Journal of Engineering Science and Technology
Vol. 17, No. 2 (2022) 1369 - 1387

A ROBUST UNCERTAINTY QUANTIFICATION AND COMPENSATION-BASED SYSTEM FOR IRRIGATION WATER DEMAND MANAGEMENT

M. ELIDRISSI*, O. ELBEQQALI, J. RIFFI

Faculty of Sciences, Sidi Mohamed Ben
Abdellah University, P.B. 1796 Fez-Atlas, 30003 Morocco
*Corresponding Author: mohammed.elidrissi29@usmba.ac.ma

Abstract

Irrigation water prediction is an important key for water irrigation management, without proper water forecasting, it will be difficult to have an idea about the amount of water that should be provided in storage ponds for irrigation purposes, too much water can lead to risks of wastage, and not enough means a risk of plant death of the thirst. On the other hand, the prediction of the amount of water contributes a lot to the worldwide water preservation plan and efficiently helps face the effects of climate change. To prevent the irrational use of water in the irrigation process, a smart water forecasting system is proposed to predict the exact quantity of water needed for irrigation purposes. The reference evapotranspiration parameter is forecasted in a given period of time using data collected by sensors scattered in the field. It was decided to use the dropout as a Bayesian optimization in regression context to model uncertainty in the neural network, as a substitution of gradient-based methods to optimize neural network model weights, the idea behind the proposed system is to quantify the neural network uncertainty by estimating the mean and the standard deviation of multiple reference evapotranspiration measurements, then calculate the total error that can be occurred and compensate it, to finally have a decision about the amount of water needed for irrigation task. Results have shown the great ability of the system to save water up to 19% compared to the situation without using the system.

Keywords: Error compensation, Neural network, Reference evapotranspiration, Uncertainty estimation, Water demand.

1. Introduction

Artificial neural networks are composed of layers and neurons, each neuron is connected to all the neurons of the adjacent layers, each neuron calculates the sum of its inputs using nonlinear functions, and its activation depends on whether the calculated sum exceeds the threshold defined by the activation function, the resulted signal (called weight) contributes to the activation of the next neurons. Weights are adjusted during the training process, the more the weights are adjusted, the better the network accuracy is. In the neural network context, the role of the optimization function used in the training process is to minimize the prediction error by looking for the global minimum of the loss function. We face multiple issues when we choose the optimization functions' parameters, such as the right value of learning rate (LR), a very small value of LR means a big time for the model to converge, and too big value leads to probable non-convergence of the model, moreover, the probability of being stacked in the local optimum instead of the global optimum remains a possibility.

Chen et al. [1] mentioned that the neural network optimization process is influenced by three critical topics: uncertainty quantification resulting from noisy data, or internal model settings or structure, training data availability, and the use of gradient-free based optimization algorithms such as Nelder Mead Simplex (Nonlinear Simplex) [2], Divided Rectangles Method [3], and Particle Swarm Optimization [4]. The challenge described in [1] is to propose a method for the optimization process without having recourse to derivative calculation, the complexity of derivative calculation as mentioned in [5] can be found in some artificial intelligence problems where the constraint function is an output of a black box that doesn't provide derivative information, this alternative method considers the weight's values as a probability distribution instead of fixed values, and uses Bayesian approach to analyse uncertainty over the distribution of the weights, without need to use derivative methods. The quantification of neural network uncertainty was introduced in several real-world machine learning applications, Zhang et al. [6] developed a framework for uncertainty prediction in complex fusion network, this strategy is applied to determine the uncertainty in autonomous vehicles trajectory, google brain team [7] proposed a module for neural network uncertainty (Bayesian layer) to capture uncertainty over the distribution of the weights in Bayesian neural network, pre-activation units in dropout regularization function, activation, or the function itself such as Gaussian Process GP.

The objective of this work is to exploit the probabilistic interpretation of dropout function to build a smart system for agricultural water demand forecasting, the proposed system quantifies uncertainty in neural network weights and estimates means and covariance of multiple measurements, it then calculates the total value of errors and compensates them, and finally provides a high precision about the exact quantity of water need to be provided for crops irrigation, this decision is notified to the farmer's control system, we next provide two choices to control the pump used to fill the tank, automatic or manual, depending on the farmer's desire.

2. The Challenge of Model Optimization in Neural Networks (NN)

The objective of a NN is to improve the accuracy of detection of the model in a given training set, by minimizing the cost function, the complexity to find an

optimized NN model is imposed by many factors [8] such as the fact that the problem the training process is dealing with is a non-convex function (has multiple local and global optimums), another important point is the big number of nodes a well as the high combination of weights the model supposed to have, without forgetting the hyperparameters, i.e., learning rate, regularization, CNN adds some other hyperparameters like number and shape of filters, number of dropout and max-pooling functions in each convolution layer, and number of nodes and layers in fully connected part. These hyperparameters are not directly learned from the training data set; while weights are constructed from the training process. But the calculation of the perfect set of values is still difficult, let us say impossible. Instead, the model adopts an optimization function, and the weights are adjusted using backpropagation.

