Applied Soft Computing Journal 89 (2020) 106103

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

A general framework and guidelines for benchmarking computational intelligence algorithms applied to forecasting problems derived from an application domain-oriented survey

Mihaela Oprea

Department of Automatic Control, Computers and Electronics, Petroleum-Gas University of Ploiesti, Bdul Bucuresti No 39, Ploiesti, 100680, Romania

ARTICLE INFO

Article history:
Received 20 May 2019
Received in revised form 10 January 2020
Accepted 13 January 2020
Available online 23 January 2020

Keywords:
Benchmarking computational intelligence algorithms
Knowledge-based general benchmarking framework
Forecasting applications
Computational intelligence engineering guidelines

ABSTRACT

Benchmarking computational intelligence algorithms provides valuable knowledge for selecting the best or, at least, the proper algorithm for a certain problem. The experimental results of the computational intelligence techniques applications in various domains, as well as the comparative studies that were reported in the literature can be analyzed and synthesized as development strategies for new successful applications of CI algorithms. Starting from an application domain-oriented survey of selected recently reported research work, the paper presents a general benchmarking framework applicable to computational intelligence algorithms and a set of guidelines for the selection of the best or more suitable CI algorithm for solving forecasting problems. Our approach proposes the integration of software and knowledge engineering best practice towards CI benchmarking, being a computational intelligence engineering methodology. The framework uses two knowledge bases, one for the application domain and one for the CI algorithms, providing heuristic knowledge for a more informed and efficient benchmarking, a case base in which solved problems are recorded with their solution and lessons that were learned, and a knowledge-based problem instance features selection. Some examples of how to apply the framework for problems of forecasting in seismology, environmental protection, hydrology and energy are also discussed. We point out that the framework might be implemented as a software tool (e.g. a decision support system) or as a tool suite. The main conclusion of our research work is that the integration of the derived knowledge from an application domain-oriented survey into the general benchmarking framework along with the set of guidelines for best or proper CI algorithms selection can improve significantly the forecasting accuracy and the response time, in case of real time forecasters.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

In the last two decades, computational intelligence (CI) had a rapid development and became a prolific subfield of artificial intelligence that provided a variety of techniques i.e. the traditional ones (artificial neural networks, fuzzy logic systems, evolutionary algorithms etc.), permanently improved and the new ones (e.g. swarm intelligence), inspired from nature or other sources, with applications in industry, environmental protection, meteorology, energy, finance, economy, geology, seismology, hydrology, chemistry, manufacturing, robotics, medicine, sociology (see some examples in [1–3]), solving different types of problems (see [4] for a recent survey). Among these, forecasting is a challenging problem type, in terms of accuracy improvements and real time response. For example, in natural hazards forecasting it is extremely important to have an accurate and real time forecast in order to reduce the potential damages. Another example is green energy (e.g. wind or solar) forecasting where accurate and real time forecasting is valuable not only for a sustainable environment (by reduced pollution) but also for cases when alternative sources of non-conventional energy are needed in the energy grid.

Various surveys highlighted the current trends and challenges of using CI algorithms for solving forecasting problems in specific domains, with better accuracy. A recent extensive survey in the hydrology domain [5] emphasized the major trends for flood prediction model accuracy improvement: hybridization, data decomposition, algorithm ensemble, and model optimization. The comparative study of different forecasting models (statistical, physical, machine learning, climatology) applied as single, hybrid or ensemble models, revealed that single prediction models are good for short-term forecast (up to 2 h), while hybrid models are good for predictions longer than 2 h. Another recent survey in the same domain [6] concluded that ensemble of CI approaches are highly efficient for flood prediction (e.g. for early flood event

E-mail address: mihaela@upg-ploiesti.ro.

https://doi.org/10.1016/j.asoc.2020.106103
1568-4946/© 2020 Elsevier B.V. All rights reserved.

detection with a low false alarm rate). Finally, a recent survey on energy load forecasting in smart energy management grids with CI algorithms [7], emphasized the state of the art, future challenges and research directions. One of the main conclusion of this survey is that hybrid methods provide higher accuracy and robustness of prediction. A current trend that was noticed in these surveys is the use of deep learning, wavelet transform and metaheuristics for a better forecasting accuracy. On the other hand, the conclusion of other surveys that performed a comparative study between statistical methods and CI methods (in particular, machine learning methods) showed mixed results (i.e. with good results of both types of methods, statistical and machine learning, as in [8]) or a lower forecasting accuracy of machine learning models (as in [9]), but with a recent promising result of the last M4-Competition (http://www.m4.unic.ac.cy) that run in 2018 for which the winner was a hybrid method ES-RNN (Exponential Smoothing-Recurrent Neural Network) [10].

Thus, selecting the best forecasting method for a certain application domain is a time consuming and difficult task, especially when a real time response is required. A solution would be a more informed benchmarking, including for example, the successful CI-based forecasting methods applied to that domain. By analyzing the comparative studies described in the literature, new knowledge related to the performance of CI algorithms can be generated. Starting from this idea, the paper presents a general framework for CI algorithms benchmarking and a set of guidelines for the selection of the best or more suitable CI algorithm applied to forecasting problems, based on an application domain-oriented survey of selected papers reported in the literature. In its current form, the proposed framework is a meta-model that integrates software and knowledge engineering best practice towards CI benchmarking being a practical guide that can be followed by researchers and engineers that develop CI-based forecasting applications. The main novelties of the framework (which is a knowledge-based benchmarking framework for solving forecasting problems) are: (1) the use of two knowledge bases, one for the application domain and one for the CI algorithms, providing heuristic knowledge for more informed and efficient benchmarking of specific forecasting problem instance solving methods; (2) the use of a case base in which solved problems are recorded with their solution and learnt lessons (i.e. derived knowledge that can be included also in the corresponding two knowledge bases); (3) the inclusion of feature selection guided by expert knowledge (application domain specific and general).

The paper is structured as follows. Section 2 presents an overview of CI algorithms and a survey of CI-based forecasting applications developed in some domains (seismology, environmental protection, hydrology, energy, materials science, engineering etc.). The general CI algorithms benchmarking framework is described in Section 3, followed by a synthesized set of guidelines for selecting CI algorithms proper for solving forecasting problems and some issues related to framework implementation. Some examples of how to apply the framework guidelines for solving problems of forecasting in seismology, air pollution, hydrology and energy are presented in Section 4. Last section concludes the paper and highlights some future work.

2. An overview on computational intelligence algorithms and selected applications

Computational intelligence can be defined as a subfield of Computer Science that provides solutions to highly complex problems (e.g., NP-hard), most of which involving significant uncertainty, for which there are no effective computational algorithms [1]. At present, there is no (unique) definition of CI, widely accepted by the academic community. However, all definitions agree on the inclusion of Fuzzy, Neural and Evolutionary computation methods (i.e. the traditional CI methods), while many of them include also machine learning methods, probabilistic methods, swarm intelligence methods, other nature-inspired methods, even some mathematical/statistical methods. Some researchers accept the synonymy of CI with Soft Computing (SC), while others do not. The reality is that CI is a challenging research field under a continuous and rapid development, to which a variety of methods from different domains (physics, chemistry, biology, engineering, mathematics, statistics etc.) [3] or nature-inspired [2] are frequently added. Moreover, the applicability of CI algorithms is almost in any domain. A simple search on the internet of research papers with the "Computer Intelligence algorithms and applications" subject recently published will return various new, improved or hybrid CI methods for different applications domains. The main problem of this quick expansion of CI with a variety of methods and applications is the lack of a deep understanding of the methods, of their main benefits as well as drawbacks. We consider that a possible solution would be knowledge extraction from periodically surveys on specific domains, i.e. from the experience of the research work reported in the literature (that can be replicated or verified on new test cases) some lessons can be learnt and applicable to future work and thus, the current amount of papers already reported in the literature for those domains can be exploited at maximum. Therefore, the maturity of the CI domain and its applicability extension within the communities of application domains specialists can be proved by a well-documented theory and practice, based on the experience of real world problem solving in their domains, including comparative studies (benchmarking) between CI methods and other methods (such as statistical and domain specific methods).

We point out that throughout the paper, we consider the following meaning of the terms: *problem type* is the type of the general problem that is solved (e.g. forecasting), *problem* is the general problem (e.g. air pollution forecasting) and *problem instance* is a specific problem (e.g. air pollution forecasting in the Ploiesti city area), i.e. a particularization of the general problem to a certain context/area/region with specific datasets, i.e. instance data (e.g. the hourly measurements of specific air pollutants concentrations in the Ploiesti city area, taken from the air quality monitoring stations existing in the Ploiesti city, during a certain period of time).

Solving a problem instance can be tackled with three main goals, when taking into account the CI approaches alternative: the problem type most appropriate method (e.g. the most appropriate forecasting method), the application domain most appropriate method and the most appropriate CI method. Fig. 1 shows an overview of these goals. In reality, this strict delimitation between the three goals can be true or not.

Thus, we can have three perspectives on solving a problem, provided by the problem type expert, the application domain expert and the CI expert. The problem type expert or specialist (e.g. in problems such as optimization, decision making, forecasting) will choose the most appropriate theoretical/practical method according to a certain criteria. The application domain expert will adopt the most appropriate domain specific method according to domain specific criteria. Finally, the CI expert will choose the most appropriate CI method according to particular criteria. A benchmark of these most appropriate methods will provide the most appropriate method (possible, the best method) to solve the problem instance, which can be a single or hybrid or ensemble method.

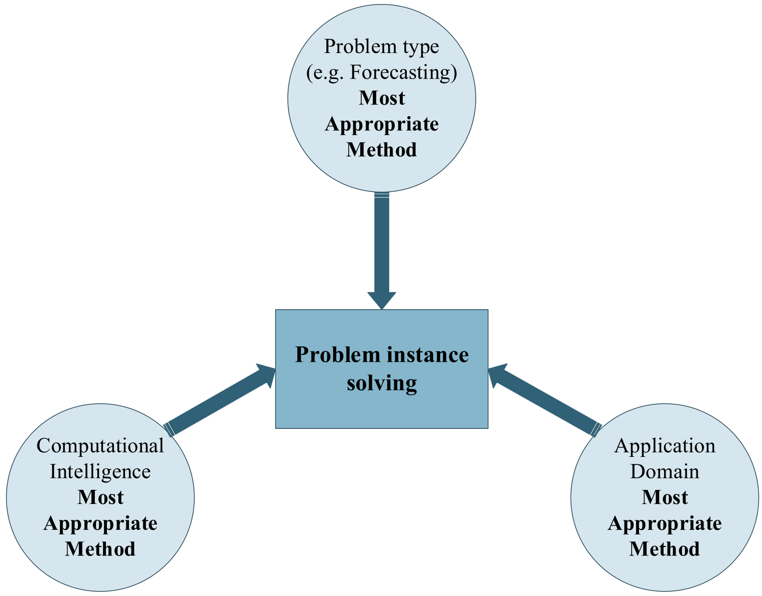

Fig. 1. An overview of the three goals of problem instance solving.

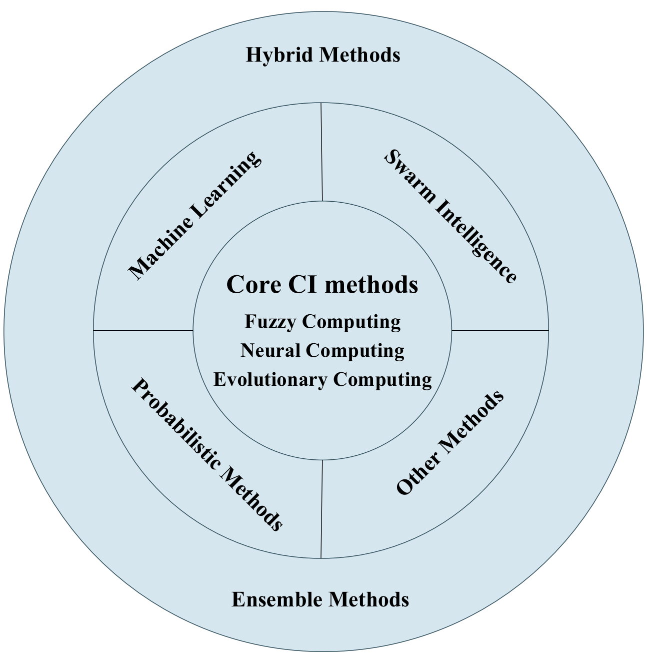

Fig. 2. The main types of CI methods.

2.1. CI algorithms overview

The CI methods include a core part with the traditional CI methods (fuzzy, neural and evolutionary computing methods) and the added classes of single methods such as machine learning, probabilistic methods, swarm intelligence, nature-inspired, other methods, and a variety of hybrid and ensemble methods. Fig. 2 shows the main types of CI methods.

Therefore, the core CI methods include fuzzy computing (fuzzy logic and fuzzy sets), neural computing (artificial neural networks), and evolutionary computing (genetic algorithms, genetic programming etc.). Other methods are machine learning (ML) methods, probabilistic methods, metaheuristic methods such as swarm intelligence and nature-inspired methods. Also, various hybrid CI methods (e.g. combinations or ensembles of CI methods) have been reported in the literature with better results than single CI methods. Examples of hybrid methods are ANFIS (Adaptive Neuro-Fuzzy Inference System) and WNN (Wavelet Neural Networks) etc. Each CI method has a certain corresponding algorithm with a set of specific parameters that need to be set or tuned in order to obtain a better performance when solving a problem instance.

A brief description of some CI algorithms is presented as follows.

2.1.1. Core CI methods

2.1.1.1. Fuzzy computing. Fuzzy sets [11] were introduced by Zadeh as a generalized classical set theory that can manipulate linguistic (fuzzy) terms expressing uncertainty. Each fuzzy term is defined by a membership function. Fuzzy logic (FL) and fuzzy inference systems (FISs) were derived from the theory of fuzzy sets. A fuzzy system is able to decide on uncertain information. A fuzzy logic model maps a set of input variables to a set of output variables using IF-THEN rules. FL is used for approximate reasoning and has no learning abilities. A major advantage of FL is the possibility of decision making for uncertain information. The main problem of the FL method is rule tuning and one disadvantage is drawback of cognitive uncertainties.

2.1.1.2. Neural computing. Artificial neural networks (ANNs) [12] are universal approximators of non-linear functions. They are composed by a number of artificial neurons (non-linear processing units) which are structured in layers. An artificial neuron has a number of inputs, each one with a specific weight, an activation function, a bias, and one output. Depending on the way are linked and structured the artificial neurons, several types of ANN architectures (topologies) can be obtained: feed forward ANN (FFANN), recurrent ANN (RNN), radial basis function ANN (RBFANN), self-organizing maps (SOM) etc. For example, FFANN has one input layer, one output layer and a number of hidden layers. The number of nodes in the input layer and the number of nodes in the output layer are set according to the input and output data given in the problem. The number of hidden layers and the number of hidden nodes in each hidden layer are determined by experiments. Some informed guidelines can be taken from the literature as e.g. in [13] where it is suggested that a maximum value for the number of hidden nodes based on the Kolmogorov's theorem can be $n_h <= 2\ n_i + 1$, where n_h is the number of hidden nodes, and n_i is the number of input nodes. An ANN is trained, i.e. the weights are determined during ANN training via a learning algorithm as e.g. backpropagation (BP), Levenberg–Marquardt (LM), under which the error is minimized. An important theorem is given in [14], the universal approximation theorem saying that a single hidden layer ANN is sufficient for uniformly approximating any nonlinear and continuous function. The ANN performance is highly dependent on the training sample size and information quality. The main advantages of ANNs are their accuracy, high fault tolerance and powerful parallel processing ability. The major disadvantages of ANNs are overfitting, subject to local convergence slow learning, and the network's tendency to become trapped in local minima. One solution to this last disadvantage is to train the ANN with a global optimization algorithm (e.g. genetic algorithms or simulated annealing).

2.1.1.3. Evolutionary computing. Evolutionary computation (EC) is composed of a variety of evolutionary methods such as genetic algorithms, genetic programming, evolutionary strategies, differential evolution etc. Such methods perform a global search and are population-based methods. They involve an iteratively updated set of trial solutions grouped in generations. Solutions of the same generation are ranked by their fitness, and the fittest part of the generation are allowed to produce the next generation by means of variation operators such as crossover operators (uniform, in one/two points) and mutations.

Genetic algorithms (GA) [15] were introduced by Holland as a population-based method for optimization problems solving. GA finds nearly optimal solution via an iterative process of applying the genetic operators: selection, mutation, crossover to a population of individuals, minimizing the fitness function defined by the problem instance [16]. The population initialization can be done via random procedures or more informed ones as e.g. by following the guidelines for seeding the initial population of multi-objective evolutionary algorithms given in [17] or following the lessons learned in the study of GA optimization performance evaluation described in [18].

Genetic programming (GP) [19] was introduced by J. R. Koza in 1992 as a generalization of GA to populations of programs. GP performs an optimal search of the nearly optimal solution.

Differential evolution (DE) [20] is a vector-based evolutionary algorithm which is more efficient than GA in many applications.

Other CI methods include metaheuristics taken from other domains: simulated annealing, swarm intelligence methods, natural-inspired methods, other machine learning methods such as SVM and random forest, and hybrid methods. We have selected a set of most used such methods and present them in the next three sections.

2.1.2. Other metaheuristics

Simulated annealing (SA) [21] is a trajectory based metaheuristic that mimics the annealing process of metal which cools and freezes into a crystalline state with minimum energy and larger crystal size in order to minimize the metallic structures defects. It is a global search method. The annealing process involves the careful process of temperature and its cooling rate, often called the annealing schedule. It has been proved that SA will converge to its global optimality if enough randomness is used in combination with very slow cooling. An important issue for the SA performance is the choice of the right initial temperature. The commonly used annealing schedules (or cooling schedules) are linear and geometric.

Swarm intelligence (SI) [2] is a branch of biologically inspired algorithms that mimics the collective behavior of swarms such as: ant colonies, particle swarms, bee colonies, mosquito swarms, birds, fish schools, flies, cuckoos etc. These swarms have a collective behavior of decentralized, self-organized systems. The local interactions lead to the emergence of a complex global behavior. Examples of metaheuristics from SI are Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Wasp swarm optimization (WSP).

