



Learning ensembles of deep neural networks for extreme rainfall event detection

Gianluigi Folino¹ · Massimo Guarascio¹ · Francesco Chiaravalloti²

Received: 25 May 2022 / Accepted: 6 January 2023 / Published online: 24 January 2023
© The Author(s) 2023, corrected publication 2023

Abstract

Accurate rainfall estimation is crucial to adequately assess the risk associated with extreme events capable of triggering floods and landslides. Data gathered from Rain Gauges (RGs), sensors devoted to measuring the intensity of the rain at individual points, are commonly used to feed interpolation methods (e.g., the Kriging geostatistical approach) and estimate the precipitation field over an area of interest. However, the information provided by RGs could be insufficient to model complex phenomena, and computationally expensive interpolation methods could not be used in real-time environments. Integrating additional data sources (e.g., radar and geostationary satellites) is an effective solution for improving the quality of the estimate, but it needs to cope with Big Data issues. To overcome all these issues, we propose a Rainfall Estimation Model (REM) based on an Ensemble of Deep Neural Networks (DeepEns-REM) that can automatically fuse heterogeneous data sources. The usage of Residual Blocks in the base models and the adoption of a Snapshot procedure to build the ensemble guarantees a fast convergence and scalability. Experimental results, conducted on a real dataset concerning a southern region in Italy, demonstrate the quality of the proposal in comparison with the Kriging interpolation technique and other machine learning techniques, especially in the case of exceptional rainfall events.

Keywords Rainfall estimation · Residual neural network · Snapshot ensemble · Multi-source heterogeneous data fusion

1 Introduction

Accurately estimating the rainfall rate allows for modelling hydrological and other environmental processes, and enabling timely countermeasures to mitigate the risks due to extreme events, such as flash floods and landslides. Real-time constraints, noise, outliers, incomplete and inconsistent data, and the need of integrating different sources of data, represent some of the main issues to consider in defining a solution able to effectively address this problem.

Rain Gauges (RGs) represent a valuable tool to estimate rainfall in complex environments as these sensors provide direct measurements of the precipitation at individual points in different time windows and permit to gather a large amount and high rate of information used to detect extreme rainfall events. Interpolation methods are usually employed to combine the information provided by different RGs and estimate the rainfall height for spatial coordinates not covered by sensors. In particular, Kriging geostatistical interpolation techniques [1] are widely adopted in the literature to address this problem.

However, since rainfall is a complex phenomenon characterized by a high spatial and temporal variability over a wide range of scales [2], the reconstruction of the space rain field from point measurements represents a challenging task [3]. Moreover, although dense, RG networks could be not able to gather enough information on the spatial variability of the precipitations, therefore as a result the interpolation of the gathered values may not provide a reliable estimate of the rainfall field for real-time warning systems or for post-analysis of critical events. Indeed, the information provided by the rain gauges is

Gianluigi Folino and Massimo Guarascio have contributed equally to this work.

✉ Massimo Guarascio
massimo.guarascio@icar.cnr.it

Gianluigi Folino
gianluigi.folino@icar.cnr.it

Francesco Chiaravalloti
francesco.chiaravalloti@irpi.cnr.it

¹ ICAR-CNR, Via Pietro Bucci 8-9/C, Rende 87036, Italy

² IRPI-CNR, Via C. B. Cavour, 4, Rende 87036, Italy

frequently inadequate to extract relevant patterns for convective storms characterized by strong intensity, high variation and short duration. As a consequence, heavy precipitations are poorly depicted and this scenario is further exacerbated in regions with a complex topography and small catchments [4].

Some recent approaches propose to address this problem by integrating heterogeneous rainfall data sources via interpolation methods [5]. The reconstruction of the spatial precipitation field can be heavily improved by coupling rain gauge measurements with precipitation estimates provided by meteorological radars [6]. Moreover, geostationary satellites (MSG) represent a further important source to estimate the rain field, since they allow for continuously monitoring large areas with problems deriving from the particular orography [7]. Processing and handling all these large amounts of heterogeneous data represents a hard challenge in this context.

A widely used approach for integrating distributed rain gauge point measurements with radar data is the Kriging with External Drift (KED), a multivariate geostatistical interpolation method allowing the processing of non-stationary random field (rainfall data, in the case of interest here) taking into account the spatial dependence of the random variable and also its linear relation to one or more additional variables (radar data) [8], but it is computationally expensive since it is based on *Gaussian Process Regression* [9] and do not take in account historical data that represents a precious source for improving the estimate.

1.1 Related works

Recently, a new line of research focused on the usage of Machine Learning (ML) techniques to automatically fuse information from heterogeneous data sources and to provide a more accurate rainfall estimation [20]. However, the adoption of ML techniques raises a number of issues to be coped with, such as imbalanced data, many missing attributes and real-time constraints, which require the development of specific solutions. In addition, integrating heterogeneous rainfall data sources to yield a more accurate estimation is a further relevant research topic, especially when the hydrological model requires a rainfall estimate as input [5]. Ensemble and Deep Learning (DL) are two data-driven paradigms able to effectively tackle these issues.

Ensembles [21] are well-known ML methods, where the output of several models trained against different data samples or using different algorithms, are then combined according to a given strategy for classifying new unseen instances. This paradigm permits us to cope with the problem of unbalanced classes and reduce the variance and

the bias of the error. In the reference scenario, ensemble-based algorithms can be employed to address the above-cited issues related to the problem of rainfall estimation, such as unbalanced data and non-linear correlation among the different sources of data (radar, Meteosat and rain gauges).

An interesting preliminary ensemble approach (including also a neural network) to improve rainfall rates estimation is proposed in [10], however, it takes into account only radar data. Kuhnlein et al. [22] also exploit an ensemble-based approach that employs Random Forests to estimate rainfall rates from MSG data. A further work moving in this direction has been proposed in [12]. The authors devised a framework integrating different data sources via a Hierarchical Probabilistic Ensemble Classifier (HPEC). Experiments conducted on a real case study prove the importance to combine different data sources and the capability of the proposed ensemble method in alleviating the above-mentioned issues.

A different solution approach relies on the usage of (Deep) Neural Networks. Deep Learning [23] allows for learning accurate Rainfall Estimation Models (*REM*) by directly fusing raw low-level data gathered from heterogeneous data sources. Indeed, a hierarchical scheme is exploited in learning Deep Neural networks: basically, a DNN can be imagined as a stack of several levels, each one composed of a number of base computational units (neurons). Here, the idea is that every level of the DNN can extract high-level features, which can be provided as input for the next level of the network. In this way, the DNN enables the automatic extraction of discriminative data abstractions, therefore these models represent an effective solution to fuse different types of data provided in different formats. A further benefit in adopting the DL paradigm is the possibility to incrementally update the model as soon as new batches of data are made available so to ensure the scalability of the overall learning system.

For the above-cited motivations and also for the high non-linearity of the correlations between sensors data, cloud properties and rainfall estimates, recently, different machine learning (and also deep learning) techniques have been explored [24] [13] to improve the quality of the models for the problem of rainfall estimation.

In the literature, some preliminary approaches based on exploiting the DL paradigm have been proposed to address the rainfall estimation problem [16–18] but these solutions employ a single data source at a time or train a separate model for each source. For instance, with the aim to estimate rainfall rate on a daily timescale, the authors of [16] define an approach based on a NN trained against satellite data (single source). In the same way, the work in [17] designs a suitable NN-Based model to process data from the Meteosat data source.

A different solution is proposed in [13], i.e., deep autoencoders and Long Short Time Memory networks are combined to process image sequences concerning rainfall events. Basically, the approach allows for denoising, extracting high-level features and analyzing the data provided as input to the network.

In [18], the authors define a new methodology to integrate different precipitation data sources by combining an artificial neural network and a vector space transformation function. In the work [19], a Multi-Layer Perceptron is trained by using the surface and altitude mapping data. A Principal Component Analysis (PCA) is exploited to preprocess raw data and extract informative features to feed the DNN. Again, a preliminary approach to solve this task is proposed in [15] in which different data sources are automatically combined for providing a rainfall estimate by using a DNN with a particular topology.

