	1.15	18.15
BaC	30	90	−0.03	1.00	9.03	PS	30	90	0.38	1.66	9.08
BaC	60	90	−0.03	1.00	17.83	PS	60	90	0.53	2.19	18.13
BaC	90	90	−49.68	0.02	26.68	PS	90	90	−0.39	0.74	27.12

It is evident that as the combination of swarm size and the maximum number of iterations increase, the reliability of all the algorithms decreases enormously until the algorithm in question cannot be used. An example is the Bat Colony that is computationally expensive and without a positive effect on the system. The values are obtained considering the averages of the parameters for all sixteen functions considered.

By limiting the analysis to only good computational and estimation accuracy algorithms, Table 6 shows the models with high correlation and solution validity (Variance Inflation Factor (VIF) is more significant than 5), which all correspond to the linear model with zero intercept.

Table 6. Comparison of swarm intelligence algorithms analyzed in terms of VIF and computational time for linear bonding.

	VIF			Time [s]		
	Mean	STD Dev	Tot Rel	Mean	STD Dev	Tot Rel
Grey Wolf	8.03	0.12	1.47%	10.49	6.41	61.16%
Fish Colony	8.01	0.16	1.97%	46.66	33.26	71.30%
Water Cycle	7.51	0.63	8.41%	10.39	6.33	60.89%
Chicken Swarm	7.85	0.25	3.13%	10.50	6.40	60.98%
Particle Swarm	7.85	0.30	3.79%	10.70	6.56	61.32%
Bee Colony	7.81	0.49	6.26%	19.91	12.12	60.85%

Comparing VIF and Time composition for the considered algorithms, the best estimating algorithm for the realization of the digital ejector twin through SI algorithms is the Grey Wolf with a swarm size equal to 60 and the number of maximum iterations equal to 60 (the mathematical model reported in Equation (1)). Regardless, Relative Standard Deviation (RSD) analysis highlights how the SI algorithms are affected considerably by the set of parameters since the high values referred to each algorithm [49].

An example of the fitting is shown in Figure 9. It shows a mean absolute error equal to 0.4 m³/h and a standard deviation equal to 0.5 m³/h for an adjusted R2 value equal to 0.998 and a *p*-value close to 0, considering an alpha of 0.05.

$$Q_{gas}^* = -0.60 \cdot P_{liq} - 0.87 \cdot P_{serb} + 1.75 \cdot Q_{liq} \quad (1)$$

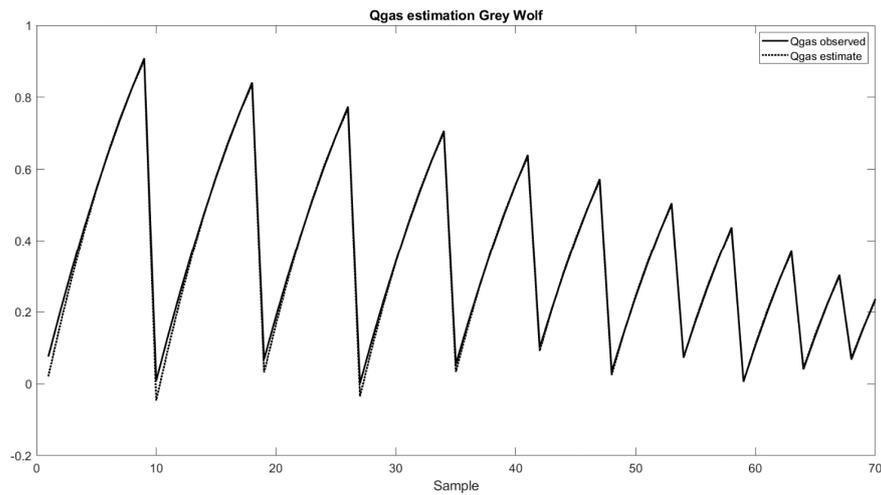


Figure 9. Example of observed and estimated variable tracking with Grey Wolf Algorithm.

3.4. Model Selection

Several tests have been performed to evaluate the optimal model to be implemented as Digital Twins of the ejector, and some are summarized in Table 7. Specifically, they concern possible operating situations of the plant in question considering the current configuration, i.e., with $D1 = 11\text{ mm}$ and $D2 = 29\text{ mm}$.

Table 7. Excerpt of tests performed for performance evaluation.

	P_{liq} (bar)	P_{diff} (bar)	P_{serb} (bar)	Q_{liq} (m ³ /h)	Q_{gas} (m ³ /h)
Test 1	6.30	1.46	1.50	11.92	26.77
Test 2	4.00	1.26	1.30	8.45	11.95
Test 3	5.00	1.66	1.70	10.09	9.01
Test 4	6.50	1.76	1.80	12.18	18.52

Figure 10 shows the comparisons between the airflow rate measured on the system and those estimated using the neural network at a hidden level and ten neurons and the Grey Wolf model. Both the neural network and the SI model correctly follow the real one, despite the oscillations it shows. In any case, the neural network constantly shows a significant deviation from the actual value.