In 1990, Marco Dorigo introduced the ACO technique [22] for solving optimization problems in a similar way to the natural behavior of ant colonies. Artificial ants search for good solutions to a given optimization problem that is transformed into the problem of finding the best path on a weighted graph. Ants incrementally build solutions by moving on the graph.

The PSO technique was presented first in [23] as a population based search algorithm that performs a stochastic global optimization by generating several initial particles with different velocity that move toward the best area. It was inspired from the behavior of fishes and birds swarms. PSO has a faster convergence than GA and few parameters to adjust. Also, it can be implemented quite easily for real world problem solving.

The ABC technique [24] was introduced by Karaboga in 2005 and it is inspired by the intelligent foraging manners of a honeybee swarm with three groups of bees: employed bees (a bee going to the food source visited by itself previously), onlookers (a bee waiting on the dance area for making decision to choose a food source), scouts (a bee carrying out random search — an employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout). A comprehensive survey on the ABC algorithm and its applications is presented in [24]. A recent overview on the last advances and applications of ABC is described in [25].

2.1.3. Other machine learning methods

Support Vector Machine (SVM) [26] is a machine learning method defined as a set of powerful supervised learning techniques used for classification and regression. Their basic principle is to construct a maximum margin separating hyperplane in some transformed feature space [27]. They use the principle of kernel substitution to turn them into a general (non-linear) model, rather than requiring the specification of an exact transformation. SVM has as feature structural risk minimization (SRM). It minimizes the upper bound of the generalization error instead of minimizing the training error, thus improving the generalization performance and overcoming the ANN shortcomings. The SVM model can perform quite accurately for small sized datasets. The main drawback of SVM is the longer time for optimum parameter finding. SVM approaches include support vector regression (SVR) [28] and the Least Square SVM (LSSVM) [29].

Random Forest (RF) [30] is a group of un-pruned classification or regression trees, trained on bootstrap samples of the training data using random feature selection in the process of tree generation. After a large number of trees have been generated, each tree votes for the most popular class. These tree voting procedures are collectively defined as random forests. Two parameters have to be tuned in this case: the number of trees and the number of attributes used to grow each tree.

2.1.4. Hybrid methods

Several hybrid methods were proposed in the literature (see e.g. [1,4]). Such methods combine different methods (CI or not-CI based).

Adaptive Neuro-Fuzzy Inference System (ANFIS) [31] was introduced in 1993 by Jang. It combines fuzzy inference systems (FIS) with ANNs by applying the ANN learning method to FISs. It can perform better than ANN in case of peak modeling. The main principle is to combine fuzzy classification with ANN in order to find the best parameters by arranging the rule base.

Wavelet Neural Networks (WNNs) [32] combine ANN with wavelet transform (WT) that is a powerful tool for analysis of non-stationary signals in time-frequency domains. WT is able to decompose time series data into different level containing low frequency and high frequency components. The transformed normalized data can be fed to an ANN where features of individual components can be learned.

2.1.5. Ensemble methods

The concept of ensemble learning was defined in [33] and basically, involves the use of a set of ML techniques for finding a better solution of the problem instance.

Table 1 presents the main parameters of some CI methods that were briefly discussed.

2.2. A survey of CI applications for prediction/forecasting problem solving

2.2.1. Forecasting problem formulation

According to [34], forecasting can be defined as "*predicting the future as accurately as possible given all the information available, including historical data and knowledge of any future events that might impact the forecasts*". Depending on what data is available, there are two main types of forecasting: quantitative forecasting (when data is available) and qualitative forecasting (when no data is available). In this research work we focus on quantitative forecasting with its particular case of time series forecasting [35] that uses historical data collected at regular intervals over time (e.g. hourly, daily, weekly, monthly, quarterly, yearly).

A general formulation of the forecasting problem and two types of formal models associated to it are presented.

Suppose x is the parameter we want to forecast (i.e. to predict its future values with accuracy). The forecasting problem can be generally formulated as follows.

Given a set of data and information (e.g. observations) related to a phenomenon, event or process from a domain, find with a good precision the next k values of a parameter x, $\{ x_{t+1}, x_{t+2}, \ldots, x_{t+k} \}$ that influence or characterize that phenomenon, event or process. The datasets can contain time series or values collected at a single point in time.

The forecasting problem can be formally modeled in two ways [34]:

(i) by using an explanatory model, given by Eq. (1):

$$x = g(f_1, \ldots, f_n, error) \tag{1}$$

where, $g()$ is the forecasting function, $f_1, \ldots, f_n$ are the relevant features (parameters, variables) that influence x, and *error* is a random variation due to the effects of relevant features missing from the model.

(ii) by using a time series forecasting equation as given by Eq. (2), in case it is used only the time series of the forecasted parameter, and as given by Eq. (3), in case are used apart from the time series of the forecasted parameter, other relevant features that influence it:

$$x_{t+k} = g(x_t, x_{t-1}, \ldots, x_{t-r}, error) \tag{2}$$

$$x_{t+k} = g(x_t, x_{t-1}, \ldots, x_{t-r}, f_1, \ldots, f_n, error) \tag{3}$$

where, $g()$ is the forecasting function, t is the current time, k is the forecast horizon, r is the past values window and x_t, x_{t-1}, …, x_{t-r} are the current and the past r values of x that compose the time series of x parameter, $f_1, \ldots, f_n$ are the relevant features that influence x, and *error* is a random variation due to relevant features missing from the model.

The specific forecasting problem will consider the forecast horizon k as short-term, medium-term or long-term. Depending on the method chosen to solve the forecasting problem, the forecasting function $g()$ can be a regression function, an artificial neural network etc. In the applications that use time series datasets it is important to detect patterns such as trend, seasonal and cyclic [34].

The major general challenges of the forecasting problems solving (synthesized from [34,35]) are:

(1) *Feature selection* — Selecting the (most) relevant features that influence the predicted parameter (with direct impact on the forecasting accuracy). The efficiency of this selection is dependent on the availability of data corresponding to selected features.

(2) *Efficient dataset analysis* — Performing an efficient dataset analysis (e.g. time series or cross-section data analysis, individual data or aggregated data analysis) is a key challenge. It is essential to detect different types of data (e.g. higher frequency data, peaks, troughs) that need to be isolated from forecast, as well as different patterns (e.g. seasonal, cyclic data, trend, randomness) which become important to the forecasting accuracy, in correlation with the forecasting time horizon. For example, in short-term forecasting, randomness is the most important data pattern, while in medium-term and long-term forecasting, seasonality and trend become more important with an increase of trend and cyclical patterns importance when very long-term time horizon is considered. Also, another challenge is how to deal with chaotic datasets.

(3) *Appropriate forecasting methods selection* — How to choose more (possible, the most) appropriate forecasting methods? What are these methods? We have to point out that the selection of a certain appropriate forecasting method depends on various criteria such as the type of forecast is required, probabilistic forecast or point forecast.

(4) *Forecasting performance improvement* — Improving the forecasting performance in terms of accuracy, computational time etc. Choosing proper metrics for measuring the forecasting accuracy (e.g. scaled errors or percentage errors). Providing the forecasting accuracy and its uncertainty (e.g. in terms of prediction intervals).

(5) *Achieving forecasting objectives* — Fulfilling the objectives for which forecasting is necessary (e.g. optimal planning or real-time forecasting). For example, in real-time forecasting systems it is extremely important to have an accurate forecast as quickly as possible (i.e. in real time).

Each of these challenges has to be addressed when solving a forecasting problem, some of them being crucial when performing comparative studies as is, for example, the selection of more appropriate forecasting methods. A general guideline to this issue is that there are two main classes of methods [34]: statistical (e.g. exponential smoothing, ARMA, ARIMA) and non-statistical (e.g. CI methods such as ANNs, SVM, ANFIS). In this paper we focus on CI methods applications to forecasting problems solving.

2.2.2. Forecasting applications

We have selected for our review four main domains of application: seismology, environmental protection, hydrology and energy, for which real time accurate forecasting is important. Some applications from other domains (e.g. engineering, transport) were added to our review. We have surveyed 70 papers with applications from seismology (earthquake prediction, tsunami prediction), hydrology (floods prediction, rainfall prediction), environmental protection (air pollution, water pollution, soil pollution), energy (energy load prediction), industry (oil and gas, chemistry) etc. Most of the papers were recently published (34 papers in the last three years — 2017–2019, 44 papers in the last 5 years — 2015–2019, and 67 papers in the last 10 years — 2009–2019). The papers were selected by using the following criteria: (1) domain of application; (2) a large geographical distribution

Table 1
A synthesis of selected CI methods parameters.

CI method	Parameters (selection)
FL	Fuzzy terms, Membership function
ANN	ANN architecture (number of layers, number of nodes in each layer), weights, learning algorithm, learning rate, number of epochs, activation function
SVM	Kernel function, soft margin parameter
GA	Population size, crossover probability, mutation probability, number of iterations
GP	Population size, crossover probability, mutation probability, number of iterations
DE	Population size, crossover probability
SA	Initial temperature and the cooling schedule
ACO	Pheromone evaporation rate, weights of arcs, displacement probability
PSO	Best position of each particle, best position in the neighborhood, learning factors

including research studies from worldwide, especially from regions that are confronted with the application domain major problems (e.g. active seismic regions, floods, urban air pollution); (3) research work applying a variety of CI algorithms under single, hybrid or ensemble methods, including also benchmarking, (4) more recently reported (current year, last two–three years), (5) novelty degree (new or recently reported improved methods with better forecasting performance) and significant results, including real time response, (6) paper quality in terms of journal impact and/or paper indexing in important databases (e.g. Web of Science, Scopus, IEEE Xplore, Google Scholar), as well as the citations number of the paper (some highly cited papers were also included). A tradeoff between these criteria was made when selecting papers for the application domain-oriented survey.

2.2.2.1. Forecasting performance evaluation. The performance of the CI methods were evaluated in the surveyed papers by different statistical metrics such as:

(i) *different types of errors*: mean square error (MSE), normalized mean square error (NMSE), root mean square error (RMSE), normalized RMSE (NRMSE), root mean square percentage error (RMSPE), cumulative variation RMSE (CV-RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), symmetric MAPE (sMAPE), mean absolute relative error (MARE), mean bias error (MBE), sum of squared error (SSE), mean bias (MB), mean normalized gross error (MNGE), mean normalized bias (MNB) etc.;

(ii) *correlations coefficients, tests, other metrics*: correlation coefficient (R), efficiency index (EI), Nash–Sutcliffe efficiency coefficient (NS), the coefficient of determination (R^2), the area under the curve of success (AUC), the Friedman and Wilcoxon signed-rank tests, classification accuracy rate (CAR), maximum absolute difference (MAX), Matthews Correlation Coefficient (MCC), McNemar's test accuracy, Theil's U, index of agreement (IA), Average Relative Variance (ARV) etc.;

(iii) *metrics for binary classification*: true/false positive rate (TP/FP), false/true negative rate (FN/TN), sensitivity (S_n), specificity (S_p), precision, recall, CAR etc.

We present a synthesis of the performed survey for each domain of application that was analyzed. We have to note that when we had no access to detailed information related to benchmarking we have marked it in synthesis tables as no benchmarking (i.e. symbol -).

2.2.2.2. Seismology. The selected papers [36–55] report the following types of applications of the CI methods in the seismology domain: earthquake occurrence prediction [36–46], tsunami characteristics prediction [47,48], prediction of specific earthquake engineering parameters (e.g. time-domain parameters: peak ground acceleration — PGA with its components — horizontal and vertical, peak ground velocity — PGV, and peak ground displacement — PGD) [49–51], prediction/estimation of the effects of earthquakes and tsunami to the structure of buildings and bridges [52–55].

(1) *Earthquake prediction*

Most of the earthquake prediction methods reported in the surveyed papers ([36–46]) are based on ANNs ([36–43]) which proved to give better forecasting results when benchmarking was performed. The other types of prediction methods that were used are: regression algorithms ([44,45]), and a GP-based method [46]. The benchmarking procedure that was followed by the papers that reported comparative studies was based on the selection of some forecasting methods (usually, the basic forecasting method that was improved, and some other methods such as ML methods or hybrid methods) that were compared with the proposed method. Only two papers included in the benchmark a domain specific method (i.e. GMDH in [37] and a region-specific method in [39]). The common performance measures that were used in most comparative studies are: FP, MCC, S_n, S_p, and R. Other metrics that were used by few papers are RMSE and McNemar's test accuracy. We have to point out that some CI algorithms were applied for the improvement of the forecasting methods performance (i.e. they were used as optimization methods) as for example, the Guided ABC algorithm, applied to MLP training [38] and enhanced PSO weight optimization of ANN models [39]. We highlight two important achievements for the improvement of earthquake prediction systems performance: the use of cloud-based big data infrastructures with regression algorithms for earthquake prediction [44] and the approach described in [45], based on quantitative association rules and regression techniques, that was applied to discover patterns which model the behavior of seismic temporal data in order to improve the earthquake prediction accuracy. Two ensemble methods were experimented with success for earthquake prediction: the ensemble of CI algorithms (ANNs, RF, SVM) [42] and GP-AdaBoost [46].

(2) *Tsunami prediction*

In the surveyed papers, two types of CI methods were applied for tsunami prediction: a GA-based method, namely least squares (LS) inversion method improved with GA and pattern search [47] and an ANN-based method [48]. We highlight that GA was applied in [47] for prediction method optimization. One of the papers [48] reported a comparative study between the proposed method, Generalized Regression Neural Network (GRNN) and the standard ANN.

(3) *Prediction/estimation of principal ground motion engineering parameters*

ANN-based methods were used with success for the prediction of principal ground motion parameters in [49,50] and [51]. The benchmarking procedure that was followed in all three papers is similar to the one described for earthquake prediction. The common performance measures are MSE, RMSE and MAE. Additional metrics were used as e.g. R in [49] and MAPE in [51]. Two papers, [49] and [50], included in the benchmarking experimental setup domain specific methods (i.e. attenuation models). We have to highlight that a CI algorithm, namely, SA, was used as an optimization method, for ANN training in order to avoid becoming trapped in local minima. Also, a robust forecasting model was developed for estimating the time-domain strong ground motion parameters (PGA, PGV, and PGD) with deep neural networks (DNN) [51]. The main advantage of this solution is no feature selection step needed due to the deep architecture of the ANN. Still, such prediction models require large databases for training, which are not always available.

(4) *Prediction/estimation of the earthquakes and tsunami effects to the structure of buildings and bridges*

Some fuzzy-based forecasting models were experimented with success for the prediction/estimation of the earthquake/tsunami effects to the structure of building and bridges as shown in [52, 53] and [54]. Also, a multi-gene GP method was reported in [55] for solving similar problems. The benchmarking procedure that is described for earthquake prediction was followed in this case, too. The common performance measures are MSE and RMSE. The common winners of the comparative studies are fuzzy-based hybrid methods, FWRBF [53] and ANFIS [54]. We have to specify that the wavelet transform was used in [53] as an optimization method in order to improve the performance of the forecasting method.

Table 2 presents a synthesis of the surveyed forecasting applications in the domain of seismology solved with CI methods.

2.2.2.3. Environmental protection. Two main types of prediction applications for the environmental protection domain were identified in the selected papers: environment pollutant concentration forecast [56–67], and prediction of the environment quality [68–71].

(1) *Environment pollutants concentration forecasting*

The selected papers report research work on air, water and soil pollutants concentration forecasting with a variety of CI-based methods, including ensemble and hybrid methods.

The majority of the papers used an ANN-based forecasting model either as a standalone method [57,60,61,63,64,66,67] or integrated in an ensemble or hybrid method [56,58,59]. Certain papers applied a fuzzy logic based model ([62,63,65,66]), some of them including the model in their benchmarking experiments, and one paper [65] tackling uncertainty management under a holistic approach. Also, other CI methods were applied for optimization tasks such as optimal features set selection or ANN weights tuning and training. Examples of such methods are GA [56,57], RF [57], PSO [64], and modified cuckoo search algorithm (MCSA) [61]. Some of the analyzed papers followed a simple benchmarking procedure that compares the improved method that is proposed with the original method (e.g. [57, 61,64]). Other papers extend the comparison to one or more CI-based algorithms (e.g. [56,63,66,67]). Only one paper [60] performed a comparison between the proposed method (Deep LSTM), a naïve method (ground PM observations), and simulations of a domain specific method, 3D-chemistry-transport model (CTM). The common best forecasting models (in terms of forecasting accuracy) are various ANN models improved by different CI-based optimization methods. The performance measures that were used in most papers are RMSE, R, IA, MAE and MSE. Examples of other metrics that were applied rarely are: Theil's U, NS, MARE, MAPE, R^2, ARV, MAX, TP and FP.

(2) *Environment quality forecasting*

The surveyed papers include air quality index (AQI) real time forecasting for different air pollutants [68–70], and water quality index (WQI) forecasting for a river [71].

All selected papers applied an ANN-based forecasting method with good forecasting accuracy, especially in AQI real time forecasting systems ([68–70]). We highlight the use of a GP method coupled with spectral analysis for seasonal multicity AQI real time prediction in [69] for short-term forecasting horizon and of a GA method for optimal features set selection [70]. The common benchmarking procedure involved a comparison between the proposed improved CI forecasting method and similar CI-based methods, either individual models (e.g. SVR) or hybrid ones (e.g. TSFIS). A common forecasting performance measure that was used is RMSE.

Table 3 presents a synthesis of the surveyed forecasting applications in the domain of environmental pollution (air, water and soil) solved with CI methods.

2.2.2.4. Hydrology. One of the natural disaster that can have multiple effects on humans, ecosystems, agriculture and farming, economy is flood. Early warning systems based on accurate flood prediction can reduce these effects. A variety of CI methods (single, hybrid or ensemble) were proposed for hydrological modeling and flood forecasting, some of them being integrated in real time early warning systems. We have surveyed papers that report the following types of applications: flood forecasting [72–80], peak flood forecasting [81,82], river flow forecasting [83], rainfall prediction [84,85], extreme rainfall prediction [86], spatial prediction of floods [87,88], tropical cyclones forecasting [89,90], and groundwater resources sustainable management [91].