The numerical optimization principles were applied for the first time to test the performance of genetic algorithms by De Jong [9] and have spread in many other areas [10], starting from job scheduling [11], aviation [12], vehicle routing [13], decision making [14]. The performance of an optimization function is tested using the well-known Rosenbrock test function, which is defined as [15]:

$$f(x) = 100({x_1}^2 - x_2)^2 + (x_1 - 1)^2, with, -2.048 \leq x_i \leq 2.048, i = 1,2 \quad (1)$$

For two-dimension function and extension to *n* variables function:

$$f(x) = \sum_{i=1}^{n-1}[100({x_i}^2 - x_{i+1})^2 + (x_i - 1)^2]\, with - 30 \leq x_i \leq 30, i = 1,2..,n \quad (2)$$

The choice of the loss function is still challenging since its main role is to boil down all the properties of the problem into one scalar, which allows improving the performance of the model by improving it.

3. Loss Function in Error Estimation

In order to estimate the error of weights in a NN model, many techniques are available in the literature, one of them is the Maximum Likelihood Estimation (MLE) which is a probabilistic framework used in statistics to estimate the parameters of a probability distribution of an observed sample of the dataset, by maximizing the likelihood function. The main objective of MLE in density estimation is to find out a set of parameters θ that maximizes the conditional probability $argmax(\, L(\theta/x_1, x_2, \ldots, x_n))$for each observed sample of the space $S=\{x_1, x_2, x_3, \ldots, x_n\}$. Regarding the chain rule for Bayesian networks, and assuming that data samples of the space *S* are disjoint given the model, the likelihood can be written as described in [16], $L(\theta/X_i) = \prod P(x_i/\theta), x_i \in X$ where x_i denotes data samples of the space *S*, and *Pa*(*Xi*) denotes the parents of *Xi* in Bayesian network. The product of multiple small probabilities can be practically converted to a summation of the log_2 of the conditional probability [17], it is called log-likelihood and is denoted as $\log L\,(\theta/X_i) = \sum \log_2 P\,(x_i/\theta)$.

Another technique of error estimation is cross-entropy (CE), called also *Kullback-Leibler distance* [18], it is defined as the difference between two discreet probabilities, its definition is given by Eq. (3). For the continuous distributions, the sum becomes integral:

$$CE(H, G) = \sum_{i=1}^{n-1} H\,(x_i) \log_2 G(x_i) \quad (3)$$

4. The Sources of Uncertainty and Quantification

When we build our NN predictive models, many questions begin penetrating our mind, are these models certain? Can we trust them? The answer for these questions is absolutely no, because there is no model 100% trustable, and this is justified by the fact that a margin of uncertainty always exists, it can be related to the choice of the internal model parameters, the noise in datasets, or even provided data are not enough for better pattern extraction. Uncertainty in predictive models in general, and Bayesian modelling particularly, has multiple sources as we mentioned in the introduction, Epistemic and Aleatoric uncertainties are the majors. Here, we remember these two concepts as proposed in [19, 20].

4.1. Epistemic uncertainty

Also called uncertainty in the model, it refers to the uncertainty in the model parameters, we do not have sufficient understanding of our model outputs based on our provided data, so quantifying weights uncertainty in classification Bayesian NN comes to replace the deterministic network's weights, by introducing distribution over these parameters. To capture this type of uncertainty in a given NN model, the distribution $P(W|X, Y)$ must be approximated, given that W is a prior distribution that can be put on weights' parameters (Gaussian distribution as an example $N(0, I)$), and a dataset $Xi=\{x1, x2, x3,\ldots,xn\}$, $Yi=\{y1, y2, y3,\ldots,yn\}$. Y. Kwon et al. [21] explain the estimation of uncertainty in NN much easier, the algebraic expression obtained from the law of the total variance is:

$$Var_q(p(y^*/x^*)) = E[yy^T] - E_q[y]E_q[y]^T = \int_\Omega [diag(E_{p(y^*/x^*,\omega)}[y^*]) - E_{p(y^*/x^*,\omega)}[y^*]E_{p(y^*/x^*,\omega)}[y^*]^T]q_\theta(\omega)d\omega + \int_\Omega [E_{p(y^*/x^*,\omega)}[y^*] - E_{q\theta(y^*/x^*)}(y^*)] [E_{p(y^*/x^*,\omega)}[y^*] - E_{q\theta(y^*/x^*)}(y^*)]^T q_\theta(\omega)d\omega \quad (4)$$

The variance refers to the difference between a given output y and the expected value y^* of a given input x^*. The epistemic uncertainty is mathematically expressed by the second term of Eq. (4).