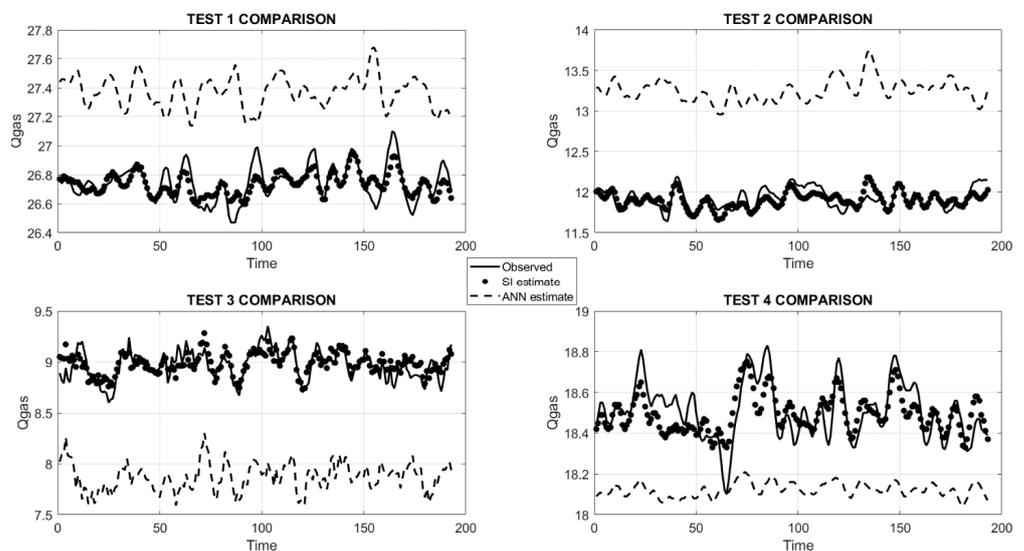


Figure 10. Comparison of estimates obtained with the neural network and the Grey Wolf models for tests.

Tests 1 and 2 show an overestimation of the actual value, while tests 3 and 4 show an underestimation. By contrast, the model realized through swarm intelligence algorithms presents a behavior constantly superimposed on the real one and much more stable than the neural model.

Table 8 and Figure 11 show how the variability of the estimation error obtained with the neural network presents a minor variability but a very high average estimation error. The average error ranged from a minimum of about 0.42 m³/h to a maximum of 1.33 m³/h on all the tests considered. The standard deviations of the error, when related to the relative mean value, show very low relative standard deviations. By contrast, the variability of the estimation error obtained with the Grey Wolf algorithm has a higher variability but a deficient average estimation error. The average error ranged from a minimum of about 0.38 m³/h to a maximum of 0.57 m³/h on all tests considered. The standard deviations of the errors, when related to the relative mean value (Relative Standard Deviation), show very high, albeit minor, relative standard deviations.

Table 8. Summary of the responses of the neural network models and Grey Wolf for the analyzed tests.

			MEAN	STD DEV	RSD
Test 1	NN	Observed	26.74	0.12	0.45%
		Estimate	27.37	0.11	0.40%
		Error	0.63	0.19	29.84%
	SI	Estimate	26.71	0.08	0.28%
		Error	0.06	0.05	84.41%
		Observed	11.92	0.12	1.04%
Test 2	NN	Estimate	13.25	0.14	1.04%
		Error	1.33	0.21	15.44%
		Estimate	11.90	0.10	0.83%
	SI	Error	0.07	0.06	77.39%
		Observed	8.97	0.15	1.62%
		Estimate	7.87	0.14	1.74%
Test 3	NN	Error	1.10	0.16	14.85%
		Estimate	8.98	0.11	1.17%
		Error	0.08	0.06	74.98%
	SI	Observed	18.51	0.14	0.74%
		Estimate	18.12	0.03	0.19%
		Error	0.39	0.13	33.40%
Test 4	NN	Estimate	18.49	0.09	0.49%
		Error	0.08	0.05	67.42%
	SI	Estimate	18.49	0.09	0.49%
		Error	0.08	0.05	67.42%

Following these experiments and considering the need to realize a light computational application, between ANN and the one through Swarm Intelligence, the second one was chosen to realize the Digital Twin of the ejector present in the plant. The demonstration is summarized in Table 9 and Figure 12.

The *t*-test returns an evaluation on the null hypothesis that the estimated data originated from a Gaussian distribution with a zero mean and unknown variance. The alternative hypothesis is that the distribution has no null mean. The H0 result, in Table 9, is 1, if the test rejects the null hypothesis at the 5% (or 10%) significance level and 0 otherwise. The value *p* is the probability of observing a more extreme test statistic than the value observed under the null hypothesis. Small values of *p* question the validity of the null hypothesis, and, in particular, for all tests obtained with ANN, the null hypothesis is rejected ($H_0 = 1$), and the *p*-value is close to zero. The Pearson test computes the correlation coefficients and *p*-values of a normally distributed variable. Table 9 shows in the Pearson section the correlation coefficient R and the relative *p*-values for testing the null hypothesis. Moreover, Rl and Ru identify the lower and upper bounds for a (1- α) confidence interval.