Table 1 summarizes the most significant approaches among those described above. Compared to these approaches, to the best of our knowledge DeepEns-REM is the first solution that merges the Deep Learning paradigm with ensemble-based methods and integrates different types of data sources.

1.2 Our proposal

To overcome the main limitations of the above-described works and address the main challenges of the weather forecast domain (i.e., handling huge amounts of data yielded by the sensors, integrating different types of data from different sources and learning incrementally as soon as new batches of data are available) we defined a scalable DL-based framework able to effectively detect severe rainfall events.

In more detail, we propose to merge the robustness of the ensemble technique with the capability of DNNs to handle and automatically combine raw low-level data from heterogeneous data sources: an Ensemble of Deep Neural Networks (DeepEns-REM) fed with different types of data sources (i.e., RGs, radars and satellites) is introduced here. In order to better face the imbalanced nature of the training data (severe rainfall events are rare cases), the DNN architecture of the base classifiers has been designed to include a combination of *dropout* layers, *batch normalization* functions and *residual-like connections*. Each model composing the ensemble is discovered by exploiting a *Snapshot procedure* [25] and their predictions are combined via a not-trainable function. Notably, while Residual Blocks permit the reduction of the number of epochs for the learning algorithm convergence, the Snapshot

Table 1 Analysis of ML/DL based techniques for rainfall estimation

Approach	Multi-source	Data source(s)	ML/DL based	Model type	Ensemble	Ensemble strategy
Tan et al. [10]	×	Radar data	ML	Support vector machine, Multi-layer perceptron	✓	Hierarchical combination
Kuhnlein et al. [11]	×	Setellite data	ML	Random forest	✓	Averaged prediction
Guarascio et al. [12]	✓	Rain gauge, radar and setellite data	ML	Random forest, Bayesian model	✓	Stacking method
Tang et al. [13]	×	Setellite data	DL	Feed forward neural network	×	–
Tao et al. [14]	×	Setellite data	DL	Stacked denoising autoencoder	×	–
Folino et al. [15]	✓	Rain gauge, radar and satellite data	DL	Custom feed forward neural network with weighted loss	×	–
Grimes et al. [16]	×	Satellite data	DL	Feed forward neural network	×	–
Rivolta et al. [17]	×	Satellite data	DL	Feed forward neural network	×	–
Turlapaty et al. [18]	✓	Several high resolution precipitation datasets	DL	Feed norward neural network with vector space Transformation	×	–
Zhang et al. [19]	×	Surface and altitude mapping data	DL	multi-layer perceptron combined with principal component analysis	×	–
DeepEns-REM (our approach)	✓	Rain gauge, radar and setellite data	DL	Residual neural network	✓	Snapshot ensemble with max score criterion

procedure allows for learning the base models composing the ensemble in a single DNN training execution to ensure efficiency and scalability. Moreover, these classifiers are trained by using an ad-hoc cost-sensitive (imbalance-aware) loss function.

The experimental evaluation has been conducted on a real scenario, described in [12]. Specifically, data concern a peninsular region of Southern Italy (Calabria) and are provided by the Italian Department of Civil Protection. It is an effective test case for its complex orography and strong climatic variability. In addition, the complex orography, the heat flux coming from the Mediterranean Sea and the high reliefs near the warm sea interact strongly and favour the convective instability of this region, which also presents frequent floods and landslides.

The proposed framework overcomes the predictive performances of widely used and recognized methods (i.e., KED) and could be of support in many real contexts, as in the case of an officer of the Department of Civil Protection, which has to analyze the rainfall in specific zones exhibiting risks of landslides or floods and has to make real-time decisions in order to avoid damages, or in the worst case, injuries and deaths.

1.3 Organization of the paper

The rest of this article is organized as follows. Sect. 2 provides some background information about the problem of rainfall estimation normally defines the addressed problem. In Sect. 3 we introduce the reference framework. Section 4 is devoted to illustrating the Rainfall Estimation Model devised to tackle the task. Experimental results both in terms of effectiveness and efficiency of the proposed approach, conducted on the above-mentioned real test case, are shown in Sect. 5. Finally, Sect. 6 concludes the work by summarizing the obtained results and introducing some new research lines for future works.

2 Background and problem definition

In this work, we address the areal rainfall estimation problem, which can be defined as the problem to estimate a highly variable spatial field, both in space and time, by using data from a number of randomly placed points.

Direct measurements of rainfall are gathered by RG networks, which provide the amount of precipitation of a limited set of points over the area of interest. Indeed, even dense networks of rain gauges may result too sparse to compute an accurate interpolated spatial rainfall field for hydrological modelling. Estimates of a spatial precipitation field can be provided by other meteorological instruments, like weather radars, which provide high-resolution

observations of precipitation spatial patterns. However, a radar measurement is an indirect estimate of rainfall, as it does not guarantee a precise quantitative measure of rainfall.

In addition, images from meteorological satellites represent a precious source to compute spatially distributed rainfall estimation [26]. In fact, from the observations taken from the satellites with different wavelengths, rainfall estimation methods can extract interesting characteristics of the clouds concerning possible ground precipitations and therefore this aspect can be used to indirectly measure the rainfall.

Notably, since the integration of the different temporal and spatial resolutions of the data sources is crucial to learning accurate estimation models, the preprocessing phase described [12] is adopted to make comparable the measurements logged by the different instruments.

Problem definition

As highlighted above, our final aim consists in discovering accurate predictive models able to estimate the accumulated rainfall raining at individual points (not monitored from the rain gauges) in a specific time interval by automatically fusing further information gathered from different heterogeneous data sources.

Basically, as shown in Fig. 1, each RG belonging to the network provides the measurement of the accumulated rainfall in the point (x, y) where it has been deployed. Anyway, it can monitor only a limited part of the area where we want to estimate the rainfall rate, therefore our task consists in estimating the values of the remaining uncovered points of the grid for a specific time window.

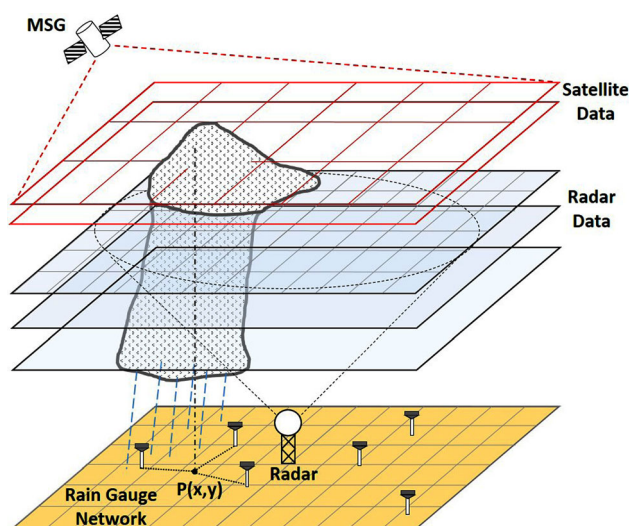


Fig. 1 View of the integrated rainfall data sources for a generic point $p = (x, y)$. Three sources (RG network, Radar and Geostationary satellites) contribute to feeding the Rainfall Estimation Model

Formally, the above-described problem can be defined as follow.

Let be $p_i = (x_i, y_i)$ the spatial coordinates of the i_{th} rain gauge belonging to the RG network composed of m nodes, and let be $f_{rg}(p_i, t) = v$ a function providing the observed height of rainfall by the i_{th} rain gauge at time t . We denote with $RG(t) = \langle (p_1, f_{rg}(p_1, t)), (p_2, f_{rg}(p_2, t)), \dots, (p_m, f_{rg}(p_m, t)) \rangle$ the list of pair containing RG spatial coordinates and rainfall observed at time t for each rain gauge belonging to the network. Notably, in our setting, we assume that the spatial coordinates of the rain gauges cannot change over time.