4.2. Aleatoric uncertainty

Also called uncertainty in observations, it captures 'residual' noise in data introduced by physical sensors, or motion noise in the case of vision systems. This kind of uncertainty is divided into two types: Heteroscedastic Aleatoric uncertainty, which assumes that for each observation (x, y), the noise is different. On the other hand, Homoscedastic Aleatoric uncertainty which supposes that noise is the same for every data point (x, y). This type of uncertainty is expressed by the first term of Eq. (4), most of the terms of this equation are matrixes, the first element is the diagonal matrix of the expected output y^*, the second element is the product of the expected output y^* and its transpose based on the predictive posterior probability $p(y^*|x^*, \omega)$, these two elements are multiplied with the variational posterior distribution and integrated over ω, which is the elements of the weights space Ω. Practically, the posterior distribution is intractable, it is approximated using Laplace approximation [22]. The entire proof of Eq. (4) including detailed derivation, implementation, and additional visualization results are presented in *Appendix A* of [21].

5. Parameter Optimization Using Bayesian Theorem

Recent researches have been interested in uncertainty in NN models, particularly probability distribution over the weights, this is the principle of Bayesian estimation. Bayes by Backprop is a variational inference algorithm that was proposed in [23-25] to ameliorate generalization in non-linear regression problems, by learning posterior distribution on NN weights *w*. The principle of Bayesian inference supposes that all the weights of an NN model have a probability distribution instead of having a single fixed value, the assumption of having the exact probability distribution is difficult as the model contains a big number of weights. For this reason, an approximate distribution q_0 ($w \mid D$) is defined and is measured using relative entropy (also called Kullback-Leibler (K-L) divergence) [26], the same principle is used by Blundell et al. [25] to take a variational approximation to exact Bayesian updates, and thus they achieved performances similar to dropout algorithm. Kendall and Gal [20] proposed a method for estimating aleatoric uncertainty alone, epistemic uncertainty alone, or combining them both in one single model. Kwon et al. [21] presented the limitations of the model proposed by Kendall et Gall and proposed an improved version based on SoftMax normalization. Shridhar et al. [23] proposed another method to estimate uncertainty using SoftPlus normalization with variational inference.

6. Comparison with Other Methods

Numerous methods are proposed to estimate the water demand, some of them provide daily, monthly or yearly forecasting, and others are useful for real-time prediction demands. In general, two families of approaches are proposed, computational neural network-based methods [27, 28] and statistical-based methods [29]. Srivastava et al. [30] used an approach based on the Fuzzy-Ranking Algorithm to identify the significant variable for monthly rainfall forecast, another proactive theoretical framework is proposed by Delgoda et al. [31] as a requirement to implement model predictive control to adjust irrigation amount, this framework takes into account the biomass parameter without giving more importance to some specific environmental variables such as temperature, solar radiation, wind speed, and weather pressure.

Mishra et al. [32] used short-term weather forecasting and data from the field to calibrate the Soil Water Plant Atmosphere (SWAP) model and provide an efficient irrigation scheduling method, this study has the same issue as [31] since it uses only rainfall forecasts and ignore the other detailed environmental variables like humidity, air temperature, wind speed and pressure, the model proposed in that study is also extremely sensitive to the soil profile. Azhar et al. [33] proposed a method to deal with zeros in data series to model the short-term irrigation scheduling, this work reported that uncertainty in precipitation arises with lead time, additionally, this model does not take into account the reference evapotranspiration and rainfall which makes it less accurate and less precise. Bai et al. [34] proposed an approach to predict urban daily water demand based on multi-scale relevance vector regression optimized with particle swarm optimization algorithm, this approach focused on forecasting ET_0 using only three environmental variables notably solar radiation, air temperature, and wind speed, but practically ET_0 variation is sensitive to much more parameters, some of them are related to cultural characteristics while others are associated to weather such as air pressure, humidity, and soil moisture. Moreover, Perera et al. [35] have evaluated reference daily evapotranspiration ET_0 using numerical weather prediction, their proposed quantification ignores to evaluate the uncertainty that can be

introduced either by data impurity, error generated by data collection systems or uncertainty in the prediction model.

The potential of our proposed method compared with all these works is the ability to predict the amount of water needed for irrigation by forecasting the reference evapotranspiration ET_0 taking into consideration various hydrological and environmental variables such as humidity, wind speed, air pressure, cultural coefficient, as well as quantifying the uncertainties that can be generated and compensate them. The best accuracy obtained value is 96.31%, and the worst precision of prediction is 85.01%.