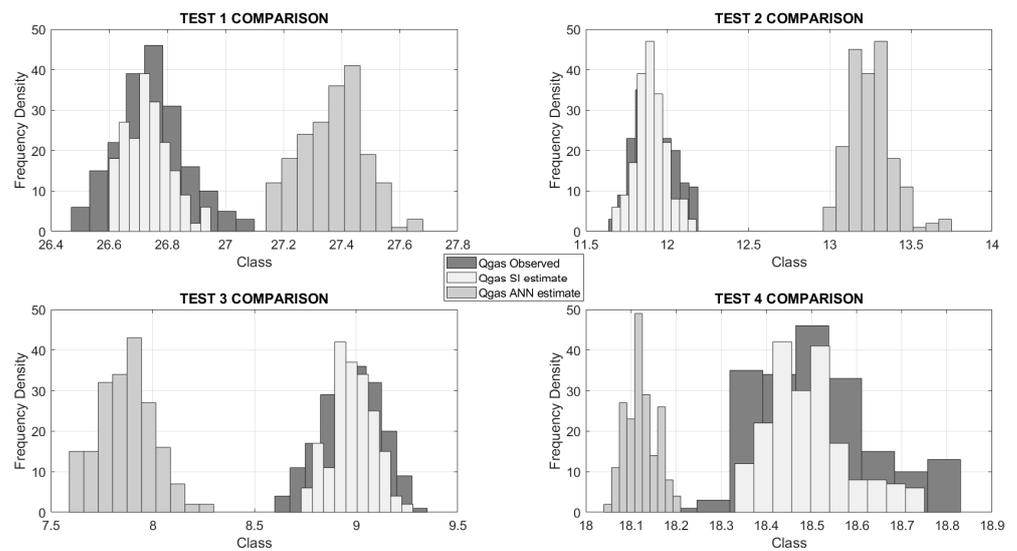


Figure 11. Comparison of the observed Q_{gas} values and the estimated ones.

Table 9. Statistical summary of fitting and comparisons procedure (p - v — p value, T- v —T value, RI—Person index lower bound, and Ru—Person index upper bound).

$\alpha = 0.05$	T Test							Pearson				
	H0	p - v	CI		T- v	SD	DF	R	p - v	RI	Ru	
T1	NN	1	4.6×10^{-8}	0.41	0.85	5.7	192	0.11	-0.31	9.6×10^{-6}	-0.43	-0.17
	SI	0	0.92	-0.16	0.14	-0.1	192	0.075	0.77	1.2×10^{-40}	0.71	0.82
T2	NN	1	3.7×10^{-18}	1.1	1.6	9.6	192	0.13	-0.23	0.001	-0.36	-0.096
	SI	0	0.84	-0.22	0.17	-0.21	192	0.098	0.67	3.5×10^{-27}	0.59	0.74
T3	NN	1	9.9×10^{-14}	-1.4	-0.83	-8.0	192	0.13	0.33	2.4×10^{-6}	0.2	0.45
	SI	0	0.91	-0.19	0.22	0.11	192	0.1	0.74	2.7×10^{-35}	0.67	0.8
T4	NN	1	2.1×10^{-23}	-0.45	-0.32	-0.11	192	0.034	0.33	1.8×10^{-6}	0.2	0.45
	SI	0	0.89	-0.19	0.17	-0.14	192	0.091	0.73	8.5×10^{-34}	0.66	0.79
$\alpha = 0.1$	T Test							Pearson				
	H0	p - v	CI		T- v	SD	DF	R	p - v	RI	Ru	
T1	NN	1	4.6×10^{-8}	0.44	0.81	5.6	192	0.11	-0.31	9.68×10^{-6}	-0.42	-0.20
	SI	0	0.91	-1.3	0.11	-0.1	192	0.075	0.78	1.29×10^{-4}	0.73	0.82
T2	NN	1	3.6×10^{-18}	1.1	1.5	9.6	192	0.13	-0.24	1.01×10^{-3}	-0.34	-0.12
	SI	0	0.83	-0.18	0.14	-0.2	192	0.098	0.68	3.51×10^{-27}	0.61	0.74
T3	NN	1	9.9×10^{-14}	-1.3	-0.87	-8	192	0.13	0.33	2.46×10^{-6}	0.22	0.43
	SI	0	0.91	-0.16	0.18	0.11	192	0.1	0.74	2.77×10^{-35}	0.69	0.79
T4	NN	1	2.1×10^{-23}	-0.44	-0.32	11	192	0.03	0.34	1.81×10^{-6}	0.23	0.44
	SI	0	0.88	-0.16	0.13	-0.14	192	0.09	0.73	8.52×10^{-34}	0.67	0.78

The smaller p -values related to SI interpolations than NN underline how the SI algorithms perform a better estimation than the artificial neural network. Indeed, by analyzing Figure 12, it is possible to see how the relationship between the actual inlet air flow rate and the esteemed one is quite linear.

The realized estimator was connected to the actual plant, and the working conditions, not used for the estimation phase, were tested to validate the obtained estimator. Figure 13 compares the actual airflow rate (blue line) and the esteemed one (red line) considering the inlet water flow rate (Q_{liq}) to be equal to about $10 \text{ m}^3/\text{h}$ at a pressure (PresLiq) equal to about 5.5 bar and the internal tank pressure (PresSerb) equal to 1.5 bar.