In addition, different types of side information, extracted from other data sources, can be coupled with the RG data to enrich the initial set of features. With this aim, let us consider a generic point p belonging to the area where we want to estimate the accumulated rainfall and not be covered by any rain gauge.

Let be $f_{radar}(p, t) = \langle rf_1, rf_2, \dots, rf_k \rangle$, where rf_i is the value of the i_{th} feature gathered by the radar in p at time t .

Let be $f_{msg}(p, t) = \langle mf_1, mf_2, \dots, mf_l \rangle$, where mf_i is the value of the i_{th} feature provided by the meteosat satellites in p at time t .

We define the function $f_d(p, j)$, which, given a point p , returns its distance from the nearest j_{th} rain gauges in $RG(t)$. We define the function $f_n(p, j)$, which, given a point p , returns the index of the nearest j_{th} rain gauge in $RG(t)$.

Afterwards, we define the function $f_c(f_{rg}(p_i, t)) = c$ as a function $\mathbb{R}_{\geq 0} \mapsto C = \{c_1, c_2, \dots, c_o\}$ which maps each value of height of rainfall with a specific (discrete) rainfall class c .

Let be

$$f_{rg'}(p, t, nb) = \langle (p_{f_n(p,1)}, f_d(p, f_n(p, 1)), f_c(f_{rg}(p_{f_n(p,1)}, t))), (p_{f_n(p,2)}, f_d(p, f_n(p, 2)), f_c(f_{rg}(p_{f_n(p,2)}, t))), \dots, (p_{f_n(p,nb)}, f_d(p, f_n(p, l)), f_c(f_{rg}(p_{f_n(p,nb)}, t))) \rangle$$

where nb is the number of neighbours considered.

We define as extended data input (*edi*) for a point p_i at time t the following: $edi(p, t, nb) = f_{rg'} \oplus f_{radar} \oplus f_{msg}$.

Finally, we define a *Rainfall Estimation Model REM* as a function $REM : Q^{nb} \times \mathbb{R}_{\geq 0}^k \times \mathbb{R}_{\geq 0}^l \rightarrow C$ where $Q = \mathbb{R}_{\geq 0}^2 \times \mathbb{R}_{\geq 0} \times C$ that maps each tuple $\tau \equiv edi(p_i, t, nb)$ with a rainfall class i.e. $\tau \mapsto REM(\tau)$ where $REM(\tau) \in C$.

In a nutshell, given as input the concatenation of the data measured by the three data sources (rain gauges, radars and Meteosat satellites) at a given time t , this model (REM) returns the estimated class c of the rainfall event.

3 Reference framework

In this work, we extend the framework defined in [12] by including a novel REM based on an Ensemble of Deep Neural Networks. The software architecture of this framework is divided into different layers, which are used for data integration and preprocessing, for building the DNN-based model and finally for evaluating the performance and visualizing the results.

Figure 2 shows the above-mentioned layers, described in the following, starting from the lowest layer comprising the different information sources (i.e., rain gauges, radar and satellite data).

First, the *Information Retrieval* layer permits the extraction and the integration of the different types of

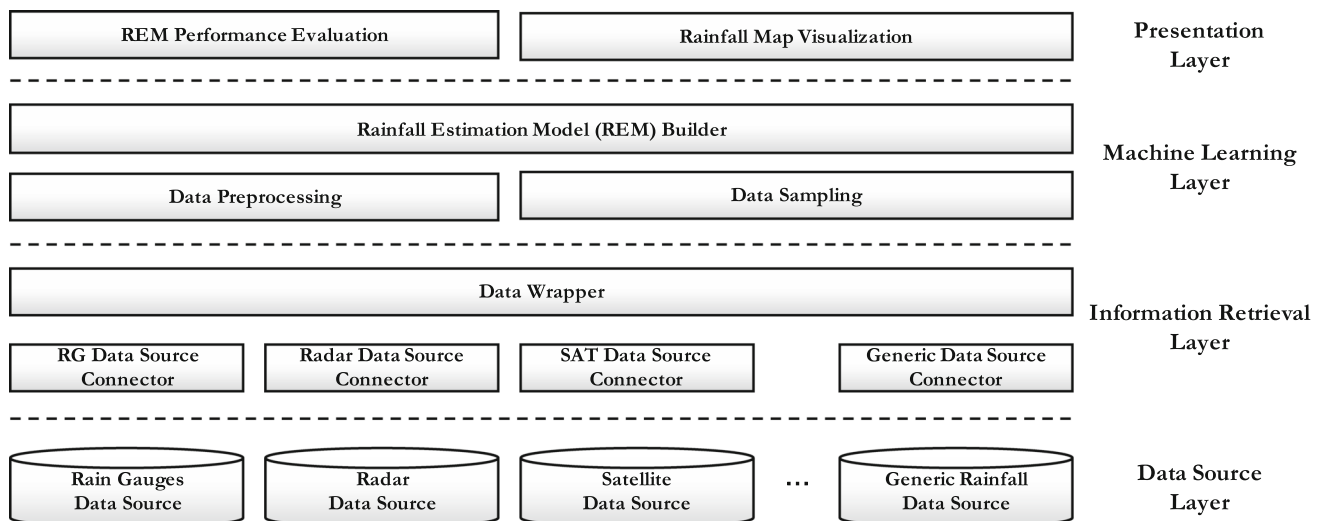


Fig. 2 Rainfall estimation framework system

information, extracted from the data sources. The connection with the different sources of data is guaranteed from the *Data Source Connectors*. Then, their output is sent to the *Data Wrapper*, which is responsible for supplying a unified view of these data to the upper level.

Afterward, data are passed to the *Machine Learning* layer, in which they are preprocessed to make them suitable for the analysis and an undersampling strategy is adopted to address the class unbalanced problem. In more detail, this layer includes three main modules: *Data Pre-processing*, *Data Sampling and Model Building*. First, missing values, outliers, and noise are handled by adopting different strategies to make data clean for the learning phase. Indeed, a big issue in handling the different distribution of the classes is represented by the so-named unbalancing problem, in which the majority class is often one or two orders of magnitude greater than the minority class. Consequently, the minority class is present only in a small portion of the training set, making really hard to train the classification models. In our framework, an ad-hoc undersampling strategy is used to mitigate this problem. Then, a model used to estimate the rainfall is generated by the *REM* builder model, trained from the data preprocessed with the other modules. Finally, test data, opportunely preprocessed similarly to the training data, and the *REM* model are passed to the last layer, in which the visualization of the final estimation of the rainfall, together with an

evaluation of the performance of the model, is generated by using respectively the Rainfall Map Visualization and the REM Performance Evaluation module.

4 DeepEns-REM

In this section, we describe the classification model adopted to distinguish different rainfall classes. Specifically, our *REM* takes the form of an Ensemble of Residual Neural Networks learned by exploiting a Snapshot procedure. Below, first, we describe the architecture of the base learner, and then we illustrate the Snapshot procedure adopted for building the ensemble model.

4.1 ResNet-REM base model

In Fig. 3a, the overall architecture of the DNNs composing the ensemble is shown. Basically, it is a *feed-forward neural network* including a number of *skip connections*.

The usage of the skip connections induces in the base DNN classifier a behaviour similar to *Residual Networks* [27], which have been proven to be effective solutions to the well-known *degradation problem* (i.e., neural networks performing worse at increasing depth), and capable of ensuring a good trade-off between convergence rapidity and expressivity/accuracy.