(1) *Flood forecasting*

The most used CI methods applied to flood forecasting in the surveyed papers are: ANN and SVM. Different types of ANNs were applied either as single models, as for example, FFANN ([76,77]), WNN [75] or as hybrid method as for example, ANFIS ([75,78]) and Wavelets Neuro-Fuzzy (WNF) [75]. Also, the SVM model was applied as an individual model ([78,80]) or as an extended model, LSSVM [73]. The majority of the analyzed papers ([72–78]) performed benchmarking in general, by comparing the proposed improved method(s) with individual or hybrid CI methods. Some of them performed the comparison only between the proposed methods (e.g. [73,75,78]). Only one paper reported a comparison with domain dependent models [76] and another paper [74] performed a comparison with a pure statistical method, ARIMA. We highlight the application of some CI methods for the prediction model optimization, as for example, GA, DE, Grey Wolf optimizer (GWO), and cuckoo search (CS), all these methods being applied to LSSVM prediction model improvement as described in [73]. The common performance measures that were used are RMSE and R^2. Other examples of metrics that were applied for prediction accuracy estimation are: MAPE, R, RMSPE, MSE, Theil's U, NS.

(2) *Peak flood forecasting*

Two applications that performed peak flood forecasting were selected in our survey, both using with success EC-based methods, namely GP-based [81] and GA-based [82], the last one being integrated in a hybrid forecasting model. The benchmarking procedure that was followed in both papers involved the comparison of the proposed methods with one or two other methods, namely with a regression method in [81] and with other CI-based hybrid methods in [82]: ANFIS and a neuro-genetic model (NGO). We highlight the use of the GA method for fuzzy rule selection in [82] which improved the accuracy of flood peak forecast. The performance measures that were used are RMSE, R^2, EI and Peak flood error.

(3) *River flow forecast*

One paper [83] was selected for short-term river flow forecast. The benchmarking procedure was a simple one, performing a comparison between the improved forecasting method (SVM improved by PSO) and the original method (SVM). The PSO metaheuristic was applied as an optimization method for the detection of SVM best parameters.

(4) *Rainfall prediction*

Two rainfall prediction systems reported in recently published papers, [84] in 2019, and [85] in 2017, were selected for our

Table 2
Synthesis of the surveyed CI applications in the seismology domain for prediction problems.

Ref.	Problem/Region	CI Method(s)	Benchmarking	Metrics	Main Conclusion/Major Benefit
[36]	Earthquake prediction system/Chile	Prediction method: ANN	ANN, other ML techniques	Minimum FP	Best results of ANN.
[37]	Time series prediction of earthquake/Japan	Prediction method: hybrid CI method (FL, ANN)	Chaos Theory, ANN, GMDH, and FL modeling for nonlinear time series	–	Best results of hybrid prediction methods. Use of chaos theory for chaotic time series.
[38]	Earthquake time series prediction	Prediction method: optimized MLP Optimization method: ABC	MLP with different learning algorithms (BP, GGABC, ABC)	RMSE	Best results of MLP trained with GGABC.
[39]	Earthquake prediction system/Hindukush, Chile, California	Prediction method: SVR-HNN	SVR, HNN, SVR-HNN, and a region specific method tested before	S_n, S_p, R, MCC, positive and negative predictive values	Better results for earthquake prediction achieved by SVR-HNN model.
[40]	Prediction of a medium–large earthquake magnitude/Japan	Prediction method: ANN	–	–	Good prediction accuracy of the ANN model.
[41]	Earthquake magnitude prediction	Prediction method: Probabilistic ANN	–	–	Good prediction results.
[42]	Earthquake prediction system/northern Pakistan	Prediction method: CI Ensemble (FFANN, RNN, RF, MLP, RBFNN, SVM)	FFANN, RNN, RF, MLP, RBFNN, SVM	MCC, TP, FP, TN, FN, S_n, S_p, R, accuracy, McNemar's test accuracy	The best results were given by FFANN.
[43]	Earthquake prediction/ Iberian Peninsula	Prediction method: ANNs	ANNs and other classifiers	–	Better prediction of ANNs.
[44]	Earthquake prediction/California	Prediction method: Regression algorithms	–	–	Good prediction results. Use of cloud-based big data infrastructures.
[45]	Earthquake prediction/Spain	Prediction method: Regression with association rules	–	–	Best accuracy of the prediction model improved with quantitative association rules.
[46]	Earthquake prediction/Hindukush, Chile, California	Prediction method: GP-AdaBoost	–	MCC, S_n, S_p, accuracy, R	Better prediction accuracy.
[47]	Estimation of initial tsunami source/Japan	Prediction method: GA-improved LS inversion Optimization method: GA	GA-based least squares inversion method, conventional method	RMSE	The proposed method is independent of the earthquake parameters.
[48]	Prediction of tsunami characteristics/Indonesia	Prediction method: GRNN	GRNN, ANN	–	Best results of GRNN. The need to develop a tsunami database.
[49]	PGA prediction/Iran	Prediction method: ANN improved with SA (SA-ANN)	SA-ANN, ANN, 10 other attenuation models	MAE, MSE, RMSE, R, r^2	Best results of SA-ANN. PGA prediction model — an explicit formula for tectonic regions of Iran.
[50]	PGA estimation/Mexico	Prediction method: ANN	ANN, a better-fitted regression approach, attenuation models	MSE	Better results of ANN.
[51]	Prediction of strong ground motion parameters (PGA, PGV, PGD)	Prediction method: DNN	DNN, ANN/SA, GP/OLS, MEP, GP/SA, RANFIS	MAE, MAPE, RMSE	DNN provides robust models for estimating PGA, PGV, and PGD by using large seismic databases from different regions.
[52]	Building assessment in earthquake prone areas/Turkey	Prediction method: FL	–	–	An early warning tool in case of earthquake occurrence.
[53]	Prediction of the structural responses to natural ground motion	Prediction method: FWRBF	FWRBF, ANFIS, RBFNN.	MSE, RMSE	Better prediction accuracy provided by FWRBF.
[54]	The prediction of the effects of perforations in a girder of a bridge due to the waves created by tsunami/Japan	Prediction methods: SVR, ANFIS, RBFNN	SVR (different kernel functions), ANFIS	–	Best results were given by ANFIS.
[55]	Analysis of geotechnical and earthquake engineering systems	Prediction method: MGGP (multi-gene GP)	ANNs and other soft computing tools.	–	The MGGP-based solutions are valuable for pre-design.

Table 3
Synthesis of the surveyed CI applications in the environmental pollution domain for prediction problems.

Ref.	Problem/Region	CI Method(s)	Benchmarking	Metrics	Main Conclusion/Major Benefit
[56]	Air pollution forecast next day PM_{10}/Warsaw, Poland	Prediction methods: RF, ANNs (MLP, RBF, SVM) Features selection: GA and stepwise fit	RF, MLP, RBF, SVM, RF, linear ARX	MAE, MAPE, MAX, RMSE, R	Best results given by RF.
[57]	Haze management/Brunei, Turkey	Prediction method: BPNN Feature selection: RF, GA	Improved BPNN (with RF-GA feature selection), BPNN	TP, FP, false alarm rate, success index	Better results of improved BPNN with RF-GA feature selection.
[58]	Air pollutant concentration forecast - Ozone/Athens, Greece	Prediction approach: HISYCOL (ensemble learning FIS, FFANN, RF, Mamdani FIS)	–	–	Better forecasting results of HISYCOL.
[59]	Air pollution forecast/Warsaw, Poland	Prediction method: Ensemble method MLP, Elman network, SVM, RF	–	–	Forecast increased accuracy with the ensemble method.
[60]	PM_{10} and $PM_{2.5}$ daily forecast/South Korea	Prediction method: Deep LSTM-based model	Deep LSTM, ground PM observations, 3D-CTM model	IA, RMSE, MB, MNGE, MNB	Better performance of Deep LSTM in terms of IA.
[61]	Prediction of chaotic time series atmospheric pollutant data/Texas, USA	Prediction method: WNN improved with MCSA model Optimization method: MCSA	WNN-MCSA, WNN-random	MSE, RMSE, NMSE	Superior prediction accuracy of the WNN-MCSA model.
[62]	Primary and secondary air pollutants forecasting/Athens, Greece	Prediction method: FL (fuzzy cognitive maps)	–	–	Better results for the forecasting accuracy.
[63]	$PM_{2.5}$ air pollutant forecasting/Ploiesti, Romania	Prediction methods: ANNs, ANFIS	ANNs, ANFIS	MAE, RMSE, IA, R	Better performance of ANN model.
[64]	Air pollutant concentration level forecasting	Prediction method: improved ANN (with PSO weights tuning and training) Optimization method: PSO	Best ANN-PSO, ANN	MSE, Theil's U, ARV, MAPE, IA	Best results of the hybrid prediction model (ANN-PSO).
[65]	Air pollution forecast/Ploiesti, Romania	Prediction approach: Holistic approach (FL, ML – DM, BN)	–	–	Better results for combined methods with uncertainty management.
[66]	Estimation of soil dispersivity (alpha)	Prediction methods: ANFIS, ANN, GEP	ANFIS, ANN, GEP, MLR	RMSE, R^2	Best results given by ANN.
[67]	Surface water quality prediction (DO)/Delaware River, USA	Prediction methods: ANN (MLP, RBF), linear GP (LGP), SVM	ANN (MLP, RBF), linear GP (LGP), SVM	RMSE, NS, MARE, R	Best results were given by SVM.
[68]	Urban air quality real-time prediction	Prediction methods: FL and LSSVM	–	–	Good prediction accuracy for simulations
[69]	Short term AQI forecasting/3 cities in India	Prediction methods: ANN, GP-spectral analysis	–	–	Good results for real time prediction of seasonal multicity AQI with GP.
[70]	Prediction of AQIs for more air pollutants /Czech Republic	Prediction methods: TSFIS, RBFNN, MLP, SVR Feature selection: GA	(TSFIS, RBFNN, MLP, SVR) with GA feature selection, RBFNN, MLP, SVR	RMSE	The compositions of individual prediction models significantly outperform single prediction models of commonly AQI.
[71]	River water quality index (WQI) prediction/Malaysia	Prediction method: ANFIS with fuzzy c-means data clustering	–	–	Good results of the proposed WQI prediction model.

survey. The first one was applied in Brazil [84], while the second one was applied in Japan [85]. Both papers reported the use of ANN forecasting models: a binary MLP [84], and two ANN models: MLP and RBFNN [85]. We highlight the inclusion of a domain specific forecasting method, the Japan Meteorological Agency (JMA) prediction method in the benchmarking described in [85], and the use of some domain specific performance measures: total hit rate (THR), hit rate of precipitation (HRP), hit rate of non-precipitation (HRN), caching rate (CR), overlooking rate (OR), and swing-and miss rate (SMR). The main conclusion of the comparison revealed that the JMA model (i.e. a non CI forecasting model) provided the most accurate rainfall prediction while MLP performed better than RBFNN. Two methods were applied for the ANN-based forecasting models optimization: random optimization (RO) for MLP and least square method (LSM) for RBFNN.

(5) *Extreme rainfall forecasting*
An ensemble extreme rainfall forecasting model based on soft computing is proposed in [86]. The benchmarking procedure that was followed involved the inclusion of a domain specific forecasting model (a global climate model). The experimental results revealed a more accurate prediction of the ensemble model. The performance measures that were used are RMSE, MAPE, MASE and R.

(6) *Spatial prediction of floods*
The spatial prediction of floods with two hybrid CI models is tackled in the selected papers, [87] and [88]. Both models

are using some optimization techniques, as for example, firefly algorithm (FA) (for ANFIS model improvement in [87], for ANN training, in combination with the Levenberg–Marquardt learning algorithm in [88]), and imperialistic competitive algorithm (ICA) (for ANFIS model improvement in [87]). The common benchmarking procedure involved the comparison between the proposed hybrid forecasting methods (ANFIS-ICA, ANFIS-FA in [87], and ANN-FA-LM in [88]) and other CI methods (the best ML techniques tested for the area of study in [87], and SVM and classification tree (CT) in [88]). Different metrics were used for the forecasting performance measure: RMSE, MSE, AUC, Friedman, Wilcoxon signal rank tests in [87], and TP/FP rate, FN/TN rate, precision, recall and CAR in [88].

(7) *Tropical cyclones forecasting*

Two tropical cyclones forecasting systems based on ANN were selected for our survey. Both systems were tested with a similar benchmarking procedure that comprised a comparison between the proposed method and other methods (e.g. GP in [89]). We highlight the use in the benchmarking of some traditional statistical models such as multiple linear regression (MLR) and of a climatology average model in [90]. The performance measures were R and RMSE. The experimental results revealed that the best forecasting methods were ANN and decision tree (DT) methods (e.g. CART, Chi-squared, CHAID).

(8) *Sustainable management of groundwater resources*

We have selected for our survey a recent application of the ANFIS method in the domain of sustainable management of groundwater resources [91]. Five metaheuristic algorithms were used for the optimization of the ANFIS parameters: DE, invasive weed optimization (IWO), FA, PSO, and bees algorithm (BA), providing five forecasting models for the spatial prediction of groundwater springs (ANFIS-DE, ANFIS-IWO, ANFIS-FA, ANFIS-PSO, and ANFIS-BA). The benchmarking procedure involved a comparison between the performances of these models in terms of AUC, and Friedman, Wilcoxon signed-rank tests. The experiments showed that all models provided high prediction performance and the most accurate forecast was given by the ANFIS-DE model.

Table 4 presents a synthesis of the forecasting applications in the domain of hydrology (flood, rainfall prediction) solved with CI methods.

2.2.2.5. Energy. Several applications of CI methods for solving prediction/forecasting problems in the energy domain were reported in the literature. In this survey we have included three main types of applications: energy load forecasting [92–94], solar power forecasting [95–97], and wind energy output prediction [98–100].

(1) *Energy load forecasting*

Three applications of energy load forecasting were surveyed. Two of them applied an ANN prediction model ([92,93]) for energy load forecasting, while the third one applied a hybrid PSO-based method for loss power minimization [94]. The PSO method was used as an optimization method, usually for ANN training as in [93]. The benchmarking procedure that was followed involved the comparison between the proposed forecasting method (e.g. DeepEnergy, a deep learning NN combined with a convolutional NN in [92] and BPNN-PSOCF, an ANN model optimized with PSO in [93]) and other CI methods (e.g. SVM, RF, DT, LSTM – long short term memory network [101] – for the benchmarking described in [92], and ANN, SVM for the benchmarking described in [93]). The common performance measure that was used is MAPE. Other metrics such as CV-RMSE [92] and MAE [93] were also computed. The experimental results showed that the proposed models provided the most accurate forecast in terms of MAPE and other specific metrics.

(2) *Solar power forecasting*

Three applications for solar power forecasting were selected for our survey, one revealing the current trend of using deep learning for improved forecasting accuracy [95], one based on a FIS method [96], and the last one based on an ANN ensemble method [97]. A simple benchmarking procedure was followed involving the comparison of the forecasting method with another method (as e.g. MLR in [96]). Examples of metrics that were computed for the evaluation of the forecasting accuracy are RMSE, R^2 and SSE.

(3) *Wind energy output prediction*

Three applications of wind energy power prediction were selected for our survey. Each application proposes a different forecasting CI method: symbolic regression based on GP (in [98]), SVR (in [99]), and a novel combined forecasting model (NCFM) (in [100]) based on a combination between three forecasting methods, ARIMA, BPNN, and ENN which provided the input data to the ELM model (Extreme Learning Machine) that was developed for a single layer FFANN. The benchmarking procedure comprised the comparison of the proposed model with the individual forecasting methods (e.g. ARIMA, BPNN and Elmann NN — ENN as in [100]). We highlight the inclusion of a statistical forecasting method, ARIMA, in the benchmarking. The common performance measures are MAE, RMSE and MAPE. Another metric that was computed is SSE.

Table 5 presents a synthesis of the forecasting applications in the domain of energy solved with CI methods.

2.2.2.6. Other applications. Five engineering applications were selected for this last part of the survey: forecasting the air-overpressure induced by mine blasting [102], crystal stability prediction [103], crystal structure prediction [104], traffic flow forecasting [105], and hydrocarbon production from shales forecasting [106]. The CI-based forecasting models that were applied with success in the majority of these applications are a type of improved ANN model (e.g. DNN in [103]) and variants of SVM (e.g. SVR in [105], LSSVM in [106]) that provided the most accurate forecast in terms of specific metrics. The PSO method was applied for the optimization of the prediction models. For example, PSO was used in [102] for the optimization of the coefficients of three prediction equations — linear, power and quadratic, and in [105] for the optimization of the SVR forecasting method. The benchmarking procedure that was followed in the analyzed papers involved the comparison of the proposed method(s) with other forecasting methods, either individual CI methods (as e.g. ANNs in [103]) or with some basic prediction methods (as e.g. the traditional curve fitting method – Response Surface Model – RSM in [106]) and domain specific forecasting methods (as e.g. USBM in [102]). The common performance measures that were used are MSE and R^2. Other metrics specific to the application domain were also computed (e.g. low energy distribution of structure in [104]).

Table 6 presents a synthesis of the forecasting applications in other domains solved with CI methods.

At the end of our survey (that was application domains oriented), we present a synthesis of the main conclusions.

2.2.3. Survey conclusions

The common successful prediction methods reported in the surveyed papers are ANN-based models applied either as individual methods (e.g. MLP, FFANN, DNN) or in combination with other methods (e.g. ANFIS, WNN, SVR-HNN), while the performance measures that were commonly used in the majority of the papers are RMSE, R, MAPE, MSE and R^2. Few papers have added some

Table 4
Synthesis of the surveyed CI applications in the hydrology domain for prediction problems.