7. Evapotranspiration Estimation

In this research paper, we propose a smart irrigation model able to forecast the amount of water that should be provided in the tank for irrigation purposes. Predicting the exact quantity of water is achieved by quantifying and compensating the error caused by uncertainty, both in the prediction model and during measurements with hardware sensors. The proposed architecture of the system supposes that a meteorological station is installed in the field to collect data about temperature, humidity, wind speed, atmospheric pressure, vapour pressure, and combined with solar radiation collected by meteorological satellite in a given high. All these parameters are then used to calculate the reference evapotranspiration ET_0 according to the Penman and Monteith equation [36] and Food and Agriculture Organization (FAO) research [37] (a detailed presentation of how we do that is available in *Appendix A*). After that, the real crops evapotranspiration *ETc* is calculated based on the reference evapotranspiration, the cultural coefficient *Kc* and the soil characteristic *Ks* as follows:

$$ET_c = K_c * K_s * ET_0 \tag{5}$$

The cultural coefficient curve illustrated in Fig. 1 is presented for the first time by Doorenbos and Pruitt [38], it is divided into three zones, initial at the beginning of the plant lifetime, mid-season in the middle of the plant lifetime, and end season at the end of the plant lifetime, the plants have a specific coefficient *Kc* in each lifetime zone, the minimal values (in the beginning and at the end of the lifetime) are observed in the summer where the humidity is low, and its maximal value (the mid-season zone) is observed in the spring and autumn seasons because the humidity ratio is high.

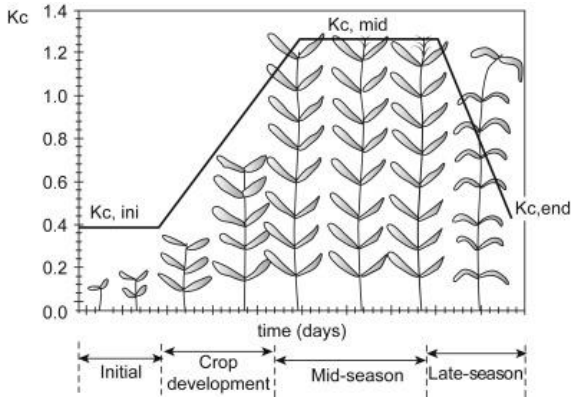

Fig. 1. The cultural coefficient curve presented by Doorenbos, and Pruitt [38].

The exact need of water (*NOW*) in a given period is equal to the expected precipitation *EP* subtracted from the predicted actual Evapotranspiration *ETc*. *EP* can be obtained either from precipitation forecasting satellite systems if they exist, or from the extrapolation of precipitation curve using historical data, so:

$$NOW = ET_c - EP \tag{6}$$

The architecture of the proposed framework of irrigation water management is given in Fig. 2, the process begins with collecting inputs from the meteorological station, and satellite data repository which is a database that contains spatial parameters required for ET_0 calculation. Then, we present these parameters to the ML (Machine learning) model that predicts the value of the evapotranspiration *ETc* in $1m^2$ of the field, assuming that the rainfall distribution is uniform in the field, and the topography of the field does not affect water infiltration, the amount of water the crops will need in a given time is calculated as follows:

Let us represent one-meter square with a container of 1m high by 1m width by 1m long. After a rain fall, the high of collected water in the container is 5 mm, the total volume of water gathered in the container is: $5E^{-3} * 1 * 1 = 5E^{-3}m^3$ or 5 litres, let's now suppose that the surface of the field that we want to irrigate is 43 hectares (*ha*), the total amount of water needed is $5E^{-3} * 46ha = 215E^3 \text{m}^3$.

The farmer chooses between automated control and manual control for the tank filling system, if the automatic mode is chosen, the system activates a pump to fill the tank with the exact predicted quantity of water following the crops lifetime calendar. If for some raisons, the farmer selects the manual mode, he must activate the pump manually at each system notification.