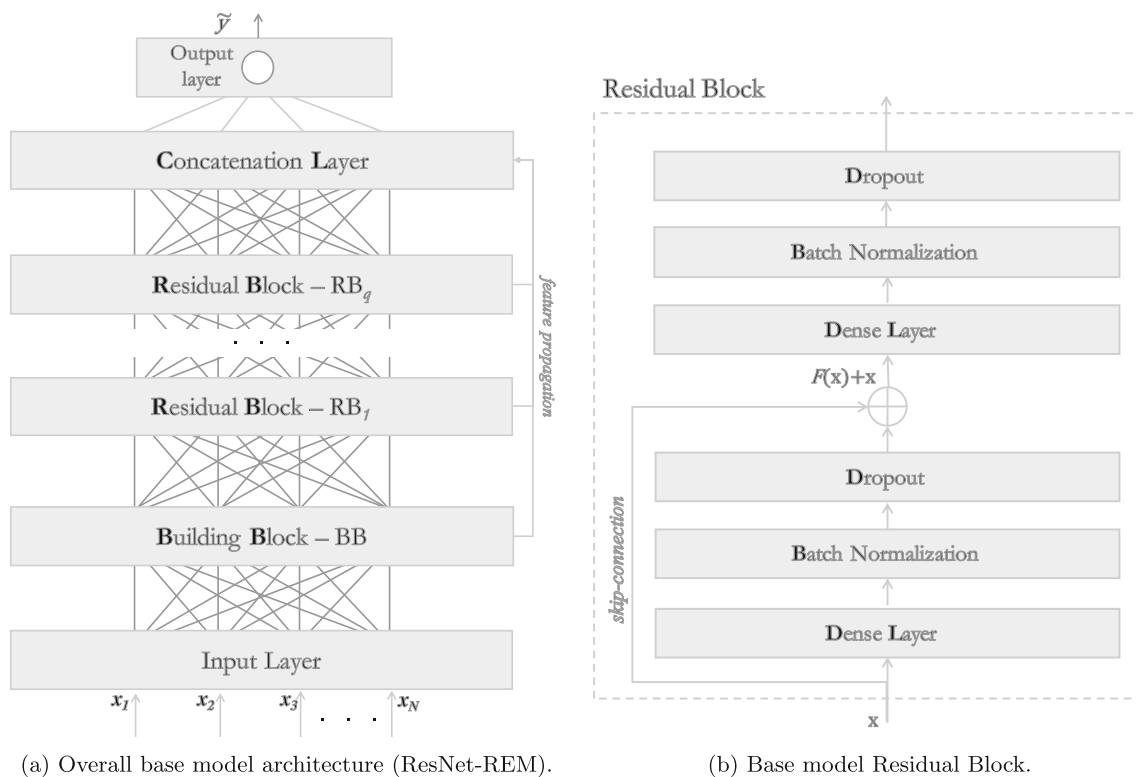


Fig. 3 DeepEns-REM base components

Notably, these benefits seem to depend on the fact that a residual network can be figured out as an ensemble of alternative transformation paths: in practice, the final output of the network is yielded by combining paths with different lengths; therefore, a ResNet could be regarded as an ensemble itself.

The base learner architecture is based on the sub-net shown in Fig. 3b including three main components: (i) a fully-connected dense layer equipped with a (ReLU) activation function [28], (ii) a batch-normalization layer that enables more stability in the learning stage [29], (iii) a final dropout layer to address the overfitting issue [30]. Notably, in Fig. 3a, we refer with *Building Block* the stack of these three components. Interestingly, as highlighted by Hinton et al. in [31], dropout mechanisms enable within the neural net an ensemble behaviour: multiple sub-nets resulting from randomly masking some neurons are combined to yield the final output.

A Residual Block (including two Building Blocks stacked with a skip connection) can be replicated several times within the model to make the architecture more depth and the number of instances of this sub-net represents a parameter of the ResNet-REM.

We highlight that we are not addressing a classification problem but an ordinal learning task. In a nutshell, our main problem consists in ranking the rainfall events on the basis of their risk and identifying the most dangerous ones. Anyway, it would be better whether the errors of the model in estimating the severity of a rainfall event correspond to adjacent classes (i.e. a slightly heavier or lighter rainfall event). On the basis of this rationale, the last layer of our architecture is equipped with a single neuron that yields a prediction value ranging in the interval $[0, \text{number_of_classes}]$.

In the literature, Mean Absolute Error (MAE) [32] has been often adopted as loss function for ranking problems due to its robustness to outliers. However, in our scenario REM should focus on detecting exceptional behaviours by tackling the unbalance problem and, hence, reducing the classification error for the minority classes (*exceptional cases*). In this respect, to further improve the detection capabilities of the network we adopted a weighted variant of the MAE (named in the following Weighted-MAE), defined as follows:

$$\text{Weighted - MAE} = \frac{1}{n} \sum_{i=1}^n \|y^{(i)} - \hat{y}^{(i)}\| \cdot \text{weight}(y^{(i)}) \quad (1)$$

where n represents the batch size i.e. the number of examples, $y^{(i)}$ and $\hat{y}^{(i)}$ are the real and the predicted value for the rainfall event respectively, and $\text{weight}(y^{(i)})$ is a

function that maps each class with a correspondent weight. In our framework, we adopt a linear weight function, i.e. instances belonging to the first class are weighted with 1, instances belonging to the second class are weighted with 2, and so on. Finally, the output of the Residual Blocks and the Building Block is concatenated and provided as input for the last layer of the model to exploit different levels of abstraction to yield the final decision. In the following, we refer to the base model as ResNet-REM.

4.2 Snapshot procedure

In our solution, we train several base models, whose output is then combined by using a not-trainable function (the max rainfall class, i.e., the most severe class event among the predicted ones by the base models). Each ResNet-REM composing the ensemble is yielded by means of a straightforward algorithm, which exploits the iterative gradient-descent optimization method of the DNN training stage. The main idea consists in updating the weights of the base DNN architecture, at each epoch, in opposite direction to the gradient, with the aim of progressively approaching a local minimum. In this scheme, the learning rate (lr) acts the role to control the convergence rate and it is iteratively updated to ensure fast convergence. Notably, both the local optimum achieved by means of the SGD procedure and its variants, depend on the initialization of the network and the lr value. As highlighted in [25], a cyclic reset of the lr yields a re-initializing of the network and a restarting of the optimization from another point in the search space and allows for discovering different variants of the model. In our framework, we adopted as a cyclic function, the shifted cosine function proposed by Loshchilov and Hutter [33].

Specifically, the DeepEns-REM is learned as follows. A ResNet-REM (named in the following as RN_1) is randomly initialized and trained. Hence, the algorithm iteratively learns the model RN_i by re-training RN_{i-1} with the initial learning rate η for a fixed number of epochs. At each epoch, η is progressively lowered. The trained RN_i is added to the ensemble list and lr is once again reinitialized. The defined Ensemble Strategy is presented in Fig. 4, where several ResNet-REMs $\{RN_1, RN_2, \dots, RN_n\}$ are suitable combined to compute the final prediction. In the application phase, the *edi* of each point to be estimated is provided as input to each base model composing the ensemble. The final rainfall severity score (i.e., the rain event class) is obtained by considering the max class value estimated by the base model i.e., $\max(C_{RN_1}, \dots, C_{RN_n})$ where n is the number of REMs trained.

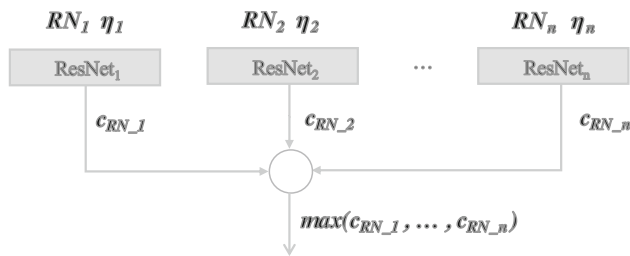


Fig. 4 Snapshot-based ensemble model

5 Evaluation

To evaluate the quality of our approach to estimating (extreme) rainfall events, a number of experiments on the real data concerning Calabria have been conducted. The first suite of experiments aims to compare our approach with some ensemble-based techniques (random forest, boosting) and, in addition, with the decision tree algorithm and the SVR (support vector regression model), which has been successfully used in the field of rainfall forecasting [34]. Furthermore, we compared DeepEns – REM with the KED (kriging with external drift) baseline method and with the Hierarchical Probabilistic Ensemble Classifier (HPEC)[12]. Note that KED represents the state-of-the-art among interpolation methods for rainfall estimation. Finally, we compared our deep ensemble model with two state-of-the-art neural architectures proposed in the literature.