Ref.	Problem/Region	CI Method(s)	Benchmarking	Metrics	Main Conclusion/Major Benefit
[72]	Flood forecasting in natural rivers/the Neckar River, Germany	Prediction methods: OS-ELM based on ML	SVM, ANN, GP, OS-ELM	R^2, NS, RMSE	Best results were given by the OS-ELM technique.
[73]	Flood forecasting with a week ahead	Prediction hybrid methods: GA-LSSVM, DE-LSSVM, GWO-LSSVM, CS-LSSVM Optimization methods: GA, DE, GWO, CS	GA-LSSVM, DE-LSSVM, GWO-LSSVM, CS-LSSVM	MSE, RMSPE, Theil's U	All algorithms gave better results when using normalized time series data.
[74]	Heavy precipitations prediction on monthly basis	Prediction method: ANN	ANN, ARIMA	RMSE, R^2	ANN gave better results, being a reliable method for heavy precipitations prediction.
[75]	Real-time flood forecasting/Richmond River, Australia	Prediction method: ANN, ANFIS, WNN, hybrid ANFIS-WNF	ANN, ANFIS, WNN, hybrid ANFIS-WNF	–	WNN and ANFIS-WNF gave better results in the longer lead-time forecasting
[76]	Three day ahead flood-forecast/coastal catchments in British Columbia, Canada.	Prediction methods: ANNs	ANNs, other domain dependent models	–	Best results were obtained with the ANN-based flood-forecast method.
[77]	Real-time flood alerting intelligent system/Prahova River, Romania	Prediction method: ANN (FFANN)	different ANN architectures	RMSE	Good results for the simulation experiments.
[78]	Monthly discharge time series forecasting	Prediction methods: ANN, ANFIS, GP, SVM	ANN, ANFIS, GP, SVM	R, NS, RMSE, MAPE	Best results given by ANFIS, GP and SVM.
[79]	Flood susceptibility assessment	Prediction method: bagging-logistic model tree (LMT)	–	–	Good prediction results.
[80]	Flood disaster loss comprehensive evaluation	Prediction method: SVM	–	–	Good results provided by SVM.
[81]	Reliable forecasting of the peak flood discharge at river basins	Prediction methods: GEP, LGP	GEP, LGP, logistic regression	R^2	GEP method provided the highest correlation.
[82]	Flood evaluation/two floods in Japan	Prediction method: ANFIS (FL, ANN-GA) Optimization method: GA	ANFIS (FL, ANN-GA), ANFIS, NGO	RMSE, Peak flood error, EI	The benchmarking showed that the use of fuzzy rule base, selected by GA in the hybrid multi-model can improve the accuracy of flood peak forecast.
[83]	Short-term forecast of daily river flow/Malaysia	Prediction methods: SVM, SVM-PSO Optimization method: PSO	SVM, SVM-PSO	–	Better results of SVM-PSO.
[84]	Rainfall prediction/Brazil	Prediction method: Binary MLP	–	–	Good prediction results.
[85]	Local heavy rainfall prediction/Japan	Prediction methods: ANNs (MLP, RBFNN)	MLP, RBFNN, JMA	THR, HRP, HRN, CR, OR, SMR	JMA model gave the best results and MLP better than RBFNN.
[86]	Forecast extreme rainfalls/Naples, Italy	Prediction method: Soft computing — Ensemble forecast	Ensemble forecast, Global climate model	RMSE, MAPE, MASE, R	Good prediction results of the ensemble model
[87]	Spatial prediction of floods/Haraz watershed, Iran	Prediction hybrid methods: ANFIS-ICA, ANFIS-FA Optimization methods: ICA, FA	ANFIS-ICA, ANFIS-FA, best ML techniques tested for the area of study	RMSE, MSE, Friedman, Wilcoxon signed-rank tests, AUC	A better performance was shown by the ANFIS-ICA model.
[88]	Spatial prediction of flash floods/Vietnam	Prediction hybrid method: ANN-FA-LM Optimization method: FA	ANN-FA-LM, ANN-LM, ANN-FA, SVM, CT	TP/FP rate, FN/TN rate, precision, recall, CAR	SVM and CT were successfully employed in flood susceptibility assessment. The best results were given by ANN-FA-LM.
[89]	Real time tropical cyclones forecasting/East coast of India	Prediction method: ANN	ANN, GA, GP	R	The ANN model gave the best results.

(*continued on next page*)

domain specific metrics (as e.g. IA, peak flood error, THR, HRP, HRN). Some papers included statistical tests, such as the Friedman and Wilcoxon signed-rank tests. The evolutionary algorithm that was mostly used as prediction method is GP. We highlight that most of the successful prediction methods were optimized with different CI methods such as GA, SA, PSO, ABC, BA, WT, DE, FA.

Table 4 (*continued*).

Ref.	Problem/Region	CI Method(s)	Benchmarking	Metrics	Main Conclusion/Major Benefit
[90]	Prediction of hourly precipitations forecasting/Shihmen Reservoir Watershed, Taiwan	Prediction methods: ANN (MLP), DT — (CART, Chi-squared, CHAID)	ANN (MLP), DT — (CART, Chi-squared, CHAID etc.), MLR, climatology average model	RMSE	The ANN and DT predictive models gave better results in comparison with traditional models.
[91]	Sustainable management of groundwater resources/Iran	Prediction methods: ANFIS-DE, ANFIS-IWO, ANFIS-FA, ANFIS-PSO, ANFIS-BA Optimization methods: DE, IWO, FA, PSO, BA	ANFIS-DE, ANFIS-IWO, ANFIS-FA, ANFIS-PSO, ANFIS-BA	AUC, Friedman, Wilcoxon signed-rank tests	All models have high prediction performance. Best prediction given by ANFIS-DE.

Table 5
Synthesis of the selected CI applications in the energy domain for prediction problems.

Ref.	Problem/Region	CI Method(s)	Benchmarking	Metrics	Main Conclusion/Major Benefit
[92]	Short-term energy load forecasting/USA	Prediction method: DeepEnergy (CNN-ANN)	DeepEnergy, SVM (RBF), RF, DT, LSTM	MAPE, CV-RMSE	High precision of DeepEnergy model.
[93]	Middle-power load forecasting/North China	Prediction method: BPNN-PSOCF Optimization method: PSO	BPNN-PSOCF, ANN, SVM	MAE, MAPE, relative degree (r), the fitness error	Best results provided by BPNN-PSOCF.
[94]	Loss power minimization	Prediction method: Hybrid PSO	–	–	Good performance of the hybrid PSO method.
[95]	Solar power forecasting on big data time series	Prediction method: DL (Deep Learning)	–	–	DL can provide accurate prediction when big time series datasets are used.
[96]	Solar collector efficiency prediction/Nicosia, North Cyprus	Prediction method: FIS	FIS, MLR	RMSE, R^2, average forecasting error, SSE	FIS provided higher accuracy due to the prediction knowledge it uses.
[97]	Short-term solar power forecasting	Prediction method: ANN ensemble method	–	–	Good forecasting accuracy provided by the ANN ensemble method.
[98]	Wind farms energy output prediction/Australia	Prediction method: Symbolic regression based on GP	–	1-R^2	Very reliable prediction of the energy output for newly supplied weather data.
[99]	Short-term wind energy forecasting	Prediction method: SVR	–	–	Good forecasting results of SVR.
[100]	Short-term wind speed forecasting/Shandong, China	Prediction method: NCFM model (ARIMA, BPNN, ENN, ELM)	NCFM, ARIMA, BPNN, ENN	MAE, RMSE, MAPE, SSE	Best results given by the NCFM model.

Table 6
Synthesis of selected surveyed CI applications in other domains for prediction problems.

Ref.	Problem/Region	CI Method(s)	Benchmarking	Metrics	Main Conclusion/Major Benefit
[102]	Forecasting the air-overpressure induced by mine blasting /Iran	Prediction models: PSO-linear model, PSO-power model, PSO-quadratic model Optimization method: PSO	ANN, USBM, PSO-linear model, PSO-power model, PSO-quadratic model	MSE, R^2	Best results were obtained by PSO-linear model.
[103]	Crystal stability prediction	Prediction method: Deep neural networks (DNN)	ANNs, DNN	MAE	Best results given by DNN.
[104]	Crystal structure prediction/Russia	Prediction method: improved USPEX (EA, DM)	–	Low energy distribution of structure	Efficient method, able to create a diverse set of low energy crystal structures.
[105]	Traffic flow forecasting	Prediction method: SVR-PSO Optimization method: PSO	SVR-PSO, MLR, BPNN	–	SVR-PSO gave better results.
[106]	Forecasting of hydrocarbon production from shales	Prediction methods: LSSVM, ANN, RSM	LSSVM, ANN, RSM	R^2, NRMSE	RSM and LSSVM provided very accurate oil recovery forecasting.

For example, GA, SA, PSO and ABC were used for ANN training. Also, some CI methods were applied for feature selection, as for example, GA, SA, RF, improving the forecasting accuracy. Some of the knowledge accumulated in successful forecasting applications in the surveyed application domains can be transferred to other domains, as e.g. more efficient ANN training algorithms, feature selection methods or forecasting methods hybridization procedures. A major critic that can be made is that the majority of the surveyed papers do not treat explicitly or in sufficient detail the benchmarking procedure that was followed. Few papers performed a detailed analysis of the algorithms performance (e.g. in terms of nonparametric statistical tests).

The common practice of benchmarking identified in the selected application domains is a relatively simple one, involving the comparison between the proposed improved prediction method, the original method and in some cases, other CI methods, statistical or problem domain specific methods, in terms of selected performance measures that provide a measure of the forecasting accuracy. The majority of the benchmarking procedures followed in the surveyed papers involved comparisons between CI methods. Few of them included statistical or domain dependent methods. Such examples are the research work reported in [74] (flood forecasting — comparison with a statistical method, ARIMA), [90] (flood forecasting — comparison with a climatology model), [49] (earthquake prediction — comparison with 10 domain specific methods, corresponding to attenuation models), [39] (earthquake prediction — comparison with a region specific method), [102] (forecasting the air-overpressure induced by mine blasting — comparison with a domain specific method), and [106] (oil recovery forecasting — comparison with a domain specific method). We have analyzed also some recent surveys. The main conclusions are synthesized as follows. Ensemble CI approaches are highly efficient [6] while hybrid methods provide higher accuracy and robustness [7]. The major trends identified in [5] are: novel hybridization, data decomposition, develop ensemble of methods, and add-on optimizer algorithms.

We have analyzed in our survey: hybrid CI methods (55.72%), single CI methods (37.14%), and ensemble CI methods (7.14%). Table 7 presents a synthesis of the CI methods usage as single, hybrid or ensemble in the applications that were surveyed. The task for which the CI method was used (i.e. prediction optimization, feature selection) is specified.

The CI methods that were mostly used in the forecasting applications that were surveyed are: ANN (reported in 43 papers) and FL (reported in 18 papers), usually in hybrid forecasting methods. The main CI algorithms that were applied as optimization algorithms (e.g. applied to algorithm parameters tuning) are: PSO (reported in 9 papers), GA (reported in 7 papers), DE, SA etc. The CI methods that were included in ensemble methods are: ANN, RF, SVM and FL.

A final conclusion of our survey is that all the papers provided some solutions for the majority of the forecasting problems solving challenges enumerated in Section 2.2.1.

3. A general CI benchmarking framework

Based on software engineering principles and the literature review on algorithm selection, we propose a general benchmarking framework (applicable to CI algorithms) to which we have added the use of a case base with solved problems and two knowledge bases. Fig. 3 shows the framework overview. The two knowledge bases (problem domain knowledge base and algorithm specific knowledge base) provide heuristic knowledge that can help in choosing a more proper algorithm, tailored for problem instance solving. The problem domain knowledge base (KB_1) will guide the process of finding more informative problem instance features. The algorithm specific knowledge base (KB_2) contains heuristic rules that can help tuning the parameters of the algorithm.

Therefore, the framework was developed as a software engineering best practice guide extended with some knowledge engineering principles (related mainly to knowledge bases development) and was transposed to CI benchmarking, being a CI engineering methodology. We make the remark that it is not a software implementation or a tool suite. However, it might be implemented as a software tool, an example being given in Section 3.1.11.

In the next sections we present in detail the general benchmarking framework that was formalized in pseudocode, a set of guidelines for CI algorithms selection in forecasting problems solving and some issues related to framework implementation.

3.1. Description of the general benchmarking framework

The main goal of the framework is to find the best CI algorithm that can solve a specific problem. Thus, the framework receives as input a problem instance (P) that need to be solved and provides as output the best CI algorithm ($BestAlg$) that can solve P. The main resources used by the framework are: the two knowledge bases (KB_1 and KB_2), the case base with solved problems (CB_{SP}) and the datasets available for the problem instance P solving (DB_p). We remind that a problem instance P is a particularization of a general problem to some specific datasets (i.e. instance data). A problem instance has associated a type of problem and a problem (the general problem or objective) which are denoted as $ProblemType(P)$ and $GeneralProblem(P)$, respectively.

The general benchmarking framework is formalized in pseudocode where,
P is the problem instance,
F is the set of (more informative) features of the problem instance,
SM is the algorithm selection mechanism (e.g. a ML technique) realizing a mapping between problem instance and algorithms by using the features of the problem instance,
A_p is the set of proper algorithms for solving the problem instance,
N is the number of proper algorithms,
A_i is the ith proper algorithm from the set of proper algorithms, A_p,
SP_{Ai} is the set of algorithm parameters,
TP_{Ai} is the set of tuned (optimized) algorithm parameters,
$Solution_i$ is the solution returned by the algorithm A_i for problem instance solving,
$PerfM_i$ is the performance measure (e.g. a set of statistical and information theoretic measures — runtime, solution quality, accuracy etc.) of A_i algorithm when solving the problem instance for which it returned $Solution_i$,
$Solutions$ is the set of solutions returned by the proper algorithms,
$PerfM$ is the performance measures of all proper algorithms applied to problem instance solving,
$BestAlg$ is the best algorithm for solving the problem instance,
$Solution_{BestAlg}$ is the solution given by the best algorithm for problem instance solving,
Obs represents the lessons learnt (i.e. new knowledge that was derived when solving P) from the new case (e.g. algorithm tuning method, algorithm optimal parameters, algorithm performance, more informative features of the problem instance).

3.1.1. Data preprocessing

Data preprocessing is performed on the available datasets by the ***DataPreprocessing*** (DB_p) function, and can perform various operations such as data cleaning and identification and correction of the errors in raw data (e.g. incorrect or corrupted data), noise elimination by filtering, pre-filtering to detect any anomalies in data (due to measurement deficiencies), remove the zero values, data normalization etc. If datasets from different sources are used then they need to be scaled (as e.g. data on one scale for the time scale). Usually, a statistical data analysis it is performed, computing for each parameter (variable) the maximum, minimum, mean and standard deviation values. In case the datasets contains time series, it is performed time series analysis in order to detect any patterns such as seasonality, trend or cyclic patterns. Also, during this step the available datasets (i.e. instance data) are divided in three datasets, corresponding to the training dataset, the validation dataset and the testing dataset. Depending on the selected algorithm type, the training dataset will be used during algorithm configuration (i.e. step 4.1 and step 4.1'), while the validation dataset and the testing dataset will be used during solution testing step (i.e. step 4.3).

Table 7
Distribution of CI methods types under the forecasting method.

CI method/Task	Forecasting method type		
	Single methods	Hybrid methods	Ensemble methods
ANN/Prediction	18	21	4
FL/Prediction	2	15	1
GA/Optimization, Feature selection	–	7	–
GP/Prediction	3	5	–
PSO/Optimization	1	8	–
SVM/Prediction	3	2	2
SVR/Prediction	3	2	–
LSSVM/Prediction	1	2	–
RF/Prediction, Feature selection	–	2	3
SA/Optimization	–	1	–
WT/Optimization	–	3	–
DE/Optimization	–	2	–
DT/Prediction	1	–	–
FA/Optimization	–	2	–
Deep Learning/Prediction	–	4	–
ABC/Optimization	–	2	–
ACO/-	–	–	–

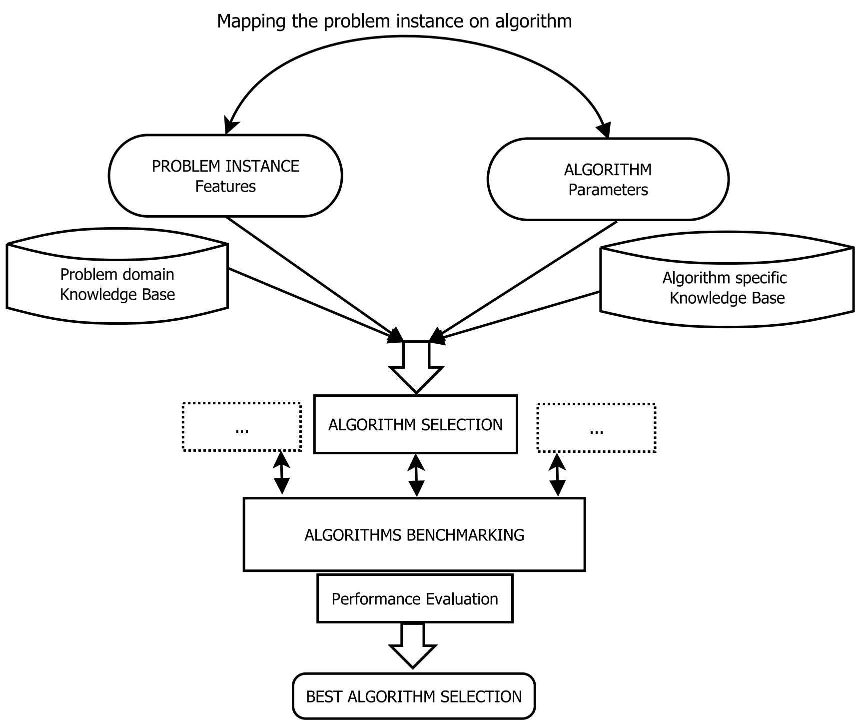

Fig. 3. Framework overview.