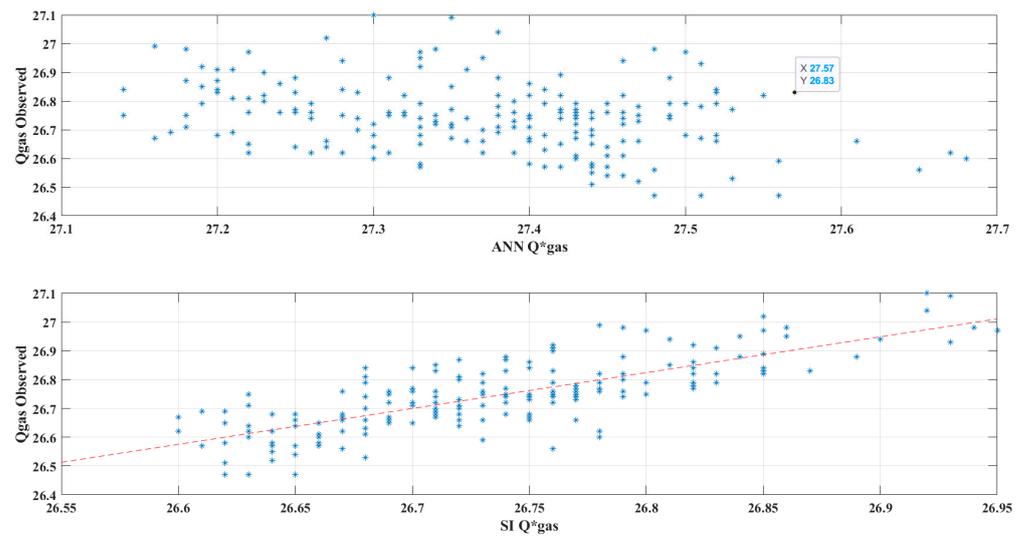


Figure 12. Comparison of the involved algorithms esteems (Q_{gas}^*) and observed (Q_{gas}) variable for test 1.

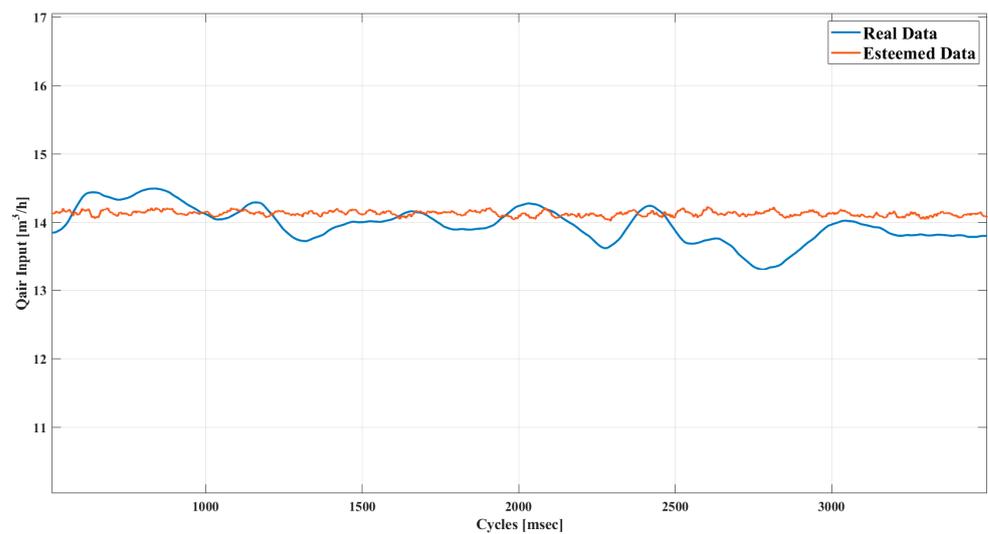


Figure 13. Comparison between actual inlet air flow rate and the esteemed one at regime condition.

The actual mean value for the inlet air flow rate (Q_{gas}) is about $13.9 \text{ m}^3/\text{h}$ with a standard deviation equal to $0.22 \text{ m}^3/\text{h}$; conversely, concerning the esteemed variable Q_{gas}^* , its mean value is equal to $14.12 \text{ m}^3/\text{h}$, and standard deviation is equal to $0.04 \text{ m}^3/\text{h}$.

Always considering the working conditions explained before, at cycle 142 for 500 cycles, the valve VM7 (see Figure 3) was opened to simulate a tank loss of pressure. The VM7 total opening implies a value of pressure inside the tank approximately equal to that of the atmosphere. In the absence of a pressure constraint, the actual inlet air flow rate is more significant than $13.9 \text{ m}^3/\text{h}$, the value identified in the regime condition. By analyzing Figure 14, this increase is evident both for the actual value (blue line) and the esteemed one (red line).

Figures 13 and 14 show a reasonable estimation of the inlet air flow rate with an estimation error close to 3% that can be considered suitable since the numerical approximation is due to the adopted algorithms and the noise introduced by the sensors. The research has highlighted the necessity of revamping some old sensors with a high noise level on the readings.

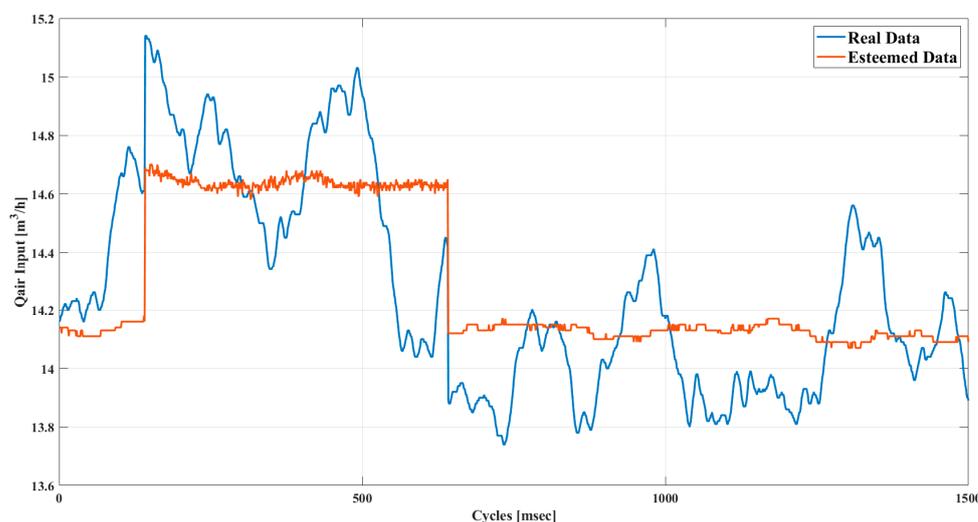


Figure 14. Comparison between actual inlet air flow rate and the esteemed one in an anomalous status.