In particular, all these test suites have been validated also on the minority class, representing extreme rainfall events.

5.1 Dataset, parameters and evaluation measures

The dataset used for the experiments measures the rainfall events for a period concerning the second half of 2016 in Calabria and it has 117, 600 tuples and 35 features. The pre-processing phase of this dataset is described in the previous subsection and in detail in [4]. Synthetically, for every cell, for a 30 minutes time step, 35 features were extracted, 11 features from the MSG satellites, 16 features from the rain gauges and 5 features from the radars. More in detail, for each point, we consider the 4 nearest cells in which a rain gauge is present (latitude, longitude, distance, and rainfall value measured for each of them). In addition, also the month of the measure is considered.

The class to be predicted, i.e., the *mm* of rain fallen in 30 minutes, was discretized into 5 classes, (i.e., less than 0.5, 0.5–2.5, 2.5–7.5, 7.5–15 and greater than 15) [12]. The last class is particularly important because it includes extreme rainfall events, which must be handled adequately.

However, as a low number of tuples belong to this class, its classification is a really challenging problem.

Specifically, the base learner includes two instances of Residual Block. Each dense layer is equipped with a *ReLU* activation function and includes 64 neurons. RMSprop is used as optimizer with an initial $lr = 0.001$. DeepEns-REM is trained over 64 epochs and $batch_size = 512$. Finally, the Snapshot procedure is performed until 5 variants of the model RN_1 have been obtained.

As for the implementation of the competitors, we adopted the well-known scikit-learn machine learning implementation¹ for the boosting, random forest and the SVR model. No tuning of the parameters was conducted. The random forest algorithm fixes a maximum depth for each base learner equals to 7, and an overall number of trees equal to 50. Standard parameters have been used for the other algorithms and parameters. Kriging with External Drift (the baseline method) is based on the “autokrige” function of the Automap library of the statistical software R², which automatically fits and chooses the best model among the different variogram models for the spatial interpolation.

Each model is trained against a sample of 66% of the dataset whereas it is evaluated on the remaining examples (test set). All the experiments were averaged over 30 runs. Note that, for a fair comparison with the kriging baseline method, we adopted the same above-defined partition for the training and test set. In addition, due to the nature of the interpolation method of the kriging, first, the points belonging to the test dataset were removed and the other points (training set) were used to train the autocorrelation models.

Due to the rarity of the minority class (exceptional rainfall events), and in order to avoid issues concerning the correctness of the evaluation of this class, we adopted measures particularly apt to cope with unbalanced problems. Therefore, in addition to the metrics of *precision* and *recall*, usually used in the balanced case, we employ other metrics specific for the rainfall estimation [36], i.e., the False-Alarm Ratio (FAR), the Critical Success Index (CSI) and the Probability of Detection (POD). However, these metrics are defined for a two-class scenario; in a multi-class scenario, they can be extended in many ways [37], mainly using two different techniques for averaging these evaluation measures: macro-averaging and micro-averaging. By using the macro-based technique, first, for each class, the performance measure is computed and then it is averaged; for the other case (micro), first, a cumulative sum of the counts (TPs, FPs, FNs, TNs) is calculated and then the overall measure is computed. We adopted macro-

¹ <http://scikit-learn.org/stable/>.

² <https://cran.r-project.org/web/packages/automap/index.html>

averaging, which weights all the classes equally (micro is more biased towards larger classes), as we are mainly interested to predict the minority class.

5.2 Convergence analysis

In this section, we analyze the model behaviour at learning time. Basically, we are interested to investigate whether DeepEns-REM architecture enables a faster convergence of the algorithm to reduce the number of epochs required to learn an accurate predictive model. Basically, we compared two architectures (respectively named in figure, *ResNet-REM* and *Base DNN*), that include the same number of building blocks (i.e., a block composed of a feed-forward dense layer, a batch normalization layer and a dropout layer). The only difference between the architecture relies on the presence of the skip connections and residual blocks in the former one.

In Fig. 5a, the training loss value per epoch is shown. First, we pinpoint that, at the first iteration, the ResNet-REM loss value is lower than the Base DNN one of 0.1; therefore, we can observe the benefits of adopting the skip connections already in the early stages of the training phase. Moreover, ~ 38 epochs are required in order for the two models to exhibit the same values of loss. This behaviour is even more evident in Fig. 5b, where we monitored the loss values in the validation set. In particular, we can observe an unstable behaviour for the loss in the Base DNN, whereas ResNet-REM is able to achieve a stable value of loss in a limited number of iterations. In the Fig. 6a b, we also report the Macro-F1 values for both architectures. Once again, as regards the values for the training set, Base DNN is not able to achieve the same performance in 40 epochs, whereas, on the validation set, we observe again an unstable behaviour for this algorithm.

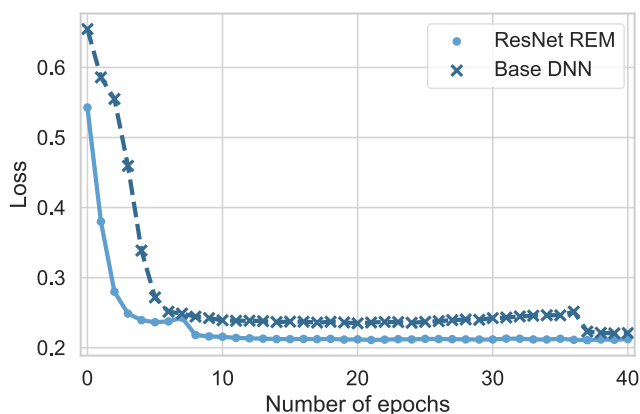
5.3 Performance analysis

Here, we evaluate the rainfall estimation capability of DeepEns-REM in comparison with other Machine Learning-based techniques, in particular ensemble methods (including the Hierarchical Probabilistic Ensemble Classifier proposed in [12]) and with the baseline method (kriging with external drift). Mainly, we are interested in analyzing the behaviour of our algorithm in coping with exceptional rainfall events (minority class).

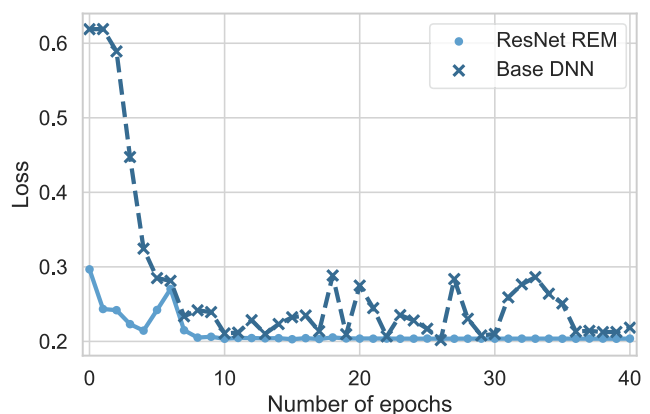
In order to evaluate whether the differences between the performance of the models are significantly different, a Wilcoxon signed-ranked test with a confidence level of 0.95 ($\alpha = 0.05$) was used. The Wilcoxon test is a non-parametric pairwise test that can be used to detect significant differences between two algorithms. It is analogous to the paired t test, however, it is largely used in the literature as it does not assume normal distributions and the outliers (exceptionally good/bad performance) have less effect on it.

In the first suite of experiments, we compare DeepEns-REM with other ML-Based methods: in Table 2, we report CSI (higher is better), FAR (lower is better) and POD (higher is better) macro-averaged for all the rainfall classes, while Table 3 shows precision, recall and F-measure for the class including extreme events. The values, which are significantly better, are reported in bold while the values that are significantly worst are reported in italic face.

As reported in 2, DeepEns-REM exhibits significantly better rainfall detection capabilities since it obtains the best value in terms of POD. Moreover, as shown in Table 3, it represents a good trade-off between precision and recall in detecting extreme events with the best value in terms of F-Measure.