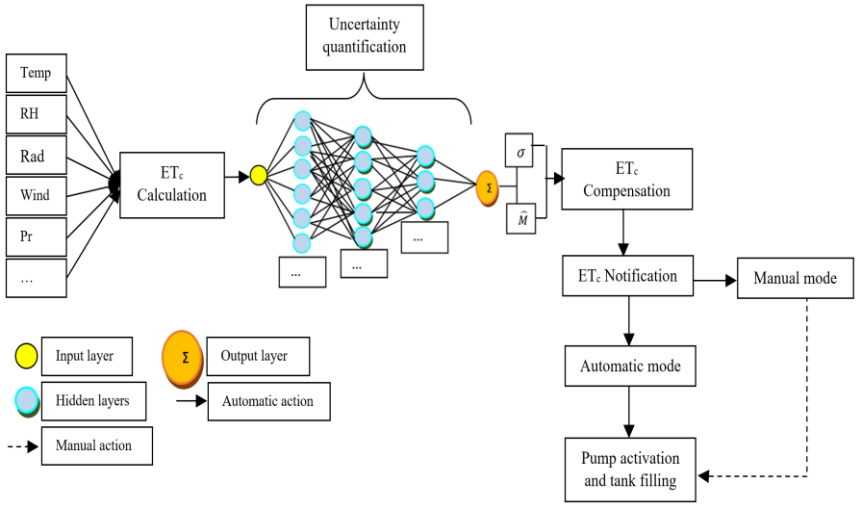

Fig. 2. Irrigation water forecasting system based on neural network combined with uncertainty quantification.

The system components:

- *ETc* calculation: a network of sensors should be installed in the field for data collection, we gather information about daily minimal and maximal temperature, the average of daily vapour and atmosphere pressures, and the

average of daily wind speed, then we calculate the actual evapotranspiration using the equation described in *Appendix A*.

- Uncertainty quantification: we described above that our observed data are always subject to aleatoric and epistemic uncertainties. To measure the confidence in the calculated *ETc*, we approximate the standard deviation with reference to an estimated mean, assuming that the same *ETc* value is measured multiple times, the uncertainty is obtained from dropout neural network, it usually refers to dropout variational distribution [39], in other words, doing Bayesian approximation. The uncertainty quantification layer is like a measuring tool that helps to predict what will be the value of its input if we repeat the measure multiple times, and in the same conditions, it statistically estimates the error of repeatability, this error type is evaluated using statistical methods that involve the mean and the standard deviation of multiple times of measurements. Statistically speaking, the uncertainty of measurement is obtained using Eq. (7):

$$u = \sigma_{n-1} \div \sqrt{n} \tag{7}$$

with *n* is the number of times the measurements are repeated, we denote by $\bar{m}$ the empirical mean of measurements, and $\bar{m} = (1 \div n) \sum_{i=1}^{n} m_i$, so the statistical deviation is expressed by Eq. (8):

$$\sigma_{n-1} = \sqrt{((1 \div (n-1)) \sum_{i=1}^{n} (m_i - \bar{m}_i)^2)} \tag{8}$$

- *ETc* compensation: while predicting *ETc*, we have taken into account both aleatoric and epistemic error types, the calculated *SD* indicates how far the predicted *ETc* is from the real value of ET_{CR} which is near the mean value in most of the statistical situations, thus we compensate this error by adding or subtracting the estimated value of uncertainty from the mean value of the predicted *ETc*, thus:

$$ET_c = \overline{ET_c} \pm U \tag{9}$$

Supposing now that the acceptable confidence value is 95%, the expanded uncertainty *U* of measurement is given by Eq. (10):

$$U = (k\sigma_{n-1}) \div \sqrt{n} \tag{10}$$

with k=1.96 is the expanding factor.

- ET_C notification: when the exact quantity of irrigation water needed is forecasted, the notification process is triggered, it notifies the farmer about the final decision made, and continue filling the tank depending on whether the farmer has chosen the manual or the automatic operation mode.

8. Experimentation

In this section, we perform the step by step experiment to evaluate and test our system, let us first recall experimentally how dropout variational inference works, to do so, we take a subset of the daily Reference (R_{ET}) and Potential (P_{ET}) Evapotranspiration dataset over Florida in 2018 collected and published by Shoemaker [40], we choose the dataset of Florida region for two reasons: our test field has the same climate and weather characteristics as Florida region dataset, on the other hand, our country doesn't have the required satellite system for weather characteristics measuring. The subset is used to look for the distribution of the R_{ET} uncertainty. Thus, we built the

model shown in Fig. 2, it has one input, four hidden layers and two outputs, we use a dropout function between every two layers with 50% of dropout rate and a RELU activation function, the input represents the measured R_{ET} that we want to estimate the uncertainty of measurement, and the outputs are the mean and the standard deviation that will help us to compensate the error made by the sensors when measuring physical parameters, as well as by the model during the training process. We first show how the reference evapotranspiration evolves during 200 measured data points, this evolution is illustrated in Fig. 3, data points are normalized before being plotted using the mean and standard deviation of the observed data:

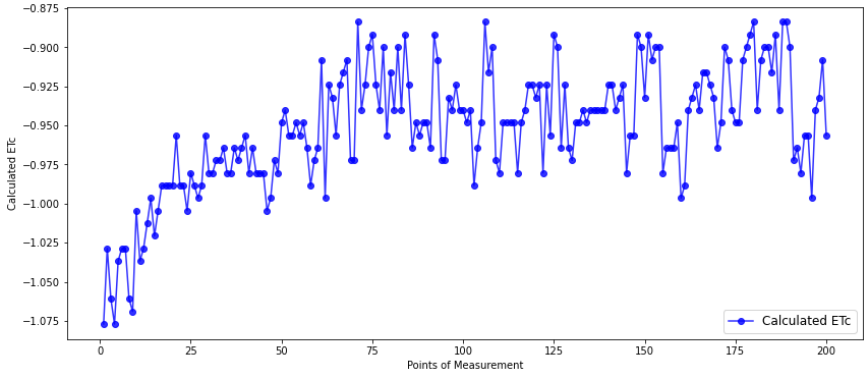

Fig. 3. Evolution of R_{ET} during 200 data points measurement.

Now, we train the model using the dataset described earlier during 30000 iterations, then we use the parameters of the pre-trained model to predict the values of R_{ET}. Figure 4 illustrates the observed values of R_{ET} and those predicted in the same graph.

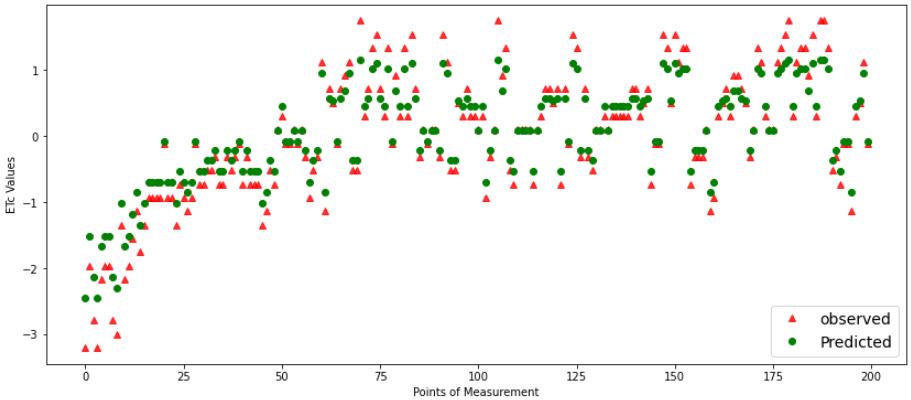

Fig. 4. Evolution of calculated and predicted R_{ET}.

As obviously shown in Fig. 4, the R_{ET} data points of the subset, and the model predicted values are identical in most of the graph regions, this means that the model has developed a good accuracy of prediction during the training process. Thanks to the dropout function, the model could avoid the overfitting

phenomenon in the early stages and catalyse all nodes to contribute to performance improvement.

The next step is to estimate the uncertainty of another 200 new standardized data points, by evaluating the distribution of the mean and the uncertainty. We repeat the estimation operation during 300 uncertainty iterations, then we take the mean and the standard deviation, given that the number of standard deviations to plot is 2.

The uncertainty distribution as shown in Fig. 5 is illustrated in the dashed blue curve; the green curve represents the mean of test values after 300 uncertainty iterations.

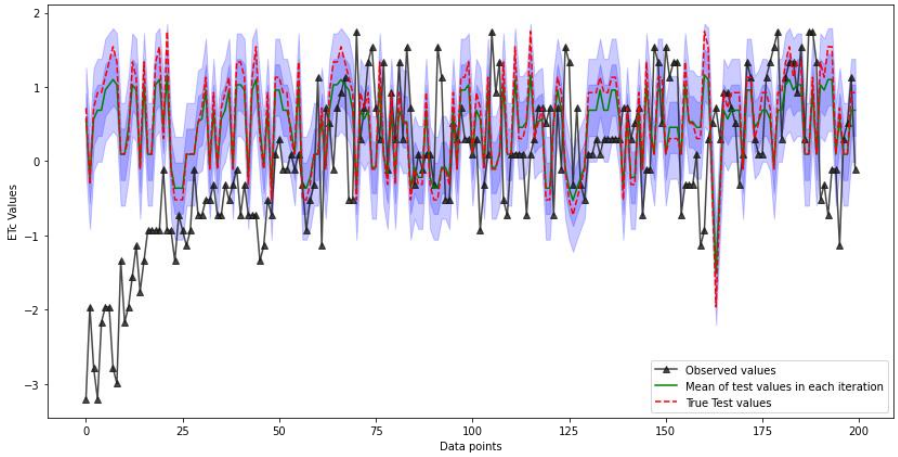

Fig. 5. Mean and uncertainty curves of predicted *ETc* values compared to observed and true test values within 300 uncertainty iterations.