4. Discussion and Conclusions

This paper proposes a reference model for implementing a Digital Twin of an ejector system operating with multiphase flows to analyze and predict the system performance.

The model encompasses all the critical phases of Digital Twin design and implementation. It starts with the analysis of the existing system. It ends by developing a helpful platform for researchers and technicians to provide a means to simulate and investigate scenarios that are otherwise too costly to explore. Moreover, the realization of a Digital Twin of the ejector aims at allowing the operators to have real-time control of the suction and motive fluids mixture performance.

The DT platform must be lean and easy to use. It must allow for the integrated management of all installation components and the prediction of the ejector performance. It involves implementing digital components that can be integrated without excessive computational effort so that the platform can be used by many users and on many different devices simultaneously. For this reason, a machine learning supervised model is necessary. Artificial Neural Networks and Swarm Intelligence algorithms were compared to identify the best one. Artificial Neural Networks were identified as the most used tool for the multiphase flow ejector, but, at the same time, no applications of Swarm Intelligence algorithms were identified for the same purpose. Swarm Intelligence algorithms were analyzed since they allow operators to examine the relationships established among the various variables of the problem in every individual stage of elaboration. Moreover, they require low computational complexity and low need for computational resources to minimize system latency.

Swarm Intelligence algorithms were identified as the best-performing ones for prediction error and computational effort by comparing the results. In particular, the Grey Wolf optimizer proved to be the best one among those used. The ease of implementation and use of the approach based on SI algorithms guarantees the possibility to realize the Digital Twin without problems. Moreover, its computational lightness allows for the improvement of the model continuously and almost instantaneously after new acquisitions.

During the test in the lab, it was possible to highlight the advantages and disadvantages of the use of the two methodologies.

Thanks to their structure, ANNs can work in parallel to process a lot of data while, in traditional calculators, each datum is processed individually and in succession. However, if some system units were to malfunction, the whole network would have performance reductions but would hardly meet with a blockage. Likewise, for SI algorithms, multiple agent systems can be easily parallelized so that large-scale optimization becomes more practical and faster from the implementation point of view. This property defines the ability of agents to perform a multitude of actions in different locations at the same time. It is fundamental, as it allows for the development of more flexible systems capable of self-organizing into groups that simultaneously consider different aspects of a particular problem. The multilayer perceptron has a nonconvex loss function in the presence of multiple local minima, highlighting how, in these cases, different random initializations of the weights can lead to different validation accuracies. Furthermore, the multilayer perceptron is related to numerous hyperparameters (the number of hidden neurons, layers, and iterations) and is sensitive to scaling characteristics.

By contrast, the search for an optimal solution in Swarm Intelligence does not rely on derived functions but on different social interaction mechanisms between artificial individuals.

Among the disadvantages of swarm intelligence, it is possible to mention premature convergence. Some types of SI (such as PSO) usually suffer from premature convergence when multiple optimization problems occur. The basis of this problem is that the particles converge at a single point, which is on the line between the best global positions and the best personal positions. In this way, the chances of being trapped in the local minimum are significantly reduced. The second disadvantage of SI algorithms is related to their sensitivity to the setting of various parameters. For example, increasing the value of inertia weight, w (in the PSO), increases the particle velocity resulting in increased exploration (global research) and lower concentration (local research) and vice versa. Setting parameters is therefore not an easy task and varies from problem to problem.

In this context, further research should focus on developing less sensitive approaches to the data provided with the same reliability as the results obtained.

Author Contributions: Conceptualization, G.M. and F.E.C.; methodology, F.E.C.; software, M.O.; formal analysis, G.M. and M.O.; supervision, M.B.; project administration, M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by INAIL (Istituto Nazionale per l'Assicurazione Contro gli Infortuni sul Lavoro), the Italian National Institute for Insurance against Accidents at Work, under the BRIC 2018 project titled "Sviluppo di soluzioni smart attraverso metodologie Digital Twin per aumentare la sicurezza degli operatori durante i processi di manutenzione degli impianti produttivi"—BRIC ID12.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript.

Glossary

Acronym	Description	Acronym	Description
PresLiq	Pressure of inlet water flow	P _{diff}	Diffuser pressure
PresSerb	Internal tank pressure	Q _{liq}	Inlet water flow rate
Q _{gas}	Actual inlet air flow rate	Q* _{gas}	Esteemed inlet air flow rate
P _{gas}	Pressure of inlet airflow	Patm	Atmospheric pressure
D1		D2	
nPop	Number of agents for GWO	MaxIter	Maximum number of iterations for GWO
n	Simulation number (*)	C	Bool value to identify the use of intercept (*)
f	Type of function (*)	nP	Number of agents for GWO (*)
NN	Neural network	SI	Swarm intelligence
T1	Test 1 (*)	T2	Test 2 (*)
T3	Test 3 (*)	T4	Test 4 (*)
MI	Maximum number of iteration for GWO (*)	MEAN	Mean value
STD DEV	Standard deviation	RSD	Relative standard deviation
α	Significance level	H0	Null hypothesis
p-v	p-value (*)	T-v	T-value (*)
CI	Confidence interval (*)	SD	Standard deviation (*)
DF	Degree of freedom (*)	R	Pearson indicator
RI	Person indicator lover bound (*)	Ru	Pearson indicator upper bound (*)

(*) These symbols have been adopted to improve the readability of the tables.