(a) Loss behaviour on the training set.



(b) Loss behaviour on the validation set.

Fig. 5 Convergence analysis for ResNet-REM and Base DNN (Weighted-loss)

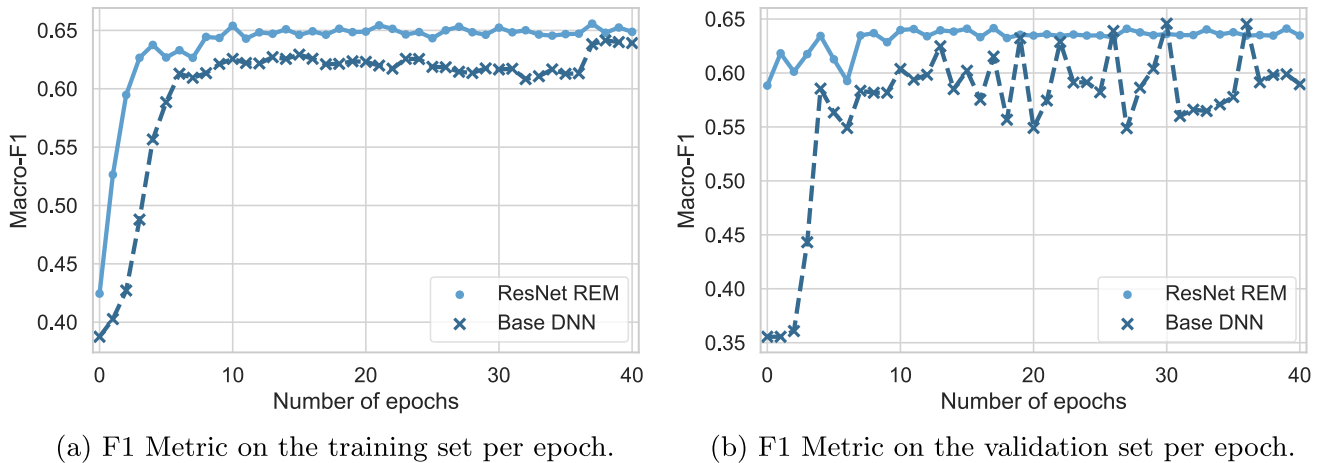


Fig. 6 Convergence analysis for ResNet-REM and Base DNN (Macro F1-score)

Table 2 CSI, FAR, POD, and MSE for the SVR, Decision tree, boosting, random forest and DeepEns-REM. The values in bold (italic face) are significantly better (worse) than at least other two methods

Algorithm	CSI	FAR	POD	MSE
SVR	<i>0.37 ± 0.010</i>	0.40 ± 0.027	<i>0.43 ± 0.011</i>	0.11 ± 0.002
Decision Tree	0.41 ± 0.009	0.38 ± 0.033	0.47 ± 0.011	0.10 ± 0.002
Boosting	0.43 ± 0.008	0.33 ± 0.026	0.49 ± 0.008	0.09 ± 0.002
Random Forest	0.43 ± 0.010	0.31 ± 0.024	0.49 ± 0.011	0.09 ± 0.002
DeepEns-REM	0.45 ± 0.009	0.38 ± 0.015	0.53 ± 0.012	0.09 ± 0.002

Table 3 Precision, recall and F-measure for the SVR, decision tree, boosting, random forest and DeepEns-REM for the minority class (extreme rainfall events). The values in bold (italic face) are significantly better (worse) than at least other two methods

Algorithm	Precision	Recall	F-measure
SVR	0.54 ± 0.14	<i>0.11 ± 0.05</i>	<i>0.18 ± 0.08</i>
Decision tree	<i>0.50 ± 0.14</i>	0.18 ± 0.07	0.26 ± 0.08
Boosting	0.69 ± 0.11	0.24 ± 0.04	0.35 ± 0.05
Random forest	0.68 ± 0.10	0.23 ± 0.05	0.35 ± 0.06
DeepEns-REM	0.53 ± 0.07	0.31 ± 0.06	0.40 ± 0.06

Table 5 Precision, recall and F-measure for the Kriging, HPEC and DeepEns-REM for the minority classes (extreme rainfall events). The values in bold (italic face) are significantly better (worse) than the Kriging method

Algorithm	Precision	Recall	F-measure
Kriging	0.45 ± 0.08	0.32 ± 0.06	0.38 ± 0.06
HPEC	<i>0.33 ± 0.06</i>	0.40 ± 0.07	0.36 ± 0.06
DeepEns-REM	0.53 ± 0.07	0.31 ± 0.06	0.40 ± 0.06

Once again, Table 4 reports CSI (higher is better), FAR (lower is better), POD (higher is better) for all the classes, while Table 5 reports precision, recall and F-measure for

the minority class, as in the previous experiments, but in this case, we compare DeepEns-REM, HPEC and RF with the baseline, Kriging.

We can observe that ML-Based methods outperform the Kriging in both the tasks i.e., accurately depicting rainfall

Table 4 CSI, FAR, POD and MSE for Kriging, HPEC and DeepEns-REM. The values in bold (italic face) are significantly better (worse) than the Kriging method

Algorithm	CSI	FAR	POD	MSE
Kriging	0.40 ± 0.010	0.39 ± 0.018	0.48 ± 0.013	0.15 ± 0.002
HPEC	0.44 ± 0.011	<i>0.46 ± 0.016</i>	0.58 ± 0.016	0.11 ± 0.002
DeepEns-REM	0.45 ± 0.009	0.38 ± 0.015	0.53 ± 0.012	0.09 ± 0.002

field and detecting extreme events. In particular, also in this case, DeepEns-REM is significantly better in terms of F-Measure for the minority class.

Finally, in Fig. 7, we show the rainfall field respectively reconstructed by DeepEns-REM, HPEC and KED for a particularly severe event that occurred on November 6, 2017, in which three RGs observed their max values in the same year (i.e., 42.4 mm for hour). Notably, we discretized the values computed by KED technique on the basis of the same classes considered for learning the ML-Based models. Specifically, DeepEns-REM exhibits an interesting behaviour by highlighting more events belonging to the class V and yielding a more detailed map for these cases, however, further investigations should be conducted to draw a final conclusion.

5.4 Comparison with deep learning architectures

In this subsection, we want to compare our deep ensemble-based solution with other neural models adopted in the literature. As highlighted in Sect. 2, current literature mainly focuses on the usage of feed-forward neural networks (combined with different types of pre-processing procedures) to address our same task. In this respect, to perform a fair comparison we considered two advanced feed forward neural networks having a number of parameters (i.e., the neural network weights) comparable with our base model. The former (hereinafter named *DNN_BB*) is a deep neural network obtained by stacking several *building blocks* composed of three layers: a Dense Layer, a Batch Normalization Layer and a Dropout Layer. In our experimentation, we tested a network including six building blocks. The latter neural model is a residual net (hereinafter name *DNN_RES*) composed of three residual blocks. To make the comparison fair, both the competitors have the same complexity (i.e., they have the same number of

neurons, optimizer, activation functions, they are trained with the same parameters, etc. of our model). In Tables 6 and 7, we reported the results of our analysis: basically, our model exhibits better performances compared with its competitors, obtaining the best results in terms of CSI, POD and MSE, maintaining the same level of FAR. In particular, if we focus the analysis on the detection of extreme rainfall events, DeepEns-REM represents the best trade-off between precision and recall; indeed, it obtains the best results in terms of recall by limiting the loss in terms of precision.

5.5 Analysis of statistical significance

In order to assess, the significance of the differences observed between our approach and the different methods considered in the experiments presented in the previous subsections, the popular *Friedman-Nemenyi* statistical procedure [38, 39], a widely used in the evaluation of classifiers, was adopted. This procedure essentially consists of two phases: first, the Friedman test is employed to possibly reject the null hypothesis H_0 that the different populations of results (produced each by a distinguished classifier-induction method) have the same mean, so that they can be considered as statistically different ($\alpha = 0.05$). Then, if H_0 is rejected, a post-hoc Nemenyi test with a significance level of 0.05 is used (as post-hoc test) to detect all the pairs of methods that are significantly different from one another: if a pair of methods is assigned a p -value under 0.05, these methods are eventually deemed as significantly different from a statistical viewpoint.