3.1.2. Feature selection

A knowledge-based features selection is performed by the ***FeatureSelection*** (P, KB_1) function. The most informative (relevant) features of the problem instance P are selected by using general or domain specific knowledge from KB_1. This type of knowledge (related to feature selection) is provided by human experts or is taken from literature. For example, a general knowledge is given by the main classes of feature selection techniques [107]: (1) variable ranking (by correlation coefficient or single variable classifiers; by using a metric such as classification success rate or hit rate; by an information theoretic ranking criterion such as mutual information estimation for each variable), (2) variable subset selection (by using wrappers as e.g. forward selection and backward elimination, filters, embedded methods), and (3) clustering (i.e. unsupervised learning). The number of features can be reduced by using different techniques such as Principal Component Analysis (PCA). A proper feature selection method can eliminate the redundancy of data. Feature selection can be guided by a human expert in the problem domain. Also, we make the remark that a new technique can be applied for feature selection, in this case the respective technique being added as a new knowledge to KB_1.

Examples of other knowledge related to feature selection (KB_1):

- **Knowledge-1** [107]: "*Selecting the most relevant variables is usually suboptimal for building a predictor, particularly when variables are redundant. A subset of useful variables may exclude many redundant, but relevant variables. Feature selection by using domain knowledge can construct a better set of "ad-hoc" features*".
- **Knowledge-2** [107]: "*Perfectly correlated variables are truly redundant in the sense that no additional information is gained by adding them*".
- **Knowledge-3** [107]: "*Two variables that are useless by themselves can be useful together. A variable that is completely*

General Benchmarking Framework

Input: P

Output: *BestAlg*

Resources:

KB_1 (problem domain knowledge base),

KB_2 (algorithm specific knowledge base),

DB_p (the available datasets for P solving)

CB_{SP} (a case base with solved problems)

1. ***DataPreprocessing***(DB_p);

// select the more informative features of the problem instance

2. F = ***FeatureSelection***(P, KB_1);

// select proper algorithms for solving the problem instance with the corresponding features

3. A_p = ***AlgorithmSelection***(P, F, SM, KB_1, KB_2);

// for each proper algorithm perform configuration, execution and performance evaluation

4. **for** $i = 1$ **to** N **do**

// algorithm parameter tuning for each algorithm (non CI or CI)

4.1 **if** (*A_i is a non CI algorithm) **then**

4.1.1 ***AlgorithmConfiguration***(A_i, SP_{Ai}, F, KB_1; TP_{Ai});

4.1' **else**

4.1'.1 ***AlgorithmConfiguration***(A_i, SP_{Ai}, F, KB_2; TP_{Ai});

4.2 $Solution_i$ = ***ApplyAlgorithm***(A_i, TP_{Ai}, P, F, DB_p);

4.3 ***SolutionTesting*** ($Solution_i$, A_i, TP_{Ai}, P, F, DB_p; Obs);

4.4 ***AlgorithmPerfEval***(A_i, $Solution_i$, $PerfM_i$)

// analyze the performance of all applied algorithms for solving P and choose the best one

5. $BestAlg$ = ***AlgorithmPerfAnalysis***(A_p, $Solutions$, $PerfM$)

// save the solution given by the best algorithm for solving P with learnt lessons (Obs) in the CB_{SP} case base

6. ***AddNewCase***(NewCase(P, F, $BestAlg$, $Solution_{BestAlg}$, Obs), CB_{SP});

// add the new, derived knowledge (included in Obs) to KB_1 and KB_2

7. ***AddNewKnowledge***(Obs, KB_1, KB_2);

return *BestAlg*.

useless by itself can provide a significant performance improvement when taken with others".

3.1.3. Algorithm selection

The selection of proper algorithms for solving the problem instance P with the corresponding set of features, F, is performed by the function ***AlgorithmSelection*** (P, F, SM, KB_1, KB_2) with the support of an algorithm selection mechanism, SM, usually, a machine learning technique, as e.g. regression, clustering and classification or the ranking meta-learning approach [108,109], that performs a mapping between problem instance and algorithms by using F, and of some heuristic knowledge from the two knowledge bases, KB_1 and KB_2. Also, human experts in the problem domain can guide this selection, by adding the most appropriate problem domain methods for solving *GeneralProblem*(P). Some guidelines related to CI algorithms selection in forecasting problems solving will be given in Section 3.2. We want to highlight that it is worthwhile to select for benchmarking in case of forecasting problems solving with CI methods at least two non-CI methods, the most appropriate statistical method and, eventually, the most appropriate domain specific method to solve the current problem (i.e. to extend the comparative study by integrating the three methods types for problem instance solving as shown in Fig. 1). The current form of the framework allows the selection of such algorithms (i.e. non CI algorithms) which are treated in step 4.1 (4.1.1). We make the remark that a new algorithm selection approach can be applied with better results, in this case the respective approach being added as a new knowledge to KB_1 or KB_2 depending on the approach generality degree (i.e. a problem domain dependent approach will be added to KB_1 while a more general one, will be added to KB_2).

3.1.4. Algorithm configuration

Each selected algorithm (CI or non CI algorithm) will be configured by algorithm parameter tuning with the function ***AlgorithmConfiguration*** (A_i, SP_{Ai}, F, KB_1; TP_{Ai}) for non CI algorithms (i.e. for statistical methods and problem domain methods which are included in KB_1 as domain specific knowledge) and with the function ***AlgorithmConfiguration*** (A_i, SP_{Ai}, F, KB_2; TP_{Ai}) for CI algorithms. Algorithm parameter tuning can be implemented with two types of algorithm configurators [110]: non-model-based and model-based approaches. Some of the approaches use

ML techniques in order to predict good parameterizations based on samples from the algorithm parameter space [111]. Finding good, even the best parameters of the algorithm for a specific problem instance is a difficult task, time consuming, depending also on the problem instance (more informative) features. Local search approaches can offer a good alternative. Some guidelines related to specific CI algorithms configuration (included in KB_2 as heuristic knowledge) are given in Section 3.2. We make the remark that a new method for algorithm configuration can be applied, in this case the respective method being added as a new knowledge to KB_2. Also, we point out that during algorithm configuration the training dataset will be used in case the selected algorithm requires a training phase.

3.1.5. Algorithm execution

Each selected algorithm that was previously configured (i.e. its parameters were tuned) is applied to solve the problem instance *P* with the features set *F*. The algorithm execution is performed by the function ***ApplyAlgorithm*** (A_i, TP_{Ai}, *P*, *F*, DB_p) that will return the solution $Solution_i$.

3.1.6. Solution testing

During this step, a good practice of software engineering is applied: performing validation, checking, testing and re-validation of the solution, i.e. if it is feasible or not, and an analysis of the instance data, i.e. if they are complete or not (e.g. some important features are missing). The validation is performed on the validation dataset, while testing is performed on the testing dataset. Some specific testing procedures can be applied as e.g. overfitting testing (for ANN-based forecasting algorithms), solution practical feasibility testing or solution optimality testing. This last procedure can be included in step 5, when the best algorithm is identified. The solution testing is performed by the function ***SolutionTesting*** ($Solution_i$, A_i, TP_{Ai}, *P*, *F*, DB_p; *Obs*) and in case the results are satisfactory it will return some lessons that were learned (*Obs*) and the framework will continue with the next step (4.4), while in case some problems occur than it will return to previous steps (e.g. to algorithm configuration, step 4.1 or step 4.1' or to feature selection, step 2).

3.1.7. Algorithm performance evaluation

The performance of each selected and executed algorithm is evaluated in terms of some performance metrics, $PerfM_i$ (e.g. selected from those given in Section 2.2.2.1). The evaluation will be performed by the function ***AlgorithmPerfEval*** (A_i, $Solution_i$, $PerfM_i$). Examples of criteria that can be used for algorithm performance evaluation are: statistical metrics that minimize the forecasting error such as RMSE, CV-RMSE, MAPE, sMAPE, correlation coefficients such as MCC, R, R^2, other metrics such as AUC, CAR, or metrics for binary classification such as TP/FP, FN/TN, S_n, S_p, and application domain specific metrics.

3.1.8. Algorithm performance analysis

The performance of all algorithms that were applied to problem instance *P* solving will be analyzed according to a performance measure *PerfM* (common to all tested algorithms), and the best algorithm will be chosen (i.e. the algorithm that obtained the optimum *PerfM*). This analysis will be realized by the function ***AlgorithmPerfAnalysis*** (A_p, *Solutions*, *PerfM*). The algorithm performance evaluation can be made with more complex statistical tests [112]: nonparametric tests (e.g. for pairwise comparisons: the Sign test, the Wilcoxon Signed-ranks test; for multiple comparisons: the Friedman test, the Quade test) and Page's trend test (*L*-statistic, *p*-value) for ordered alternatives (similar to Friedman test), as well as statistical procedures (parametric and nonparametric), post-hoc procedures (e.g. for multiple comparisons: Bonferroni, Holm, Hochberg etc.). A ranking of the algorithms will be made. We highlight that statistical tests are commonly used for EA performance comparison.

3.1.9. The case base with solved problems

The best solution of each solved problem instance *P* given by the best algorithm will be saved as a new case in the case base CB_{SP} with solved problems. Also, the lessons learnt during problem *P* solving, *Obs*, will be saved in the case base. Each new case is added to the case base with the function ***AddNewCase*** (NewCase(*P*, *F*, *BestAlg*, $Solution_{BestAlg}$, *Obs*), CB_{SP}). The new case can be included also in an existing benchmarking online repository.

3.1.10. The knowledge bases KB_1 and KB_2

The knowledge base KB_1 includes knowledge related to feature selection and knowledge related to domain methods for solving different types of problems. The knowledge base KB_2 includes knowledge specific to algorithm parameters tuning, and knowledge related to algorithm selection. As the problem instance solving can provide new knowledge under the form of lessons that were learnt, *Obs*, as e.g. a new feature selection technique, a new algorithm selection approach, a new algorithm configuration method, the best algorithm for solving the problem instance, it is useful to add these knowledge to the corresponding knowledge bases, KB_1 and KB_2, not only to the case base. This is achieved with the function ***AddNewKnowledge*** (*Obs*, KB_1, KB_2) that was added as the last step of the framework. For example, a new feature selection technique will be added to KB_1, while a new algorithm configuration method will be added to KB_2. We make the remark that in case *P* is a forecasting problem instance from a specific domain of application, KB_1 will contain apart from the problem domain knowledge and feature selection related knowledge, the statistical methods for solving forecasting problems.

Some examples of generic knowledge (in meta-rule form) from KB_1 which were derived from the application domain oriented survey we have made are given below.

Rule S12
if *ProblemType*(*P*) = "forecasting" **and** *GeneralProblem*(*P*) = "earthquake prediction" **then**
    Recommendations:
    Relevant features: earthquake magnitude, seismic rate changes, earthquake precursors etc;
    Proper algorithms:
        Domain specific method: seismic region specific model, attenuation models etc;
        Statistical method: proper statistical models (e.g. ARIMA);
        CI algorithms: ANN, SVM, WNN, DNN;
    Find more features by applying Rule SF12;
    Find more specific proper algorithms by applying Rule SA5.

Rule A8
if *ProblemType*(*P*) = "forecasting" **and** *GeneralProblem*(*P*) = "ozone prediction" **then**
    Recommendations:
    Relevant features: NO, NO_x, VOC, air temperature, wind speed, precipitations etc;
    Proper algorithms:
        Domains specific method: chemistry transport model, a climatology model etc;
        Statistical method: exponential smoothing, ARMA, ARIMA etc;
        CI algorithms: ANN, ANFIS, GP, SVM, DNN;
    Find more features by applying Rule AF26;
    Find more specific proper algorithms by applying Rule AA2.

3.1.11. General guidelines for the framework implementation

A set of guidelines for the implementation of the framework are synthesized in Table 8. For each guideline, we have given the reference and the framework step where it can be applied.

Viewpoint of problem domain specialists

Specialists from some domains have developed specific algorithms benchmarking platforms for the identification of most performant algorithms. For example, the specialists in solving forecasting problems can use as guidelines the conclusions of the recent M4 competition on forecasting algorithms benchmarking synthesized in [10]. Such guidelines represent valuable knowledge for KB_1.

Example of framework implementation solution

The framework might be implemented as a software tool, for example, as a decision support system (DSS), that will assist the

Table 8
Guidelines for the implementation of the general framework.

Reference	Guideline	Framework step
[113]	Use the benchmark library, ASlib version 2.0.	Step 3
[114]	Empirical performance model. How to build with ML techniques a model of the algorithm's runtime as a function of problem instance features.	Step 4.3
[110]	A benchmarking approach that uses surrogate scenarios — a practical and cheap alternative.	Step 4.3
[115]	A new approach of algorithm configuration that exploit information about algorithm's performance on previous benchmarks in order to warmstart its configuration on new types of benchmarks.	Step 4.1, 4.1'
[111]	Automatically algorithm's parameters setting with GA-based approach under a non-model-based GA framework.	Step 4.1, 4.1'
[108]	Algorithm selection with meta-learning methods.	Step 3, Step 4.1, 4.1'
[109]	The ranking meta-learning approach for ML methods.	Step 3
[116]	The lessons learnt from the survey and perspectives on automated algorithm selection can be used.	Step 3
[117]	Algorithm configuration with AClib: a benchmarking library for algorithm configuration.	Step 4.1, 4.1'
[118]	Guidelines for the identification of suitable optimization algorithms for certain problem instances.	Steps 2-5
[119]	Use the framework for benchmarking CI algorithms, CIlib, developed as an open source library of CI algorithms.	Steps 2-5

software developers (e.g. researchers and engineers from a software company) when applying the framework to new applications implementation. If such a solution is adopted, some artificial intelligence-based modules will be included in the DSS, as for example, an expert system (which is composed of a knowledge base and an inference engine), and potentially, a data mining module added to the database management system for knowledge extraction and rule learning.

Knowledge bases implementation

The knowledge bases (KB_1 and KB_2) that are used as resources by the framework include knowledge represented in the production rule form (i.e. IF-THEN form). Such rules can be stored in some machine readable formats (including semantic schemes), as e.g. XML [120], OWL [121], RuleML [122], or SWRL [123]. If an ontology-based approach is applied for knowledge bases implementation than an ontology will be developed, i.e. a conceptualization of the computational intelligence domain and of the application domain will be made. Thus, the ontology will include the concepts specific to CI algorithms and the application domain along with their relationships and will be stored in a specific data format (as e.g. OWL/XML supported in Protégé ontology editor [124]) and will be correlated with a compatible rule format scheme. The knowledge bases testing can be made with some rule engines as e.g. JESS [125] or DROOLS [126].

3.2. A set of guidelines for CI algorithms selection in forecasting problems solving

Based on our domain of application oriented survey [36–100] and [102–106] we have identified a set of guidelines that can be followed by researchers and engineers when developing new forecasting systems based on CI algorithms. The guidelines are synthesized in Table 9. These guidelines represent expert knowledge from the algorithm specific knowledge base, KB_2.

Other general and specific guidelines for CI algorithms use in solving forecasting problems (derived from the survey we have made) are given in Table 10. These guidelines represent knowledge from the algorithm specific knowledge base, KB_2.

4. Examples of benchmarking framework application

Four case studies with examples of how to apply the guidelines of the benchmarking framework for some problems of forecasting in seismology (earthquake prediction), air pollution (ozone prediction), hydrology (flood prediction) and energy (energy load prediction) are discussed, highlighting the use of heuristic knowledge derived from the domain oriented survey presented in Section 2. For each case study it was selected for benchmarking a statistical forecasting method and a problem domain specific forecasting method. We consider that the datasets specific to each problem instance contains real world data, most of the data being measurements of different parameters taken from (real-time) remote monitoring stations (e.g. stations of the national air quality monitoring network, flood monitoring stations, surface water level monitoring stations, earthquake monitoring stations) or in-situ stations (e.g. in case of soil pollution analysis). We make the remark that our approach is not a software implementation, it is a software engineering guide for developing forecasting applications with CI algorithms, and thus, in the next sections we shall describe only guidelines that can be followed and no details related to case studies implementation.

4.1. Case study 1 — Earthquake prediction

Suppose we want to predict earthquakes in a certain seismic region A1.

Problem Type: forecasting; *General Problem*: earthquake prediction;
Instance Problem P: Predict the magnitude range of future earthquake that will occur in the seismic region A1 and the time interval within which it will happen, performing a short-term prediction (hours to days), giving the datasets specific to region A1, i.e. *Instance Datasets* with measurements recorded from real-time seismology monitoring stations existing in region A1 and other data (e.g. data related to the seismic area A1: geographical characteristics etc.).
Type of data: real world data; *Data sources*: real-time seismology monitoring stations.

In this case, the main resources of the framework are: KB_1 (seismology — earthquake prediction knowledge base), KB_2 (CI algorithms knowledge base), CB_{SP} (the case base with earthquake prediction problems solved in the past for the seismic region A1), the datasets DB_p with past earthquakes that occurred in the specified geographical area A1, including the earthquake magnitude, date, time, and other parameters (e.g. taken from seismic plots).

KB_1 contains knowledge provided by seismologists, geologists, civil engineers, seismic risk managers, geotechnical and structural earthquake engineering. Also, it contains knowledge from literature (e.g. textbooks, books, research papers). For example, the basic knowledge on seismology can be taken from [131], a recent textbook on seismology.

4.1.1. Data preprocessing

Apart from the general data preprocessing methods discussed in Section 3.1.1, some domain specific data preprocessing approaches can be applied. For example, the seismic raw data can be divided by different criteria, such as the maximum earthquake in a given period which is considered the main shock. Also, a statistical analysis and a seismic time series analysis can be performed in order to identify the optimal set of inputs. Some useful earthquake statistics are: frequency–magnitude relations, aftershocks, and earthquake probabilities.

Table 9
Guidelines for CI algorithms selection in forecasting problems solving.