We introduced the compensation feature to eliminate the uncertainty estimated by the model, the compensation is done by subtracting or adding the uncertainty value to the mean of the estimated value, assuming that the confidence value is 95%. Table 1 summarizes the findings of the experiment.

Table 1. In order to touch the usefulness of our system, we made a comparison between the amount of water consumed before and after using the system, to irrigate a field with 46 hectares of surface area, data samples are extracted in random days, and weather forecasted precipitations are collected from Moroccan weather forecasting system available here [41].

Sampling number	1
ETc data points measurement (mm)	1,03
Forecasted ETc (mm)	2,5913
Weather Forecasted Precipitation (mm)	0,02
Estimated mean	2,6150
Estimated SD	0,5388
Compensated error	0,0610
ETc data after compensation	2,5540
Difference between Etc and rainfall	2,5340
Amount of needed water after using the system (Liters)	116,57
Amount of needed water without the system (Liters)	47,38

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1,45	1,76	1,93	1,92	2,00	2,03	2,04	2,02	2,01	2,05	2,10	2,07	2,11	2,18	2,16	2,21	2,26	2,25	2,24
2.6550	2.6368	2.6543	2.6141	2.6386	2.5847	2.7004	2.6038	2.6339	2.6604	2.6984	2.6429	2.6619	2.6377	2.6809	2.6433	2.6596	2.6542	2.6342
0,40	4,00	3,50	5,00	2,40	0,06	0,08	1,00	0,06	0,07	0,09	0,01	0,02	0,07	0,08	0,02	0,05	0,09	0,10
2,6259	2,6345	2,6387	2,6443	2,6419	2,6427	2,6403	2,6400	2,6448	2,6419	2,6399	2,6447	2,6442	2,6445	2,6427	2,6437	2,6519	2,6501	2,6477
0,6553	0,7304	0,7704	0,8471	0,7723	0,8379	0,7928	0,8530	0,8166	0,8075	0,7725	0,8000	0,8324	0,7965	0,8450	0,7850	0,9372	0,9176	0,9015
0,0741	0,0826	0,0872	0,0959	0,0874	0,0948	0,0897	0,0965	0,0924	0,0914	0,0874	0,0905	0,0942	0,0901	0,0956	0,0888	0,1061	0,1038	0,1020
2,5518	2,5519	2,5515	2,5484	2,5545	2,5479	2,5506	2,5435	2,5523	2,5506	2,5525	2,5542	2,5500	2,5544	2,5471	2,5549	2,5459	2,5463	2,5457
2,1518	-1,4481	-0,9485	-2,4516	0,1545	2,4879	2,4706	1,5435	2,4923	2,4806	2,4625	2,5442	2,5300	2,4844	2,4671	2,5349	2,4959	2,4563	2,4457
98,98	-66,61	-43,63	-112,77	7,11	114,44	113,65	71,00	114,65	114,11	113,27	117,03	116,38	114,28	113,49	116,60	114,81	112,99	112,50
66,70	80,96	88,78	88,32	92,00	93,38	93,84	92,92	92,46	94,30	96,60	95,22	97,06	100,28	99,36	101,66	103,96	103,50	103,04

21	2,27	2.6479	0,09	2.6475	0,8564	0,0969	2,5506	2,4606	113,19	104,42
22	2,23	2.6473	0,05	2.6460	0,8567	0,0969	2,5491	2,4991	114,96	102,58
23	2,25	2.6004	0,09	2.6481	0,8707	0,0985	2,5496	2,4596	113,14	103,50

9. Discussion

Table 1 provides the details of a real experiment that we performed to consolidate our findings, to better explain the results, let's take the first row and explain how each parameter is found, *ETc* data points measurement is calculated using the parameters gathered from the field sensors, Forecasted *ETc* is the predicted value of *ETc* taking into account the uncertainty quantification expressed by the estimated mean and *SD* parameters. *ETc* after compensation is found by extracting the quantified error described in Eq. (10) from the estimated mean of the predicted *ETc*, next, the real amount of water needed is found by extracting the weather forecasted rainfall from the compensated value of *ETc*. The exact amount of water is calculated by multiplying the surface area (46 hectares) by the real amount of water. We can interpret the negative values observed in the amount of needed water without the system (Table 1 and Fig. 6) as a water surplus between the forecasted need for water and the forecasted rainfall, no irrigation is needed in this case.