This statistical study includes both the machine-learning based methods (i.e, SVR, Decision Tree, Boosting and Random Forest) and the specific-domain approach (i.e., DNN_REM, Kriging and HPEC). Among the different metrics, considered in our experiments, we show only the

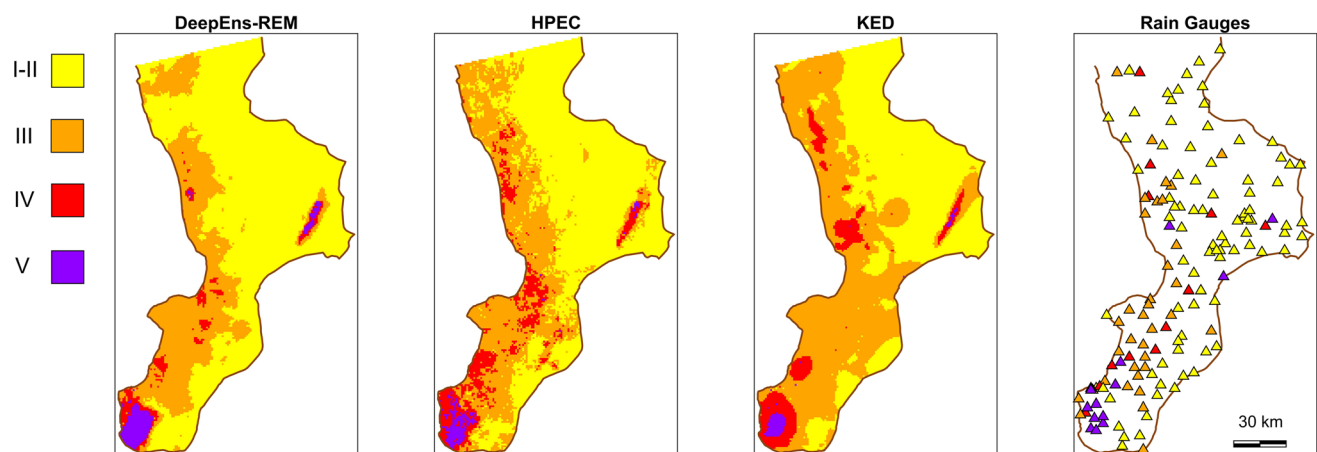


Fig. 7 A qualitative example of how DeepEns-REM, HPEC [12] and KED reconstruct the rainfall map, together with the map of the real values obtained in the rain gauges

Table 6 CSI, FAR, POD, and MSE for DNN_BB, DNN_RES and DeepEns-REM. The values in bold (italic face) are significantly better (worse) than at least other two methods

Algorithm	CSI	FAR	POD	MSE
DNN_BB	0.43 ± 0.11	0.38 ± 0.035	0.51 ± 0.012	0.10 ± 0.002
DNN_RES	0.43 ± 0.12	0.38 ± 0.039	0.50 ± 0.019	0.10 ± 0.002
DeepEns-REM	0.45 ± 0.009	0.38 ± 0.015	0.53 ± 0.012	0.09 ± 0.002

Table 7 Precision, recall and F-measure for for DNN_BB, DNN_RES and DeepEns-REM for the minority class (extreme rainfall events). The values in bold (italic face) are significantly better (worse) than at least other two methods

Algorithm	Precision	Recall	F-measure
DNN_BB	0.50 ± 0.15	0.26 ± 0.07	0.33 ± 0.05
DNN_RES	0.52 ± 0.17	0.24 ± 0.09	0.30 ± 0.08
DeepEns-REM	0.53 ± 0.07	0.31 ± 0.06	0.40 ± 0.06

most relevant for our problem: POD and MSE for all the classes and the F-Measure in the case of the extreme rainfall events.

Figures 8, 9 and 10 show the *critical difference* (CD) diagram, respectively for the metrics of POD, MSE and F-measure (for the case of extreme rainfall events), obtained by using the Friedman-Nemenyi procedure described above. In this diagram, two methods are connected through a horizontal line iff they resulted not significantly different from one another.

From the Fig. 8, all the methods are significantly different (with the exception of Random Forest and Boosting) and only the HPEC technique obtains better performance than our method, which outperforms all the others.

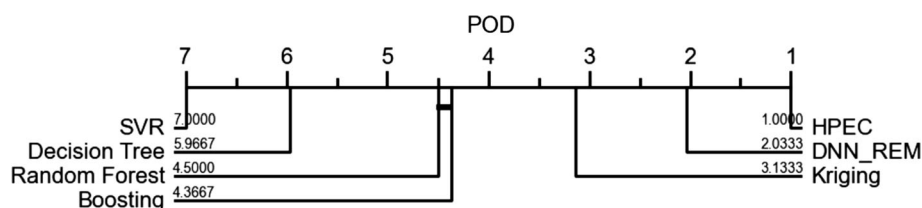
In terms of MSE and F-Measure, our approach (DNN_REM) overcomes the other methods. However, in the case of MSE, it is not significantly better than Random Forest, while for the F-Measure, it is not significantly better than the Kriging technique.

6 Conclusions and future works

A novel Machine Learning-based model for spatial field estimation has been proposed in this work. The main idea consists in merging Ensemble and Deep Learning techniques for estimating rainfall intensity in areas not covered by rain gauge sensors. Specifically, a Rainfall Estimation Model (*REM*) is fed with heterogeneous data from different sources (i.e., Rain gauge network, Radar and Geostationary satellites) and allows for automatically fusing these data so to yield more accurate estimates. In our setting, our *REM* takes the form of an Ensemble of Deep Neural Networks built by using a Snapshot procedure and integrating Residual sub-nets to guarantee a fast convergence and scalability and make the approach apt to efficiently cope with large real datasets.

An experimental evaluation, conducted on a real application scenario, concerning a peninsular region of Southern Italy (Calabria), demonstrates the effectiveness of our proposal in recognizing severe events in comparison with state-of-the-art approaches used in the literature to solve this task. In more detail, our approach exhibits significantly better rainfall detection capabilities both in comparison with other ML-based techniques and with a state-of-the-art technique in the domain of rainfall estimation. In addition, as for the task of detecting extreme rainfall events, the obtained estimation is a good trade-off in terms of the metrics of precision and recall.

In future work, we plan to investigate spatio-temporal correlations by adopting more sophisticated Deep Learning architectures keeping into account these correlations and exploring the accuracy of the method in detecting specific extreme rainfall events.

**Fig. 8** Critical difference (CD) diagram for POD metric (Friedman test + Nemenyi test, $\alpha = 0.05$). All (and only) the pairs of methods resulting not significantly different according to the test (i.e., receiving a p value ≥ 0.05) are connected through a horizontal line

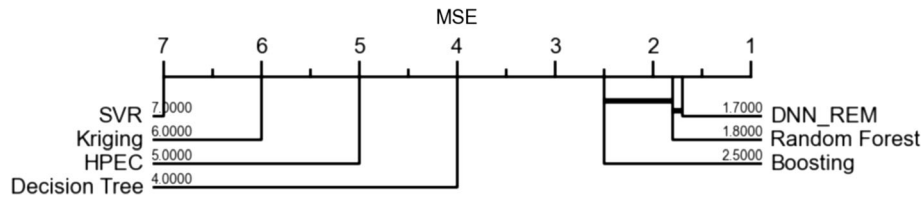


Fig. 9 Critical difference (CD) diagram for MSE metric (Friedman test + Nemenyi test, $\alpha = 0.05$). All (and only) the pairs of methods resulting not significantly different according to the test (i.e., receiving a p value ≥ 0.05) are connected through a horizontal line

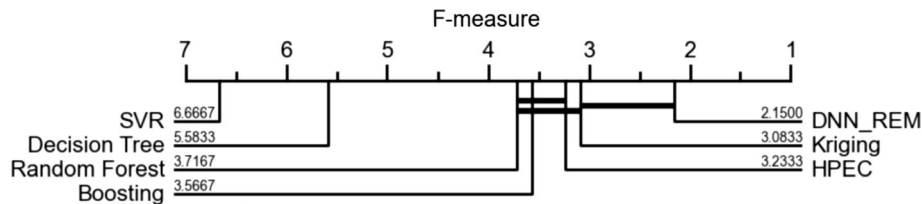


Fig. 10 Critical difference (CD) diagram for F-measure metrics concerning the extreme rainfall events (Friedman test + Nemenyi test, $\alpha = 0.05$). All (and only) the pairs of methods resulting not

significantly different according to the test (i.e., receiving a p value ≥ 0.05) are connected through a horizontal line

Funding Open access funding provided by Consiglio Nazionale Delle Ricerche (CNR) within the CRUI-CARE Agreement.