CI Method (forecasting)	Guideline/Heuristic knowledge (prediction/forecasting problem solving)
ANN	– used as a prediction/forecasting method, good results for real-time prediction – weights initialization with SA, PSO, GA – parameters tuning (ANN training) with SA, PSO, GA and ABC – most used CI method (single/hybrid/ensemble) for solving prediction problems
FL	– usually, used as prediction/forecasting method, very good when dealing with uncertainty – used for reasoning on a higher level, with linguistic terms – fuzzy rule selection with GA – suitable for prediction problems as a single or hybrid method
GA	– usually, used as an optimization algorithm – can perform ANN training – can improve fuzzy rule selection in hybrid methods based on ANFIS – alone or in combination with RF can determine an optimal set of problem instance features (identifying the most important features of a problem instance)
GP	– usually, used as an optimization algorithm – valuable for pre-design practices, instead of ANN and other soft computing methods – better that ANN when peak values need to be identified
PSO	– used as an optimization algorithm, better than GA – applied to ANN parameters tuning and ANN training – can perform weights optimization of each layer of an ANN
SA	- used as an optimization algorithm – used for ANN parameters (weights and biases) initialization – used for ANN training (which avoid trapping in local minima)
ABC	– used as an optimization algorithm – used as an ANN learning algorithm instead of BP (which can be trapped in local minima)
WT	– used as an optimization algorithm, efficient in combination with ANNs – used as a decomposition method for the datasets
SVM	– used as prediction/forecasting method – a good predictor (e.g. DO parameter estimation for surface river water quality prediction)
RF	– usually, used in ensemble methods and as an optimization algorithm – in combination with GA can determine optimal features of the problem instance

Table 10
General and specific guidelines for CI algorithms use in forecasting problems solving.

Reference/CI Method	General/Specific Guideline (forecasting problem solving)
[127]	– information and guidelines for multi-step forecasting for big time series based on ensemble learning
[128]	– PSO parameter settings guidelines
[129]	– analysis of PSO applications
[130]	– an evolutionary computation benchmarking repository
CI algorithm parameter settings	– trial and error method, information gain ratio method, PCA (principal components analysis), PSO, GA etc.
Swarm intelligence	– improves the accuracy of prediction/forecast
MCSA [61]	– MCSA used for WNN initialization, improving the accuracy of the forecast
SVM	– a reliable method to compute large nonlinear time series data – performs quite accurately for small sized datasets

4.1.2. Feature selection

A set of heuristic knowledge (under the rule form) derived from the literature were included in KB_1, guiding the feature selection framework step. An example of such generic rule is **Rule S12**, given in Section 3.1.9.

According to [132], the most important features for accurate earthquake prediction are the following:

f_1 — the information provided by the dynamic Gutenberg–Richter's law
f_2 — the information provided by the Omori–Utsu's law
f_3 — the average time elapsed between earthquakes
f_4 — the mean time elapsed between earthquakes
f_5 — the associated standard deviation
f_6 — the mean magnitude of the last earthquake occurred
f_7 — the rate of square root energy release

Thus, the initial set of problem features can be $F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$. Depending on the problem requirements, the set can be reduced to the most relevant features for *P*, by using some rules from KB_1, as the following given ones:

Rule SF-1 (*knowledge source*: [132])
if *short-term prediction **then**
* most valuable features are f_1, f_2.

Rule SF-2 (*knowledge source*: [132])
if * more accurate prediction is needed **then**
* select features with higher correlation (e.g. f_3, f_4, f_5).

Additional features can be added to *F*, as e.g. earthquake precursors, if they exist for the seismic region A1 (i.e. observable behavior that precedes earthquake): foreshocks (i.e. earthquakes that occur before a main shock), changes in the properties of rock within a fault zone preceding a large earthquake (as e.g. the presence of the radon gas). Also, some seismic features generation methods (discussed in [133]) such as frequency-wave number analysis and spatial averaged autocorrelation method can be used.

Other feature selection methods that can be applied are: the maximum relevance and minimum redundancy criteria, information gain (which can eliminate redundant features, but does not take into account features interaction), GA, clustering or a domain specific feature extraction method as the precursory pattern-based extraction method based on information gain introduced in [134].

4.1.3. Algorithm selection

The algorithm selection can be performed by human experts in seismology that will provide domain specific methods (e.g. attenuation models, physical models or on-site investigation models) and human experts in forecasting and CI field that will provide a set of CI algorithms and statistical methods proper for solving *P*. For example, a set of proper algorithms is:

$A_p = \{\text{ANN, SVM, DNN, WNN, ARIMA, AnAttenuationModel}\}$

The algorithms were selected according to the conclusion of the survey presented in Section 2.2.2.2.

4.1.4. Algorithm configuration, execution and evaluation

For each algorithm in A_p it is performed its configuration, execution and performance evaluation, according to specific metrics. An example of knowledge that can guide the ANN algorithm configuration is the use of the SA technique in order to avoid being trapped in local minima (this is a knowledge included in KB_2, according to the guidelines given in Table 9). Other options are PSO or GA.

4.1.5. Algorithm performance evaluation

The metrics that can be selected for algorithm performance evaluation are RMSE, forecasting accuracy, and the coefficient of correlation. The algorithm that provided the best performance in terms of the selected metrics is chosen as the best one for solving *P*.

4.1.6. The knowledge base KB_1

Examples of rules from the KB_1 knowledge base are given below.

Rules derived from [135]:

.....

Rule S1
if * focal depth increases **then**
* seismic intensity decreases;

Rule S2
if * superposition of deep seismic reflection waves **then**
* it is probable to occur high intensity points in low intensity areas (i.e. abnormal intensity);

Comment: the conclusion depends also on the topography and geological structure of the area.

Rule S3
if * focal depth is closer to the epicenter **then**
* greater damage is caused;

Rule derived from [136]:

if * a prediction model gives a correlation coefficient R > 0.8 **then**
* the model shows a strong correlation between the predicted and the measured values.

An example of new knowledge that was deduced from solving past cases (i.e. *Obs*) reported in [51]: DNN has no feature selection due to its complex architecture — as it increase the number of layers and number of neurons in each layer (it exploit the unattended attributes inherent in the input dataset as a whole). The major disadvantage is that it needs more training data and the algorithm run time increases. This knowledge is included in KB_2.

4.2. Case study 2 — Ozone prediction

The problem discussed in the second case study is ozone prediction. Ozone air pollutant (O_3), especially, ground ozone level affecting soil quality, is derived from chemical reactions determined by NO_x, VOC, under certain meteorological conditions (e.g. light wind, warm temperature, clear sky, i.e. stable atmospheric conditions), and is increased by the presence of SO_2 (derived from industry) [137].

Problem Type: forecasting; *General Problem*: ozone prediction;
Instance Problem P: Short-term prediction of ozone concentration in a certain urban area A2 giving DB_p, the instance datasets (usually, time series) with air pollutants concentrations and meteorological parameters measurements taken from the monitoring systems existing in the urban area A2. Examples of monitored air pollutants are: CO, NO, NO_2, NO_x, SO_2, PM_{10}. The main meteorological parameters that can be used are: wind speed, wind direction, solar radiation, relative humidity, atmospheric pressure, air temperature.
Data type: real world data; *Data sources*: air pollution monitoring stations from the national air quality monitoring network, meteorological stations from the national meteorological monitoring network.

In this case, the main resources of the framework are: KB_1 (air pollution prediction knowledge base), KB_2 (CI algorithms knowledge base), CB_{SP} (the case base with ozone prediction problems solved in the past in the urban area A2), and the instance datasets DB_p.

KB_1 contains knowledge provided by air pollution experts. Also, it contains knowledge from literature (e.g. textbooks, books, research papers). For example, the basic knowledge on air pollution can be taken from [137].

For this case study we shall provide some details related to data preprocessing, feature selection and algorithm selection.

4.2.1. Data preprocessing

In this case, data processing will involve an analysis of the datasets for the detection of any pattern such as seasonality or trends that exist in the available air pollutants time series, specific to the urban area A2. Also, other data processing operations will be applied as discussed in Section 3.1.1.

4.2.2. Feature selection

As ozone is dependent on different parameters such as air temperature, season, atmospheric stability, VOC, NO_x, SO_2, these parameters will be selected as features of *P*. Also, past values of ozone will be added to the feature set. Atmospheric stability is given by wind speed, air temperature, precipitations, atmospheric pressure. Therefore, the most important features that influence ozone are:

f_1 — NO_x
f_2 — SO_2 (if there is an industrial source of this pollutant in the analyzed area)
f_3 — VOC
f_4 — air temperature
f_5 — wind speed
f_6 — atmospheric pressure
f_7 — precipitations
$f_8 \div f_{8+p}$ — past values of ozone ($p \geq 2$), at least 3 past values.

An initial set of features is $F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_{8,\dots,} f_{8+p}\}$.

To this set it will be applied a feature selection technique as discussed in 3.1.2. An example of feature selection criterion that was experimented with success for air quality forecasting is partial mutual information as reported in [138]. Other techniques that were used in the surveyed papers are: information gain, GA, GA-RF, a linear method of stepwise fit (optimized approach), Fisher discriminant, forward and backward elimination technique, linear SVM network, RF, and PCA etc.

4.2.3. Algorithm selection

The algorithm selection can be performed by human experts in air pollution that will provide some domain specific methods (e.g. chemistry transport models, climatology models) and human experts in forecasting and CI field that will provide a set of CI algorithms and statistical methods proper for solving *P*. For example, a set of proper algorithms is:

$$A_p = \{\text{ANN, ANFIS, GP, SVM, DNN, exponential smoothing, OzoneSpecificModel}\}$$

In case of using time series with trend, a proper statistical model can be the Holt and Damped exponential smoothing [35]. The set of algorithm performance measure will include IA, RMSE and R.

4.2.4. The knowledge base KB_1

An example of a generic rule from KB_1 that is applied when performing steps 2 and 3 of the benchmarking framework is **Rule A8**, given in Section 3.1.9. Other examples of rules from the KB_1 knowledge base are given below.

Rule A1 (*knowledge source*: [137])
if AirTemp = High **and** Precipitations = None **and** WindSpeed = Low **then** AtmCond = Stable;

Rule O2 (*knowledge source*: [65])
if AtmCond = Stable **and** SO_2(t) = Insignificant **and** NO_x(t) = Significant **and** Ozone(t) = Significant **and** Ozone(t-1) = Significant **and** Ozone(t-2) = Insignificant
then Ozone(t+1) = Significant;

Rule A2
if PollutionSource=Traffic **then** PossibleAirPollutants = {PM_{10}, $PM_{2.5}$, CO, NO_x, VOC, less SO_2};
Rule A3
if PollutionSource=Industry **then** PossibleAirPollutants = {PM_{10}, $PM_{2.5}$, CO, NO_x, VOC, SO_2};
Rule A4
if PollutionSource=Domestic **then** PossibleAirPollutants={PM_{10}, $PM_{2.5}$, CO};

These rules can guide the framework application when selecting relevant features.

4.3. Case study 3 — Flood prediction

The third case study refers to river flood prediction (i.e. prediction of flood lead time and occurrence location). We highlight some details related to feature selection and algorithm selection.

Problem Type: forecasting; *General Problem*: river flood prediction;
Instance Problem P: flood prediction in a natural river R giving the *Instance Datasets* with hydrological data related to the river R (water level, river flow, peak flow, daily flow etc.) measured by hydrology monitoring stations existing along the river, meteorological dataset and other hydro-geomorphological datasets specific to river R.
Type of data: real world data; *Data sources*: hydrology monitoring stations.

KB_1 contains knowledge provided by hydrologist experts and knowledge taken from literature (e.g. textbooks, books, research papers). Also, the knowledge derived from our survey on hydrology applications is part of this knowledge base.

4.3.1. Feature selection

A possible set of features are geomorphological features of river R area, lead time of flood forecasting, past precipitations, upstream river flow, meteorological parameters (e.g. min, max temperature, mean relative humidity, mean wind speed, evaporation), season etc.

For example, a minimum set of features is $F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$, where f_1 is minimum temperature, f_2 is the maximum temperature, f_3 is the mean relative humidity, f_4 is the mean wind speed, f_5 is the evaporation, f_6 is the rainfall and f_7 is the river flow. An additional relevant feature is the upstream river flow that was added to F as feature f_8. The features were selected by using domain knowledge from KB_1. An example of feature selection method that was experimented with success on very short-term heavy rainfall prediction (that can influence the river flood prediction result) is wrapper-based genetic feature selection using ML techniques such as SVM or k-NN [139].

4.3.2. Algorithm selection

A set of proper algorithms for this case study is the following one:

$$A_p = \{\text{ANN, ANFIS, GP, SVR, MLR, RiverSpecificHydrologyModel}\}$$

Domain specific prediction methods can be taken from the results of the hydrology domain surveyed papers. Other examples of proper methods are: GP, GEP, LR, SVM, exponential smoothing, ARIMA, physical based models (e.g. rainfall/runoff, storm, hydraulic models of flow), climatology models (e.g. GCM-based model — a global climate model). We have to point out that some human experts agree on the fact that for peak flood forecasting, statistical methods are inadequate due to the problem complexity and they propose better methods such as GP and GEP (this is a knowledge derived from [81] that was included in KB_1 and was used during framework application).

4.4. Case study 4 — energy

The last case study refer to energy load forecasting based on weather data. Apart from the electricity energy, two alternative sources of green energy are considered: wind and solar energy. We highlight some details related to feature selection and algorithm selection.

Problem Type: forecasting; *General Problem*: energy load forecasting;
Instance Problem P: energy load forecasting in an urban area A3 giving the *Instance Datasets* with historical energy consumption, weather information, the time period, different hours of the weekdays, weekends, holidays that can influence the energy demand in the urban area A3. Also, some additional information specific to the urban area A3 can be used: electricity price tariff, economic and social factors.
KB_1 contains knowledge provided by experts in energetic industry and knowledge taken from literature (e.g. textbooks, books, research papers). Also, the knowledge derived from our survey on energy applications is part of this knowledge base.

4.4.1. Feature selection

Taking into account the knowledge from KB_1 a possible initial set of relevant features is the following:

$$F = \{f_1, f_2, f_3, f_4, f_5, f_6, \ldots, f_{6+p}\}$$

f_1 — wind speed
f_2 — wind direction
f_3 — solar energy
f_4 — maximum air temperature
f_5 — relative humidity
$f_6 \div f_{6+p}$ — past values of energy consumption ($p \geq 2$), at least 3 past values (weekly, monthly)

According to KB_1 two examples of feature selection methods that can be used are: PCA and techniques based on SVM.

4.4.2. Algorithm selection

A set of proper algorithms for this case study can be the following one:

A_p = {ANN, PSO, LSSVM, ARIMA, EnergeticSpecificModel}

Domain specific prediction methods can be taken from the results of the energy domain surveyed papers. Other examples of proper methods are: RF, LSTM and DNN.

4.5. Discussion

The application of the benchmarking framework for a particular problem will follow the steps detailed in Section 3.1. We make the remark that the heuristic knowledge included in KB_1 and KB_2 was taken from the surveyed literature (on forecasting, on CI algorithms and specific application domain oriented surveys). Also, we highlight that the first option when starting to apply each step of the benchmarking is to use human experts' knowledge that was integrated in KB_1 and KB_2. The main benefit of applying a knowledge-based benchmarking framework as the one we proposed for solving forecasting problems is a higher efficiency (i.e. higher forecasting accuracy, and shorter response time), as each step is guided by expert knowledge and the best selection is made (either of the features or of the proper algorithms). Also, we consider that a decision support system for benchmarking framework application would be extremely useful, this being a potential future work.

At the end of this discussion we make a brief reference to some benchmarks developed in some Computational Intelligence domains: evolutionary computation, neural computing and machine learning. Also, we mention a forecasting algorithms competition.

The practical application of benchmarking in the context of evolutionary computation was tackled in different publications (e.g. [112,140]). Two examples of benchmarking developed in the evolutionary computation domain are: the black box optimization benchmarking (BBOB) [141], and the IEEE Congress on Evolutionary Computation (CEC) (see. e.g. [142] for CEC 2013). Most of the EC benchmarking are presented and analyzed under the Genetic and Evolutionary Computing Conference (GECCO) (see a recent proposal of single and multi-objective game benchmark described in [143], presented at GECCO 2019). Several comparisons of EC algorithms under these two benchmarks (BBOB and CEC) were already presented in the literature (see e.g. [144] for CEC and [141] for BBOB), providing practical guidelines for researchers. Other examples of benchmarks are: NN, in the neural computing domain, developed for time series competition (see e.g. NN5 [145]), and Penn Machine Learning Benchmark (PMLB) [146], a benchmark developed in the area of machine learning that includes most of the real-world benchmark datasets. Related to forecasting benchmarking, we mention the M competition (see [10] for the last edition, M4). The experience accumulated under existing CI benchmarks can provide valuable knowledge. Future research work may involve an analysis of such benchmarks use for solving forecasting problems.

5. Conclusion and future work

The paper proposes a set of guidelines for the selection of the CI algorithms proper to solve forecasting problems, based on a survey of selected papers reported recently in the literature in various domains (e.g. seismology, hydrology, environmental protection, energy, materials science, and engineering). Also, we have described a general benchmarking framework, adaptable to CI algorithms benchmarking that uses two knowledge bases, one application domain related and one for CI algorithm parameters tuning, and a case base with solved problems. Our approach provides guidelines that can be followed by software developers (e.g. researchers, engineers) of new CI applications. It is a meta-model that integrates software engineering and knowledge engineering best practices towards CI benchmarking, being a CI engineering methodology. The framework might be implemented as a software tool. The main conclusion of our preliminary study is that valuable knowledge can be extracted from the research work performed in the computational intelligence field that is reported in the literature, especially from the benchmarking of CI algorithms. These knowledge can be used as guidelines of developing more accurate forecasting systems.

Future work involves the extension of the survey to more papers and domains of application of CI algorithms in order to provide additional guidelines for the selection of the best algorithm for a given problem. Related to our approach implementation, we intend to develop a decision support system for benchmarking framework application to forecasting problems solving and to perform an evaluation of it on some case studies.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2020.106103.

Acknowledgments

The author wishes to thank the anonymous referees for their valuable comments and suggestions that contributed considerable to the improvement of the paper quality.