Appendix A. List of Function

Table A1. List of functions used to determine the simulated ejector model.

Function	f(x)	Intercept	Description
Linear	1	false	$b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + b_4 * x_4$
Linear	1	true	$b_1 + b_2 * x_1 + b_3 * x_2 + b_4 * x_3 + b_5 * x_4$
Simple quadratic	2	false	$b_1 * x_1^2 + b_2 * x_2^2 + b_3 * x_3^2 + b_4 * x_4^2$
Simple quadratic	2	true	$b_1 + b_2 * x_1^2 + b_3 * x_2^2 + b_4 * x_3^2 + b_5 * x_4^2$
Structured quadratic	3	false	$b_1 * x_1^2 + b_2 * x_2^2 + b_3 * x_3^2 + b_4 * x_4^2 + b_5 * x_1 * x_2 + b_6 * x_1 * x_3 + b_7 * x_2 * x_3 + b_8 * x_1 * x_4 + b_9 * x_2 * x_4 + b_{10} * x_3 * x_4$
Structured quadratic	3	true	$b_1 + b_2 * x_1^2 + b_3 * x_2^2 + b_4 * x_3^2 + b_5 * x_4^2 + b_6 * x_1 * x_2 + b_7 * x_1 * x_3 + b_8 * x_2 * x_3 + b_9 * x_1 * x_4 + b_{10} * x_2 * x_4 + b_{11} * x_3 * x_4$
Simple cubic	4	false	$b_1 * x_1^3 + b_2 * x_2^3 + b_3 * x_3^3 + b_4 * x_4^3$
Simple cubic	4	true	$b_1 * 1 + b_2 * x_1^3 + b_3 * x_2^3 + b_4 * x_3^3 + b_5 * x_4^3$
Structured cubic	5	false	$b_1 * x_1^3 + b_2 * x_2^3 + b_3 * x_3^3 + b_4 * x_4^3 + b_5 * x_1^2 * x_2 + b_6 * x_1 * x_2^2 + b_7 * x_1^2 * x_3 + b_8 * x_1 * x_2 * x_3 + b_9 * x_2^2 * x_3 + b_{10} * x_1 * x_3^2 + b_{11} * x_2 * x_3^2 + b_{12} * x_1^2 * x_4 + b_{13} * x_1 * x_2 * x_4 + b_{14} * x_2^2 * x_4 + b_{15} * x_1 * x_3 * x_4 + b_{16} * x_2 * x_3 * x_4 + b_{17} * x_3^2 * x_4 + b_{18} * x_1 * x_4^2 + b_{19} * x_2 * x_4^2 + b_{20} * x_3 * x_4^2$
Structured cubic	5	true	$b_1 * x_1^3 + b_2 * x_2^3 + b_3 * x_3^3 + b_4 * x_4^3 + b_5 * x_1^2 * x_2 + b_6 * x_1 * x_2^2 + b_7 * x_1^2 * x_3 + b_8 * x_1 * x_2 * x_3 + b_9 * x_2^2 * x_3 + b_{10} * x_1 * x_3^2 + b_{11} * x_2 * x_3^2 + b_{12} * x_1^2 * x_4 + b_{13} * x_1 * x_2 * x_4 + b_{14} * x_2^2 * x_4 + b_{15} * x_1 * x_3 * x_4 + b_{16} * x_2 * x_3 * x_4 + b_{17} * x_3^2 * x_4 + b_{18} * x_1 * x_4^2 + b_{19} * x_2 * x_4^2 + b_{20} * x_3 * x_4^2 + b_{21}$
Logarithmic	6	false	$b_5 * \exp(b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + b_4 * x_4)$
Logarithmic	6	true	$b_6 + b_5 * \exp(b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + b_4 * x_4)$
Exponential	7	false	$b_1 * \log(x_1) + b_2 * \log(x_2) + b_3 * \log(x_3) + b_4 * \log(x_4)$
Exponential	7	true	$b_1 * 1 + b_2 * \log(x_1) + b_3 * \log(x_2) + b_4 * \log(x_3) + b_5 * \log(x_4)$
Power	8	false	$x_1^{b_1} + x_2^{b_2} + x_3^{b_3} + x_4^{b_4}$
Power	8	true	$b_1 + x_1^{b_1} + x_2^{b_2} + x_3^{b_3} + x_4^{b_4}$