Data availability The data that support the findings of this study are available from Italian Department of Civil Protection but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Data are, however, available from the authors upon reasonable request and with permission of Italian Department of Civil Protection.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Ly S, Charles C, Degré A (2013) Different methods for spatial interpolation of rainfall data for operational hydrology and hydrological modeling at watershed scale. A review. *Biotechnol Agron Société et Environ* 17(2):392
- Krajewski WF, Ciach GJ, Habib E (2003) An analysis of small-scale rainfall variability in different climatic regimes. *Hydrol Sci J* 48(2):151–162. <https://doi.org/10.1623/hysj.48.2.151.44694>
- Pechlivanidis IG, McIntyre N, Wheater HS (2017) The significance of spatial variability of rainfall on simulated runoff: an evaluation based on the upper lee catchment, UK. *Hydrol Res* 48(4):1118–1130. <https://doi.org/10.2166/nh.2016.038>
- Gabriele S, Chiaravalloti F, Procopio A (2017) Radar-rain-gauge rainfall estimation for hydrological applications in small catchments. *Adv Geosci* 44:61. <https://doi.org/10.5194/adgeo-44-61-2017>
- McKee JL, Binns AD (2016) A review of gauge-radar merging methods for quantitative precipitation estimation in hydrology. *Can Water Resour J/Rev Can des Ressour Hydr* 41(1–2):186–203
- Sokol Z, Szturc J, Orellana-Alvear J, Popová J, Jurczyk A, Célleri R (2021) The role of weather radar in rainfall estimation and its application in meteorological and hydrological modelling—a review. *Remote Sens* 13(3):351. <https://doi.org/10.3390/rs13030351>
- Levizzani V, Bauer P, Turk FJ (2007) *Measuring Precipitation from Space: EURAINSAT and the Future*, vol 28. Springer, Berlin/Heidelberg, Germany
- Haberlandt U (2007) Geostatistical interpolation of hourly precipitation from rain gauges and radar for a large-scale extreme rainfall event. *J Hydrol* 332(1):144–157. <https://doi.org/10.1016/j.jhydrol.2006.06.028>
- Qian B, Zhou J, Tong L, Shen X, Liu F (2016) Nonnegative matrix factorization with endmember sparse graph learning for hyperspectral unmixing. In: 2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25–28, 2016, pp. 1843–1847. <https://doi.org/10.1109/ICIP.2016.7532677>
- Tan TZ, Lee GKK, Liong SY, Lim TK, Chu J, Hung T (2008) Rainfall intensity prediction by a spatial-temporal ensemble. In: *Proc. of the IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 1721–1727. IEEE
- Kühnlein M, Appelhans T, Thies B, Nauss T (2014) Precipitation estimates from MSG SEVIRI daytime, nighttime, and twilight data with random forests. *J Appl Meteorol Climatol* 53(11):2457–2480
- Guarascio M, Folino G, Chiaravalloti F, Gabriele S, Procopio A, Sabatino P (2020) A machine learning approach for rainfall estimation integrating heterogeneous data sources. *IEEE Trans Geosci Remote Sens* 1–11. <https://doi.org/10.1109/TGRS.2020.3037776>

13. Tang G, Long D, Behrangi A, Wang C, Hong Y (2018) Exploring deep neural networks to retrieve rain and snow in high latitudes using multisensor and reanalysis data. *Water Resour Res* 54(10):8253–8278
14. Tao Y, Gao X, Ihler A, Hsu K, Sorooshian S (2016) Deep neural networks for precipitation estimation from remotely sensed information. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 1349–1355
15. Folino G, Guarascio M, Chiaravalloti F, Gabriele S (2019) A deep learning based architecture for rainfall estimation integrating heterogeneous data sources. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. <https://doi.org/10.1109/IJCNN.2019.8852229>
16. Grimes DIF, Coppola E, Verdecchia M, Visconti G (2003) A neural network approach to real-time rainfall estimation for Africa using satellite data. *J Hydrometeorol* 4(6):1119–1133
17. Rivolta G, Marzano F, Coppola E, Verdecchia M (2006) Artificial neural-network technique for precipitation nowcasting from satellite imagery. *Adv Geosci* 7:97
18. Turlapaty AC, Anantharaj VG, Younan NH, Joseph Turk F (2010) Precipitation data fusion using vector space transformation and artificial neural networks. *Pattern Recognit* 31(10):1184–1200. <https://doi.org/10.1016/j.patrec.2009.12.033>
19. Zhang P, Jia Y, Gao J, Song W, Leung H (2020) Short-term rainfall forecasting using multi-layer perceptron. *IEEE Trans Big Data* 6(1):93–106. <https://doi.org/10.1109/TBDATA.2018.2871151>
20. Tebbi MA, Haddad B (2016) Artificial intelligence systems for rainy areas detection and convective cells' delineation for the south shore of Mediterranean sea during day and nighttime using MSG satellite images. *Atmos Res* 178(Supplement C):380–392
21. Breiman L (1996) Bagging predictors. *Mach Learn* 24(2):123–140
22. Kühnlein M, Appelhans T, Thies B, Nauss T (2014) Improving the accuracy of rainfall rates from optical satellite sensors with machine learning—a random forests-based approach applied to msg seviri. *Remote Sens Environ* 141(Supplement C):129–143
23. Le Cun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
24. Choi Y, Cha K, Back M, Choi H, Jeon T (2021) Rain-f: A fusion dataset for rainfall prediction using convolutional neural network. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, pp. 7145–7148. <https://doi.org/10.1109/IGARSS47720.2021.9555094>
25. Huang G, Li Y, Pleiss G, Liu JE, Zand Hopcroft, Weinberger, KQ (2017) Snapshot ensembles: Train 1, get M for free. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings
26. Kühnlein M, Appelhans T, Thies B, Nauss T (2014) Improving the accuracy of rainfall rates from optical satellite sensors with machine learning—a random forests-based approach applied to msg seviri. *Remote Sens Environ* 141:129–143. <https://doi.org/10.1016/j.rse.2013.10.026>
27. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 770–778
28. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. ICML'10, pp. 807–814
29. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37. ICML'15, pp. 448–456
30. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15:1929–1958
31. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint [arXiv:1207.0580](https://arxiv.org/abs/1207.0580)
32. Guarascio M, Manco G, Ritacco E (2018) Deep learning. *Encycl Bioinform Comput Biol ABC Bioinform* 1–3:634–647
33. Loshchilov I, Hutter F (2017) SGDR: stochastic gradient descent with warm restarts. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings
34. Pai PF, Hong WC (2007) A recurrent support vector regression model in rainfall forecasting. *Hydrol Process* 21:819
35. Hiemstra P (2015) Automatic interpolation package
36. Stanski HR, Wilson LJ, Burrows WR (1989) Survey of common verification methods in meteorology. WMO World Weather Watch Tech. Rep. 8
37. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. *Inf Process Manage* 45(4):427–437
38. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
39. García S, Herrera F (2009) An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *J Mach Learn Res* 9:2677–2694

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.