Appendix. List of abbreviations

ABC — Artificial Bee Colony
ACF — Autocorrelation function
ACO — Ant Colony Optimization
AdaBoost — Adaptive Boosting
ANFIS — Adaptive Neuro-Fuzzy Inference System
ANN — Artificial Neural Network
ARMA — Autoregressive Moving Average
ARIMA — Autoregressive Integrated Moving Average
ARV — Average Relative Variance
ARX — AutoRegressive with eXogeneous terms
AQI — Air Quality Index
AUC — Area Under the Curve of success
BA — Bees Algorithm
BN — Bayesian Network
BP — Backpropagation
BPNN — Backpropagation NN
CAR — Classification Accuracy Rate
CART — Classification and Regression Trees
CB — Case Base
CHAID — Chi-square Automatic Interaction Detector
CI — Computational Intelligence
CNN — Convolutional Neural Network
CR — Caching Rate
CS — Cuckoo Search
CSA — CS Algorithm
CT — Classification Tree
CTM — Chemistry-Transport Model
CV-RMSE — Cumulative Variation RMSE
DE — Differential Evolution
DM — Data Mining
DNN — Deep NN
DO — Dissolved Oxygen

DSS — Decision Support System
DT — Decision Tree
EA — Evolutionary Algorithm
EC — Evolutionary Computation
EDSS — Environmental Decision Support System
EI — Efficiency Index
ELM — Extreme Learning Machine
ENN — Elman NN
EP-GPBoost — Earthquake Predictor-GPBoost
EvS — Evolutionary Strategy
ES — Exponential Smoothing
ES-RNN — ES-Recurrent Neural Network
FA — Firefly Algorithm
FCM — Fuzzy c-means data clustering
FFANN — Feed Forward Artificial Neural Network
FIS — Fuzzy Inference System
FL — Fuzzy Logic
FN — False Negative rate
FP — False Positive rate
FWRBF — Fuzzy Wavelet Radial Basis Function
GA — Genetic Algorithms
GABC — Guided Artificial Bee Colony
GCM — Global Climate Model
GEP — Gene Expression Programming
GGABC — Gbest Guided ABC
GIS — Geographical Information System
GMDH — Group Method of Data Handling
GP — Genetic Programming
GRNN — Generalized Regression NN
GWO — Grey Wolf Optimizer
HGABC — Hybrid GABC
HNN — Hybrid Neural Network
HRN — Hit Rate of Non-precipitation
HRP — Hit Rate of Precipitation
HS — Harmony Search
IA — Index of Agreement
ICA — Imperialistic Competitive Algorithm
IWO — Invasive Weed Optimization
JMA — Japan Meteorological Agency
KB — Knowledge Base
LGP — Linear GP
LM — Levenberg–Marquardt
LMT — Logistic Model Tree
LR — Linear Regression
LS- Least Squares
LSM — Least Squares Method
LSSVM — Least Square SVM
LSTM — Long Short-Term Memory
MAE — Mean Absolute Error
MAPE — Mean Absolute Percentage Error
MARE — Mean Absolute Relative Error
MAX — Maximum Absolute Difference
MB — Mean Bias
MBE — Mean Bias Error
MCC — Mattews Correlation Coefficient
MCSA — Modified Cuckoo Search Algorithm
MGGP — Multi-Gene GP
ML — Machine Learning
MLP — Multi Layer Perceptron
MLR — Multiple Linear Regression
MNB — Mean Normalized Bias
MNGE — Mean Normalized Gross Error
MSE — Mean Square Error
NCFM — Novel Combined Forecasting Model
NMSE — Normalized Mean Square Error
NGO — Neuro-Genetic Model
NN — Neural Network
NRMSE — Normalized RMSE
NS — Nash–Sutcliffe efficiency coefficient
OLS — Orthogonal Least Squares
OR — Overlooking Rate
OS-ELM — Online Sequential ELM
PACF — Partial ACF
PCA — Principal Component Analysis
PGA — Peak Ground Acceleration
PGD — Peak Ground Displacement
PGV — Peak Ground Velocity
PM — Particulate Matter (an air pollutant)
PSO — Particle Swarm Optimization
R — Correlation coefficient
R^2 — Coefficient of determination
RANFIS — Randomized ANFIS
RBFNN — Radial Basis Function Neural Network
RF — Random Forest
RMSE — Root Mean Square Error
RMSPE — Root Mean Square Percentage Error
RNN — Recurrent Neural Network
RO — Random Optimization
RSM — Response Surface Model
S_n — Sensitivity
S_p — Specificity
SA — Simulated Annealing
SC — Soft Computing
SI — Swarm Intelligence
SMR — Swing-and Miss Rate
SOM — Self-Organizing Maps
sMAPE — symmetric MAPE
SRM — Structural Risk Minimization
SSE — Sum of Squared Error
SVM — Support Vector Machine
SVR — Support Vector Regression
SVR-HNN — SVR-Hybrid Neural Network
SWARA — Step-Wise Assessment Ratio Analysis
THR — Total Hit Rate
TN — True Negative rate
TP — True Positive rate
TSFIS — Takagi–Sugeno FIS
TSK — Takagi-Sugeno Kang
USBM — United States Bureau of Mines
WNF — Wavelets Neuro-Fuzzy
WNN — Wavelet Neural Network
WQI — Water Quality Index
WSP — Wasp Swarm Optimization
WT — Wavelet Transform

References

[1] J. Konar, Computational Intelligence: Principles, Techniques, and Applications, Springer, 2005.
[2] X.-S. Yang, Nature-Inspired Optimization Algorithms, Elsevier, 2014.
[3] X.-S. Yang, Optimization Techniques and Applications with Examples, Wiley, 2018.
[4] H.A. Bouarara, A survey of computational intelligence algorithms and their applications, in: Chapter in Handbook of Research on Soft Computing and Nature-Inspired Algorithms, IGI Global, 2017, pp. 133–176.
[5] A. Mosavi, P. Ozturk, K.-W. Chau, Flood prediction using machine learning models: Literature review, Water 10 (2018) 1536, http://dx.doi.org/10.3390/w10111536.
[6] F. Fotovatikhah, M. Herrera, S. Shamshirband, K.-W. Chau, S.F. Ardabili, Md. J. Piran, Survey of computational intelligence as basis to big flood management: Challenges, research directions and future work, Eng. Appl. Comput. Fluid Mech. 12 (1) (2018) 411–437.
[7] S.N. Fallah, R.C. Deo, M. Shojafar, M. Conti, S. Shamshirband, Computational intelligence approaches for energy load forecasting in smart energy

management grids: State of the art, future challenges, and research directions, Energies 11 (2018) 596, http://dx.doi.org/10.3390/en11030596.
[8] N.K. Ahmed, A.F. Atiya, N.E. Gayar, H. El-Shishiny, An empirical comparison of machine learning models for time series forecasting, Econometric Rev. 29 (5–6) (2010) 594–621, http://dx.doi.org/10.1080/07474938.2010.481556.
[9] S. Makridakis, E. Spiliotis, V. Assimakopoulos, Statistical and machine learning forecasting methods: Concerns and ways forward, PLoS One 13 (3) (2018) http://dx.doi.org/10.1371/journal.pone.0194889.
[10] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The M4 competition: Results, findings, conclusion and way forward, Int. J. Forecast. 34 (2018) 802–808, http://dx.doi.org/10.1016/j.ijforecast.2018.06.001.
[11] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[12] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice Hall, 1999.
[13] R. Hecht-Nelson, Kolmogorov's mapping neural network existence theorem, in: Proc. of the 1st IEEE Annual Int. Conf. on Neural Networks, Vol. 3, IEEE Press, San Diego, 1987, pp. 11–14.
[14] J. Cybenko, Approximations by superpositions of a sigmoidal function, Math. Control Signals Syst. 2 (1989) 303–314.
[15] J.H. Holland, Genetic algorithms and the optimal allocation of trials, SIAM J. Comput. 2 (2) (1973) 88–105.
[16] D. Goldberg, Genetic Algorithms in Search Optimization Machine Learning, Pearson Education, NJ, 2013.
[17] T. Friedrich, M. Wagner, Seeding the initial population of multi-objective evolutionary algorithms: A computational study, Appl. Soft Comput. 33 (2015) 223–230.
[18] M.A. Jayaram, Optimization performance evaluation of evolutionary algorithms: A design problem, in: V.V. Das, N. Thankachan (Eds.), CIIT 2011, CCIS 250, Springer-Verlag, Berlin Heidelberg, 2011, pp. 326–333.
[19] J.R. Koza, Genetic Programming II: Automatic Discovery of Reusable Programs, MIT Press, Cambridge Massachusetts, 1994.
[20] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[21] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[22] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, 2004.
[23] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proc. of the IEEE Int. Conf. on Neural Networks, IV, 1995, pp. 1942–1948.
[24] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: Artificial bee colony (ABC) algorithm and applications, Artif. Intell. Rev. 24 (2014) 21–57.
[25] A.E. Márquez, C. Expósito-Izquierdo, An overview of the last advances and applications of artificial bee colony algorithm, in: Chapter in Handbook of Research on Soft Computing and Nature-Inspired Algorithms, IGI Global, 2017, pp. 520–540.
[26] V. Vapnik, S. Golowich, A. Smola, Support vector method for function approximation, regression estimation, and signal processing, in: Advances in Neural Information Processing Systems, MIT Press, 1997.
[27] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machine and Other Kernel-Based Learning Methods, Cambridge, UK, Cambridge University Press, 2000.
[28] H. Drucker, C. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, in: Advances in Neural Information Processing Systems, Vol. 9, MIT Press, 1997, pp. 155–161.
[29] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, Singapore, 2002.
[30] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[31] R. Jang, ANFIS: Adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (3) (1993) 665–685.
[32] Q. Zhang, A. Benveniste, Wavelet networks, IEEE Trans. Neural Netw. 3 (1992) 889–898.
[33] R. Polikar, Ensemble Learning, Springer, US, 2012, pp. 1–34.
[34] R.J. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice, third ed., OTexts, Melbourne, Australia, 2019 (Online version accessed on 11 July 2019).
[35] P.J. Brockwell, R.A. Davis, Introduction to Time Series and Forecasting, third ed., Springer, 2016.
[36] J. Reyes, A. Morales-Esteban, F. Martínez-Álvarez, Neural networks to predict earthquakes in Chile, Appl. Soft Comput. 13 (2) (2013) 1314–1328, http://dx.doi.org/10.1016/j.asoc.2012.10.014.
[37] H. Furuta, Y. Nomura, Time series prediction of earth quake input by using soft computing, in: Proc. of the 4th Int. Symp. on Uncertainty Modeling and Analysis – ISUMA 2003, 2003, pp. 351–356, http://dx.doi.org/10.1109/ISUMA2003.1236185.
[38] H. Shah, R. Ghazali, T. Herawan, N. Khan, M.S. Khan, Hybrid guided artificial bee colony algorithm for earthquake time series data prediction, in: Communication Technologies, Information Security and Sustainable Development, in: Book Series: Communications in Computer and Information Science, vol. 414, Springer, 2014, pp. 204–215, http://dx.doi.org/10.1007/978-3-319-10987-9_19.
[39] K.M. Asim, A. Idris, T. Iqbal, F. Martínez-Álvarez, Earthquake prediction model using support vector regressor and hybrid neural networks, PLoS One 13 (7) (2018) e0199004, http://dx.doi.org/10.1371/journal.pone.0166004.
[40] G. Asencio-Cortés, F. Martínez-Álvarez, A. Troncoso, A. Morales-Esteban, Medium–large earthquake magnitude prediction in Tokyo with artificial neural networks, Neural Comput. Appl. 28 (5) (2017) 1043–1055.
[41] H. Adeli, A. Panakkat, A probabilistic neural network for earthquake magnitude prediction, Neural Netw. 22 (7) (2009) 1018–1024, http://dx.doi.org/10.1016/j.neunet.2009.05.003.
[42] K.M. Asim, M. Awais, F. Martínez-Álvarez, T. Iqbal, Seismic activity prediction using computational intelligence techniques in northern Pakistan, Acta Geophys. 65 (5) (2017) 919–930, http://dx.doi.org/10.1007/s11600-017-0082-1.
[43] A. Morales-Esteban, F. Martínez-Álvarez, J. Reyes, Earthquake prediction in seismogenic areas of the Iberian Peninsula based on computational intelligence, Tectonophysics 593 (2013) 121–134.
[44] G. Asencio-Cortés, A. Morales-Esteban, X. Shang, F. Martínez-Álvarez, Earthquake prediction in California using regression algorithms and cloud-based big data infrastructure, Comput. Geosci. 115 (2018) 198–210.
[45] F. Martínez-Álvarez, A. Troncoso, A. Morales-Esteban, J.C. Riquelme, Computational intelligence techniques for predicting earthquakes, in: E. Corchado, M. Kurzyński, M. Woźniak (Eds.), HAIS 2011, Part II, LNAI 6679, 2011, pp. 287–294.
[46] K.M. Asim, A. Idris, T. Iqbal, F. Martínez-Álvarez, Seismic indicators based earthquake predictor system using genetic programming and adaboost classification, Soil Dyn. Earthq. Eng. 111 (2018) 1–7.
[47] I.E. Mulia, T. Asano, Initial tsunami source estimation by inversion with an intelligent selection of model parameters and time delays, J. Geophys. Res.-Oceans 121 (1) (2016) 441–456, http://dx.doi.org/10.1002/2015JC010877.
[48] K. Hadihardaja, H. Latief, I.E. Mulia, Decision support system for predicting tsunami characteristics along coastline areas based on database modelling development, J. Hydroinform. 13 (1) (2011) 96–109.
[49] M. Gandomi, M. Soltanpour, M.R. Zolfaghari, A.H. Gandomi, Prediction of peak ground acceleration of Iran's tectonic regions using a hybrid soft computing technique, Geosci. Front. 7 (2016) 75–82.
[50] S.R. Garcia, M.P. Romo, J.M. Mayoral, Estimation of peak ground accelerations for Mexican subduction zone earthquakes using neural networks, Geofisica Internac. 46 (1) (2007) 51–63.
[51] A. Derakhshani, A.H. Foruzan, Predicting the principal strong motion parameters: A deep learning approach, Appl. Soft Comput. 80 (2019) 192–201.
[52] Z. Sen, Supervized fuzzy logic modeling for building earthquake hazard assessment, Expert Syst. Appl. 38 (12) (2011) 14564–14573, http://dx.doi.org/10.1016/j.eswa.2011.05.026.
[53] S. Gholizadeh, E. Salajegheh, Optimal seismic design of steel structures by an efficient soft computing based algorithm, J. Construct. Steel Res. 66 (1) (2010) 85–95, http://dx.doi.org/10.1016/j.jcsr.2009.07.006.
[54] S. Akib, S. Rahman, S. Shamshirband, D. Petkovic, Soft computing methodologies for estimation of bridge girder forces with perforations under tsunami wave loading, Bull. Earthq. Eng. 13 (3) (2015) 935–952, http://dx.doi.org/10.1007/s10518-014-9656-3.
[55] A.H. Gandomi, A.H. Alavi, A new multi-gene genetic programming approach to non-linear system modeling. Part II: Geotechnical and earthquake engineering problems, Neural Comput. Appl. 21 (1) (2012) 189–201, http://dx.doi.org/10.1007/s00521-011-0735-y.
[56] K. Siwek, S. Osowski, Data mining methods for prediction of air pollution, Int. J. Appl. Math. Comput. Sci. 26 (2) (2016) 467–478, https://doi.org/10.1515/amcs-2016-0033.
[57] S.Q. Dotse, M.I. Petra, L. Dagar, L.C. De Silva, Application of computational intelligence techniques to forecast daily PM10 exceedances in Brunei Darussalam, Atmos. Pollut. Res. 9 (2) (2018) 358–368, https://doi.org/10.1016/j.apr.2017.11.004.
[58] I. Bougoudis, K. Demertzis, L. Iliadis, HISYCOL a hybrid computational intelligence system for combined machine learning: The case of air pollution modeling in Athens, Neural Comput. Appl. 27 (5) (2016) 1191–1206, http://dx.doi.org/10.1007/s00521-015-1927-7.
[59] K. Siwek, S. Osowski, Feature selection methods and ensemble of predictors for prediction of air pollution, in: Int. Work-Conference on Time Series (ITISE 20), 2014, pp. 1207–1217.
[60] H.S. Kim, I. Park, C.H. Song, K. Lee, J.W. Yun, H.K. Kim, M. Jeon, J. Lee, Development of daily PM_{10} and $PM_{2.5}$ prediction system using a deep long short-term memory neural network model, in: Atmospheric Chemistry and Physics Discussions, 2019, http://dx.doi.org/10.5194/acp-2019-268.
[61] P. Ong, Z. Zainuddin, Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction, Appl. Soft Comput. J. 80 (2019) 374–386.