References

1. Guo, J.; He, Y.; Zheng, X. Pangu: Towards a Software-Defined Architecture for Multi-function Wireless Sensor Networks. In Proceedings of the 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS), Shenzhen, China, 15–17 December 2017; pp. 730–737. [\[CrossRef\]](#)
2. Yang, L.; Li, Y.; Lin, Q.; Li, X.-Y.; Liu, Y. Making sense of mechanical vibration period with sub-millisecond accuracy using backscatter signals. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, New York, NY, USA, 3–7 October 2016; pp. 16–28. [\[CrossRef\]](#)
3. Habib, G.; Haddad, N.; El Khoury, R. Case study: WIRELESSHART vs. ZIGBEE network. In Proceedings of the 2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE), Beirut, Lebanon, 29 April–1 May 2015; pp. 135–138. [\[CrossRef\]](#)
4. Hancke, G.; Allen, B. Ultrawideband as an Industrial Wireless Solution. *IEEE Pervasive Comput.* **2006**, *5*, 78–85. [\[CrossRef\]](#)
5. Ryou, S.; Rahmani, A.R.; Yoon, K.Y.; Prodanović, M.; Kotsmar, C.; Milner, T.E.; Johnston, K.P.; Bryant, S.L.; Huh, C. Theoretical and experimental investigation of the motion of multiphase fluids containing paramagnetic nanoparticles in porous media. *J. Pet. Sci. Eng.* **2012**, *81*, 129–144. [\[CrossRef\]](#)
6. Narimani, E.; Sorin, M.; Micheau, P.; Nesreddine, H. Dynamic modeling of an R245fa ejector based refrigeration system. *Int. J. Refrig.* **2019**, *107*, 262–274. [\[CrossRef\]](#)
7. Galanis, N.; Sorin, M. Ejector design and performance prediction. *Int. J. Therm. Sci.* **2016**, *104*, 315–329. [\[CrossRef\]](#)
8. Kim, M.I.; Kim, O.S.; Lee, D.H.; Kim, S.D. Numerical and experimental investigations of gas–liquid dispersion in an ejector. *Chem. Eng. Sci.* **2007**, *62*, 7133–7139. [\[CrossRef\]](#)
9. Li, C.; Li, Y. Investigation of entrainment behavior and characteristics of gas–liquid ejectors based on CFD simulation. *Chem. Eng. Sci.* **2011**, *66*, 405–416. [\[CrossRef\]](#)
10. Zheng, P.; Li, B.; Qin, J. CFD simulation of two-phase ejector performance influenced by different operation conditions. *Energy* **2018**, *155*, 1129–1145. [\[CrossRef\]](#)
11. Mazzelli, F.; Giacomelli, F.; Milazzo, A. CFD modeling of condensing steam ejectors: Comparison with an experimental test-case. *Int. J. Therm. Sci.* **2018**, *127*, 7–18. [\[CrossRef\]](#)
12. Zhu, Y.; Cai, W.; Wen, C.; Li, Y. Shock circle model for ejector performance evaluation. *Energy Convers. Manag.* **2007**, *48*, 2533–2541. [\[CrossRef\]](#)
13. Schluse, M.; Rossmann, J. From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems. In Proceedings of the 2016 IEEE International Symposium on Systems Engineering (ISSE), Edinburgh, UK, 3–5 October 2016; pp. 1–6. [\[CrossRef\]](#)
14. Weyer, S.; Meyer, T.; Ohmer, M.; Gorecky, D.; Zühlke, D. Future Modeling and Simulation of CPS-based Factories: An Example from the Automotive Industry. *IFAC-PapersOnLine* **2016**, *49*, 97–102. [\[CrossRef\]](#)
15. Schluse, M.; Priggemeyer, M.; Atorf, L.; Rossmann, J. Experimentable Digital Twins—Streamlining Simulation-Based Systems Engineering for Industry 4.0. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1722–1731. [\[CrossRef\]](#)
16. Seman, L.O.; Miyatake, L.K.; Camponogara, E.; Giuliani, C.; Vieira, B.F. Derivative-free parameter tuning for a well multiphase flow simulator. *J. Pet. Sci. Eng.* **2020**, *192*, 107288. [\[CrossRef\]](#)
17. Kanin, E.; Osipov, A.; Vainshtein, A.; Burnaev, E. A predictive model for steady-state multiphase pipe flow: Machine learning on lab data. *J. Pet. Sci. Eng.* **2019**, *180*, 727–746. [\[CrossRef\]](#)
18. Zhu, Y.; Jiang, P. Experimental and numerical investigation of the effect of shock wave characteristics on the ejector performance. *Int. J. Refrig.* **2014**, *40*, 31–42. [\[CrossRef\]](#)
19. Allouche, Y.; Bouden, C.; Varga, S. A CFD analysis of the flow structure inside a steam ejector to identify the suitable experimental operating conditions for a solar-driven refrigeration system. *Int. J. Refrig.* **2013**, *39*, 186–195. [\[CrossRef\]](#)
20. Li, F.; Tian, Q.; Wu, C.; Wang, X.; Lee, J.-M. Ejector performance prediction at critical and subcritical operational modes. *Appl. Therm. Eng.* **2017**, *115*, 444–454. [\[CrossRef\]](#)
21. Wu, Y.; Zhao, H.; Zhang, C.; Wang, L.; Han, J. Optimization analysis of structure parameters of steam ejector based on CFD and orthogonal test. *Energy* **2018**, *151*, 79–93. [\[CrossRef\]](#)
22. Farmani, Z.; Azin, R.; Fatehi, R.; Escrochi, M. Analysis of Pre-Darcy flow for different liquids and gases. *J. Pet. Sci. Eng.* **2018**, *168*, 17–31. [\[CrossRef\]](#)
23. Fu, W.; Li, Y.; Liu, Z.; Wu, H.; Wu, T. Numerical study for the influences of primary nozzle on steam ejector performance. *Appl. Therm. Eng.* **2016**, *106*, 1148–1156. [\[CrossRef\]](#)
24. Quiroz-Pérez, E.; Vázquez-Román, R.; Lesso-Arroyo, R.; Barragán-Hernández, V.M. An approach to evaluate Venturi-device effects on gas wells production. *J. Pet. Sci. Eng.* **2014**, *116*, 8–18. [\[CrossRef\]](#)
25. Nasseh, S.; Mohebbi, A.; Jeirani, Z.; Sarrafi, A. Predicting pressure drop in venturi scrubbers with artificial neural networks. *J. Hazard. Mater.* **2007**, *143*, 144–149. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Shippen, M.E.; Scott, S.L. A Neural Network model for Prediction of Liquid Holdup in Two-Phase Horizontal Flow. In Proceedings of the SPE Annual Technical Conference and Exhibition, San Antonio, TX, USA, 29 September–2 October 2002. [\[CrossRef\]](#)
27. Osman, E.-S.A. Artificial Neural Network Models for Identifying Flow Regimes and Predicting Liquid Holdup in Horizontal Multiphase Flow. *SPE Prod. Facil.* **2004**, *19*, 33–40. [\[CrossRef\]](#)

28. Sözen, A.; Arcaklıoğlu, E. Exergy analysis of an ejector-absorption heat transformer using artificial neural network approach. *Appl. Therm. Eng.* **2007**, *27*, 481–491. [[CrossRef](#)]
29. Haoran, C.; Wenjian, C. Artificial neural network modeling for variable area ratio ejector. In Proceedings of the 2014 9th IEEE Conference on Industrial Electronics and Applications, Hangzhou, China, 9–11 June 2014; pp. 220–225. [[CrossRef](#)]
30. Al-Naser, M.; Elshafei, M.; Al-Sarkhi, A. Artificial neural network application for multiphase flow patterns detection: A new approach. *J. Pet. Sci. Eng.* **2016**, *145*, 548–564. [[CrossRef](#)]
31. Wang, H.; Cai, W.; Wang, Y. Modeling of a hybrid ejector air conditioning system using artificial neural networks. *Energy Convers. Manag.* **2016**, *127*, 11–24. [[CrossRef](#)]
32. Eberhart, R.C.; Hu, X. Human tremor analysis using particle swarm optimization. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; pp. 1927–1930. [[CrossRef](#)]
33. Mazzuto, G.; Stylios, C. Empower Fuzzy Cognitive Maps Decision Making abilities with Swarm Intelligence Algorithms. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; pp. 2602–2607. [[CrossRef](#)]
34. Karaboga, D.; Ökdem, S.; Ozturk, C. Cluster based wireless sensor network routings using Artificial Bee Colony Algorithm. In Proceedings of the 2010 International Conference on Autonomous and Intelligent Systems, AIS 2010, Povo de Varzim, Portugal, 21–23 June 2010; pp. 1–5. [[CrossRef](#)]
35. Fuellerer, G.; Doerner, K.; Hartl, R.F.; Iori, M. Ant colony optimization for the two-dimensional loading vehicle routing problem. *Comput. Oper. Res.* **2009**, *36*, 655–673. [[CrossRef](#)]
36. Ma, Q.; Lei, X. Application of artificial fish school algorithm in UCAV path planning. In Proceedings of the 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), Changsha, China, 23–26 September 2010; pp. 555–559. [[CrossRef](#)]
37. Spiess, A.-N.; Neumeyer, N. An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: A Monte Carlo approach. *BMC Pharmacol.* **2010**, *10*, 6. [[CrossRef](#)]
38. Floreano, D.; Mattiussi, C. «Manuale sulle Reti Neurali», Infoscience. 2002. Available online: <https://infoscience.epfl.ch/record/63947> (accessed on 3 July 2020).
39. Wu, D.; Xu, S.; Kong, F. Convergence Analysis and Improvement of the Chicken Swarm Optimization Algorithm. *IEEE Access* **2016**, *4*, 9400–9412. [[CrossRef](#)]
40. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [[CrossRef](#)]
41. Yang, X.-S.; He, X. Bat algorithm: Literature review and applications. *Int. J. Bio-Inspired Comput.* **2013**, *5*, 141. [[CrossRef](#)]
42. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [[CrossRef](#)]
43. Bhargava, V.; Fateen, S.; Bonilla-Petriciolet, A. Cuckoo Search: A new nature-inspired optimization method for phase equilibrium calculations. *Fluid Phase Equilibria* **2013**, *337*, 191–200. [[CrossRef](#)]
44. Yang, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78. [[CrossRef](#)]
45. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
46. Neshat, M.; Sepidnam, G.; Sargolzaei, M.; Toosi, A.N. Artificial fish swarm algorithm: A survey of the state-of-the-art, hybridization, combinatorial and indicative applications. *Artif. Intell. Rev.* **2012**, *42*, 965–997. [[CrossRef](#)]
47. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [[CrossRef](#)]
48. Bhattacharyya, B.; Babu, R. Teaching Learning Based Optimization algorithm for reactive power planning. *Int. J. Electr. Power Energy Syst.* **2016**, *81*, 248–253. [[CrossRef](#)]
49. Zaránd, P.; Polgár, I. A theoretical study on the relative standard deviation of TLD systems. *Nucl. Instruments Methods Phys. Res.* **1983**, *205*, 525–529. [[CrossRef](#)]