[62] V.D. Anezakis, K. Dermetzis, L. Iliadis, S. Spartalis, Fuzzy cognitive maps for long-term prognosis of the evolution of atmospheric pollution, based on climate change scenarios : The case of athens, in: Computational Collective Intelligence, in: LNAI, vol. 9875, 2016, pp. 175–186, http://dx.doi.org/10.1007/978-3-319-45243-2_16.
[63] M. Oprea, S. Mihalache, M. Popescu, Computational intelligence-based $PM_{2.5}$ air pollution forecasting, Int. J. Comput. Commun. Control 12 (3) (2017) 365–380.
[64] F.S. de Albuquerque Filho, F.M.S. Fernandes, P.S.G. de Mattos Nelo, T.A.E. Ferreira, Time-series forecasting of pollutant concentration levels using particle swarm optimization and artificial neural networks, Quim. Nova 36 (6) (2013) 783–789.
[65] M. Oprea, A knowledge modelling framework for intelligent environmental decision support systems and its application to some environmental problems, Environ. Model. Softw. 110 (12) (2018) 72–94, http://dx.doi.org/10.1016/j.envsoft.2018.09.001.
[66] S. Emamgholizadeh, K. Bahman, S.M. Bateni, H. Ghorbani, I. Marofpoor, J.R. Nielson, Estimation of soil dispersivity using soft computing approaches, Neural Comput. Appl. 28 (Suppl. 1) (2017) S207–S216, http://dx.doi.org/10.1007/s00521-016-2320-x.
[67] E. Olyaie, H.Z. Abyaneh, A.D. Mehr, A comparative analysis among computational intelligence techniques for dissolved oxygen prediction in Delaware River, Geosci. Front. 8 (2017) 517–527.
[68] R.R. Li, Y.Q. Dong, Z.J. Zhu, C. Li, H.F. Yang, A dynamic evaluation framework for ambient air pollution monitoring, Appl. Math. Model. 65 (2019) 52–71, http://dx.doi.org/10.1016/j.apm.2018.07.052.
[69] S.S. Tikhe, K.C. Khare, S.N. Londhe, Multicity seasonal air quality index forecasting using soft computing techniques, Adv. Environ. Res. 4 (2) (2015) 83–104, http://dx.doi.org/10.12989/aer2015.4.2.083.
[70] P. Hajek, V. Olej, Predicting common air quality index – the case of Czech Microregions, Aerosol Air Qual. Res. 15 (2015) 544–555, http://dx.doi.org/10.4209/aaqr.2014.08.0154.
[71] Z.M. Yaseen, M.M. Ramal, L. Diop, O. Jaafar, V. Demir, O. Kisi, Hybrid adaptive neuro-fuzzy models for water quality index estimation, Water Resour. Manag. 32 (7) (2018) 2227–2245, http://dx.doi.org/10.1007/s11269-018-1915-7.
[72] B. Yadav, S. Ch, S. Mathur, J. Adamowski, Discharge forecasting using an online sequential extreme learning machine (OS-ELM) model: A case study in Necka River, germany, Measurement 92 (2016) 433–445, http://dx.doi.org/10.1016/j.measurement.2016.06.042.
[73] Z. Mustafa, M.H. Sulaiman, F. Ernawan, N.K.M. Noor, Short term forecasting based on hybrid least squares support vector machines, Adv. Sci. Lett. 24 (10) (2018) 7455–7460, http://dx.doi.org/10.1166/as1.2018.12958.
[74] J. Sulaiman, S.H. Wahab, Heavy rainfall forecasting model using artificial neural network for flood prone area, in: IT Convergence and Security 2017, Vol. 1, in: Lecture Notes in Electrical Engineering, vol. 449, Springer, 2018, pp. 68–76, http://dx.doi.org/10.1007/978-981-10-6451-7_9.
[75] H. Badrzadeh, R. Sarukkalige, A.W. Jayawardena, Hourly runoff forecasting for flood risk management: Application of various computational intelligence models, J. Hydrol. 529 (part 3) (2015) 1633–1643, http://dx.doi.org/10.1016/j.jhydrol.2015.07.057.
[76] S.W. Fleming, D.R. Bourdin, D. Campbell, R.B. Stull, T. Gardner, Development and operational testing of a super-ensemble artificial intelligence flood forecast model for a Pacific North West river, J. Am. Water Resour. Assoc. 51 (2) (2015) 502–512, http://dx.doi.org/10.1111/jawr.12259.
[77] M. Oprea, V. Buruiana, A. Matei, A microcontroller-based intelligent system for real-time flood alerting, Int. J. Comput. Commun. Control 5 (5) (2010) 844–851.
[78] W.C. Wang, K.-W. Chau, C.-T. Cheng, L. Qiu, A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series, J. Hydrol. 274 (3–4) (2009) 294–306.
[79] K. Chapi, et al., A novel hybrid artificial intelligence approach for flood susceptibility assessment, Environ. Model. Softw. 95 (2017) 229–245.
[80] Z. Huang, J. Zhou, L. Song, Y. Lu, Y. Zhang, Flood disaster loss comprehensive evaluation model based on optimization support vector machine, Expert Syst. Appl. 37 (5) (2010) 3810–3814.
[81] N. Seckin, A. Guven, Estimation of peak flood discharges at ungauged sites across Turkey, Water Resour. Manag. 26 (9) (2012) 2569–2581, http://dx.doi.org/10.1007/s11269-012-0033-1.
[82] Y. Chidthong, H. Tanaka, S. Supharatid, Developing a hybrid multi-model for peak flood forecasting, Hydrol. Process. 23 (12) (2009) 1725–1738, http://dx.doi.org/10.1002/hyp.7307.
[83] N. Zaini, M.A. Malek, M. Yusoff, N.H. Mardi, S. Norhisham, Daily river flow forecasting with hybrid support vector machine – particle swarm optimization, in: IOP Conf. Series: Earth and Environmental Science, Vol. 140, IOP Publishing, 2018, 012035, http://dx.doi.org/10.1088/1755-1315/140/1/012035.
[84] J.T. Esteves, G.D. Rolim, A.S. Ferrando, Rainfall prediction methodology with binary multilayer perceptron neural networks, Clim. Dynam. 52 (3–4) (2019) 2319–2331, http://dx.doi.org/10.1007/s00382-018-4252-x.
[85] T. Kashiwao, K. Nakayama, S. Ando, K. Ikeda, M. Lee, A. Bahadori, A neural network-based local rainfall prediction system using meteorological data on the internet: A case study using data from the Japan Meteorological Agency, Appl. Soft Comput. 56 (2017) 317–330.
[86] N. Diodato, G. Bellocchi, Using historical precipitation patterns to forecast daily extremes of rainfall for the coming decades in Naples (Italy), Geosciences 8 (2018) 293, http://dx.doi.org/10.3390/geosciences8080293.
[87] D.T. Bui, M. Panahi, H. Shahabi, V.P. Singh, A. Shirzadi, K. Chapi, K. Khosravu, W. Chen, S. Panahi, S. Li, B.B. Ahmad, Novel hybrid evolutionary algorithms for spatial prediction of floods, Sci. Rep. 8 (2018) 15364, http://dx.doi.org/10.1038/s41598-018-33755-7.
[88] P.-T.T. Ngo, N.-D. Hoang, B. Pradhan, Q.K. Nguyen, X.T. Tran, Q.M. Nguyen, V.N. Nguyen, P. Samui, D.T. Bui, A novel hybrid swarm optimized multilayer neural network for spatial prediction of flash floods in tropical areas using sentinel-1 SAR imagery and geospatial data, Sensors 18 (2018) 3704, http://dx.doi.org/10.3390/s18113704.
[89] B. Sahoo, P.K. Bhaskaran, Prediction of storm surge and coastal inundation using artificial neural network — A case study for 1999 Odisha super cyclone, Weather Clim. Extrem. 23 (2019) (1999) 100196.
[90] C.C. Wei, Soft computing techniques in ensemble precipitation nowcast, Appl. Soft Comput. 13 (2) (2013) 793–805, https://doi.org/10.1016/j.asoc.2012.10.006.
[91] K. Khosravi, M. Panahi, D.T. Bui, Spatial prediction of groundwater spring potential mapping based on an adaptive neuro-fuzzy system and metaheuristic optimization, Hydrol. Earth Syst. Sci. 22 (2018) 4771–4792, http://dx.doi.org/10.5194/hess-22-4771-2018.
[92] P.-H. Kuo, C.-J. Huang, A high precision artificial neural networks model for short-term energy load forecasting, Energies 11 (2018) 213, http://dx.doi.org/10.3390/en11010213.
[93] D. Niu, J. Li, J. Li, D. Liu, Middle-power load forecasting based on particle swarm optimization, Comput. Math. Appl. 57 (2009) 1883–1889.
[94] A.A.E. Ahmed, L.T. Germino, C.Z. Antonio, A hybrid PSO applied to loss power minimization, IEEE Trans. Power Syst. 20 (2) (2005) 859–866.
[95] J.F. Torres, A. Troncoso, I. Koprinska, Z. Wang, F. Martínez-Álvarez, Deep learning for big data time series forecasting applied to solar power, in: Proceedings of the International Joint Conference SOCO'18-CISIS'18-ICEUTE'18, Springer Int. Publisher, 2019, pp. 123–133.
[96] L.E. Vafaei, M. Sah, Predicting efficiency of flat-plate solar collector using a fuzzy inference system, in: 9th Int. Conf. on Theory and Application of Soft Computing, Computing with Words and Perception, ICSCCW 2017, Procedia Comput. Sci. 120 (2017) 221–228.
[97] A. Chaouachi, R.M. Kamel, R. Ichikawa, H. Hayashi, K. Nagasaka, Neural network ensemble-based solar power generation short-term forecasting, in: Engineering and Technology, World Academy of Science, 2009, p. 54.
[98] E. Vladislavlela, T. Friedrich, F. Neumann, M. Wagner, Predicting the energy output of wind farms based on weather data: Important variables and their correlation, Renew. Energy 50 (2013) 236–243.
[99] O. Kramer, F. Gieseke, Short-term wind energy forecasting using support vector regression, in: Int. Conf. on Soft Computing Models in Industrial and Environmental Applications, Springer, 2011, pp. 271–280.
[100] P. Jiang, Z. Liu, Variable weights combined model based on multi-objective optimization for short-term wind speed forecasting, Appl. Soft Comput. J. 82 (2019) 105587, http://dx.doi.org/10.1016/j.asoc.2019.105587.
[101] S. Hochreiter, J. Urgen Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.
[102] A. AminShokravi, H. Eskandar, A.M. Derakhsh, H.N. Rad, A. Ghanadi, The potential application of particle swarm optimization algorithm for forecasting the air-overpressure induced by mine blasting, Eng. Comput. 34 (2018) 277–285.
[103] W. Ye, C. Chen, Z. Wang, I.-H. Chu, S.P. Ong, Deep neural networks for accurate predictions of crystal stability, Nature Commun. 9 (2018) 3800, http://dx.doi.org/10.1038/s41467-018-06322-x.
[104] P.V. Bushlanov, V.A. Blatov, A.R. Oganov, Topology-based crystal structure generator, Comput. Phys. Comm. 236 (2019) 1–7, http://dx.doi.org/10.1016/j.cpc.2018.09.016.
[105] J. Hu, P. Gao, Y. Yao, X. Xie, Traffic flow forecasting with particle swarm optimization support vector regression, in: Proc. of the 17th Int. IEEE Conf. on Intelligent Transport Systems (ITSC), 2014, http://dx.doi.org/10.1109/ITSC.2014.6958049.
[106] P. Panja, R. Velasco, M. Pathak, M. Deo, Application of artificial intelligence to forecast hydrocarbon production from shales, Petroleum 4 (2018) 75–89.
[107] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[108] P. Brazdil, C. Giraud-Carrier, Metalearning and algorithm selection: Progress, state of the art and introduction to the special issue, Mach. Learn. 107 (2018) 1–14.

[109] R.B.C. Prudêncio, M.C.P. de Souto, T.B. Ludermir, Selecting machine learning algorithms using the ranking meta-learning approach, in: N. Jankiwski, et al. (Eds.), Meta-Learning in Computational Intelligence, Springer, 2011, pp. 225–243, SCI 358.

[110] K. Eggensperger, M. Lindauer, H.H. Hoos, F. Hutter, K. Leyton-Brown, Efficient benchmarking of algorithm configurators via model-based surrogates, Mach. Learn. 107 (2018) 15–41.

[111] C. Ansótegui, Y. Malitsky, H. Samulowitz, M. Sellmann, K. Tierney, Model-based genetic algorithms for algorithm configuration, in: Proc. of 24th Int. Joint Conf. on Artificial Intelligence (IJCAI 2015), 2015, pp. 733–739.

[112] J. Derrac, S. Garćia, S. Hui, P.N. Suganthan, F. Herrera, Analyzing convergence performance of evolutionary algorithms: A statistical approach, Inf. Sci. 289 (2014) 41–58.

[113] B. Bischl, P. Kerschke, L. Kotthoff, M. Lindauer, Y. Malitsky, A. Fréchette, H. Hoos, F. Hutter, K. Leyton-Brown, K. Tierney, J. Vanschoren, ASLib: A benchmark library for algorithm selection, Artificial Intelligence 237 (2016) 41–58.

[114] F. Hutter, L. Xu, H.H. Hoos, K. Leyton-Brown, Algorithm runtime prediction: Methods & evaluation, Artificial Intelligence 206 (2014) 79–111.

[115] M. Lindauer, F. Hutter, Warm starting of model-based algorithm configuration, in: Proceedings of the 32th AAAI Conference on Artificial Intelligence (AAAI-18), 2018, pp. 1355–1362.

[116] P. Kerschke, H.H. Hoos, F. Neumann, H. Trautmann, Automated algorithm selection: Survey and perspectives, Evol. Comput. 27 (1) (2018) 3–45.

[117] F. Hutter, M. López-Ibáñez, C. Fawcett, T.M. Lindauer, H.H. Hoos, K. Leyton-Brown, T. Stützle, AClib: A benchmark library for algorithm configuration, in: Proc. of the 8th Int. Conf. on Learning and Intelligent Optimization, LNCS 8426, (2014) pp. 36–40.

[118] K.A. Smith-Miles, J.I. van Hemert, Discovery the suitability of optimisation algorithms by learning from evolved instances, Ann. Math. Artif. Intell. 61 (2) (2011) 87–104.

[119] T. Cloete, A.P. Engelbrecht, G. Pampara, Cilib : A collaborative framework for computational intelligence algorithms - part II, in: Proceedings of the IEEE International Joint Conference on Neural Networks (World Congress on Computational Intelligence), 2008, http://dx.doi.org/10.1109/IJCNN.2008.4634037.

[120] XML on W3C website. https://www.w3.org/XML. (Accessed on 9 January 2020).

[121] OWL on W3C website. https://www.w3.org/OWL. (Accessed on 9 January 2020).

[122] H. Boley, S. Tabet, G. Wagner, Design rationale of RuleML: A markup language for semantic web rules, in: Proceedings of SWWS'01, Stanford, 2001.

[123] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Submission, 2004.

[124] Protégé website. http://prot\'{e}g\'{e}.stanford.edu. (Accessed on 10 January 2020).

[125] E. Friedman-Hill, Jess in Action: Rule-Based Systems in Java, Manning Publications, 2012.

[126] DROOLS website, http://www.drools.org (Accessed on 9 January 2020).

[127] A. Galicia, R. Talavera-Llames, A. Troncoso, I. Koprinska, F. Martínez-Álvarez, Multi-step forecasting for big data time series based on ensemble learning, Knowl.-Based Syst. 163 (2019) 830–841.

[128] S.M. Kamalapur, V. Patil, Parameter settings in particle swarm optimization, in: Chapter in Handbook of Research on Soft Computing and Nature-Inspired Algorithms, IGI Global, 2017, pp. 101–132, http://dx.doi.org/10.4018/978-1-5225-2128-0.ch004.

[129] R. Poli, An Analysis of Publications on Particle Swarm Optimization Applications, Technical Report CSM-469, Department of Computer Science, University of Essex, UK, (ISSN: 744-8050) 2007.

[130] B. Sendhoff, M. Roberts, X. Yao, Evolutionary computation benchmarking repository, IEEE Comput. Intell. Mag. 60 (2006) 50–51.

[131] A. Udías, E. Buforn, Principles of Seismology, second ed., Cambridge University Press, 2018.

[132] F. Martínez-Álvarez, J. Reyes, A. Morales-Esteban, C. Rubio-Escudero, Determining the best set of seismicity indicators to predict earthquakes. Two case studies: Chile and the Iberian Peninsula, Knowl.-Based Syst. 50 (2013) 198–210.

[133] A. Köhler, M. Ohrnberger, C. Riggelsen, F. Scherbaum, Unsupervised feature selection for pattern search in seismic time series, in: JMLR: WS and Conf. Proc., vol. 4, 2008, pp. 106–121.

[134] L. Zhang, L. Si, H. Yang, Y. Hu, J. Qiu, Precursory pattern based feature extraction techniques for earthquake prediction, IEEE Access 7 (2019) http://dx.doi.org/10.1109/ACCESS.2019.2902224.

[135] H. Jia, J. Lin, J. Liu, An earthquake fatalities assessment method based on feature importance with deep learning and random forest models, Sustainability 11 (2019) 2727, http://dx.doi.org/10.3390/su11102727.

[136] G.N. Smith, Probability and Statistics in Civil Engineering, Collins, London, 1986.

[137] D. Valero, Fundamentals of Air Pollution, fifth ed., Academic Press, Elsevier, 2014.

[138] L. Mesin, F. Orione, R. Taormina, E. Pasero, A feature selection method for air quality forecasting, in: K. Diamantaras, W. Duch, L.S. Iliadis (Eds.), ICANN 2010, Part III, 2010, pp. 489–494, LNCS 6354.

[139] J.-H. Seo, Y.-H Kim, Genetic feature selection for very short-term heavy rainfall prediction, in: G. Lee, et al. (Eds.), ICHIT 2012, (2012) Springer-Verlag, 2012, pp. 312–322, LNCS 7425.

[140] T. Bartz-Beielstein, Experimental Research in Evolutionary Computation: The New Experimentalism, Springer, New York, 2006.

[141] O. Mersmann, M. Preuss, H. Trautmann, B. Bischl, C. Weihs, Analyzing the BBOB results by means of benchmarking concepts, Evol. Comput. 23 (1) (2015) 161–185.

[142] J.J. Liang, B.Y. Qu, P.N. Suganthan, A.G. Hernandez-Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, Technical Report 201212, CEC 2013, Zhengzhou China, Singapore, 2013.

[143] V. Volz, B. Naujoks, P. Kerschke, T. Tušar, Single and multi-objective game-benchmark for evolutionary algorithms, in: Proc. of the Genetic and Evolutionary Computation Conference GECCO', vol. 19, 2019, pp. 647–655, http://dx.doi.org/10.1145/3321707.3321805.

[144] B.Y. Qu, J.J. Liang, Z.Y. Wang, Q. Chen, P.N. Suganthan, Novel benchmark functions for continuous multimodal optimization with comparative results, Swarm Evol. Comput. 26 (2016) 23–34.

[145] R.R. Andrawis, A.F. Atiya, H. El-Shishiny, Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition, Int. J. Forecast. 27 (2011) 672–688.

[146] R.S. Olson, W. La Cava, P. Orzechowski, R.J. Urbanowicz, J.H. Moore, PMLB: A large benchmark suite for machine learning evaluation and comparison, BioData Min. 10 (2017) 36, http://dx.doi.org/10.1186/s13040-017-0154-4.