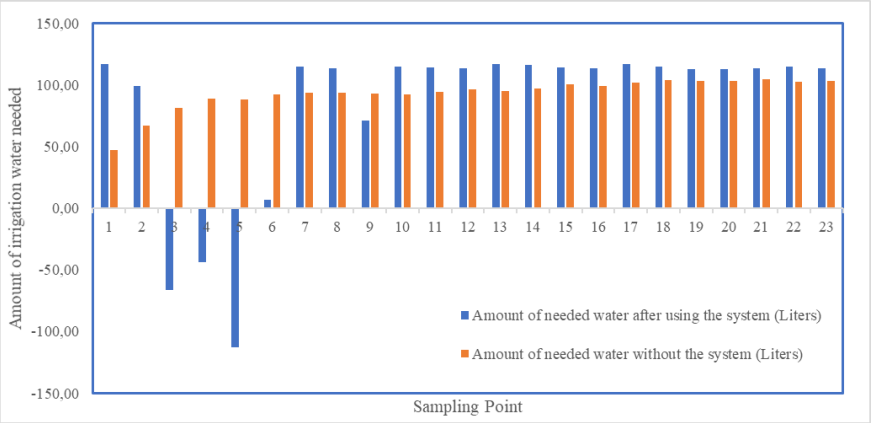

Fig. 6. Comparison between the amount of water needed for irrigation purposes with and without using the system.

The quantity of saved water is ranging from 15% and 20% in the growing season of the wheat plant. The uncertainty estimation model can be considered as a measurement tool that repeats the prediction of *ETc* for many iterations before calculating the mean and the variance, then evaluates the total uncertainty that was occurred in the measurement chain (error in physical sensor layer, aleatoric and epistemic errors in the prediction model), the compensation phase

gives the real value of *ETc* that should be compensated. Without using the system, the tank is always filled with water as well as the quantities of water provided to the crops are random, it does not take into account either forecasting precipitations or evapotranspiration factor, because the big concern of the farmer is to keep his crops alive during its growing lifecycle, this classical approach founded on intuition and sensation can cause a big wastage of water, and can threat human life, especially in some regions where the potable water is used in daily irrigation tasks.

10. Conclusion and Future Work

This contribution tries to be involved in the worldwide water preservation plan, it aims to consolidate the principle of rational use of water in all our daily activities, especially in irrigation process, water resources are rapidly decreasing because of climate change, and the irrational use of water by humankind, which is why finding scientific models that can help to fight these challenges becomes more urgent than before.

In this paper we have exploited the potential of Dropout function as a probabilistic interpretation to build robust water need forecasting system for irrigation purposes, we have built a neural network model for evapotranspiration prediction, and estimate errors that can be occurred either in the prediction process or data collection operation by sensors in the field, next we have proposed a way to compensate the estimated errors based on the mean and the standard deviation, assuming that confidentiality level is 95%.

The forecasting skills of the proposed system can be improved in future conducted studies taking into account more meteorological and environmental variables notably plant internal characteristics and some parameters related to the soil and the quantity of water evaporated from the storage area.

Nomenclatures

a	The reflexion coefficient
D	Data
$diag$	Diagonal matrix
E	Expectation of a Random Variable
G	Solar constant (0.0820 $MJm^{-2}min^{-1}$)
G, H	Two discreet probability densities
H_{min}, H_{max}	Maximal and Minimal of Humidity values
h, Z	The height where the measurement is performed
J	Number of days in the year between 1 and 365 or 366
K	The expanding factor
Kc	Cultural coefficient
Ks	Soil coefficient
$\overline{m}$	Mean value
N	Uncertainty iteration number
$N(0,I)$	Gaussian distribution.
P	The atmospheric Pressure
P_a	Parent probability distribution in Bayesian network

PT, TT	Psi Term and Temperature Term, both of them are used as an auxiliary for wind speed calculation
P_{xi}, q_w	Probability distribution
R_a	The extra-terrestrial radiation ($MJm^{-2}day^{-1}$)
R_n	The net solar radiation ($MJm^{-2}day^{-1}$)
R_{nl}	The net outgoing long waves solar radiation ($MJm^{-2}day^{-1}$)
R_{ns}	The net shortwave solar radiation ($MJm^{-2}day^{-1}$)
R_s	The mean daily solar radiation
S	Space of samples
T_m	The mean daily air temperature at 2m height, °C
Var	Variance
w	Neural network weight
x_i, y_i	Observation samples
x^*, y^*	Optimal values of x and y

Greek Symbols

γ	Psychrometric constant, KPa $°C^{-1}$
Δ	The slop of saturation vapour pressure curve
ΔT	Delta Term, used as an auxiliary for radiation calculation
δ	Longitude, radian
δ_d	Refers to the solar declination, Rad
σ_{n-1}	Standard deviation
θ	Model hyperparameters
$\mu 2$	Refers to the average daily wind speed measured at 2 m height, ms^{-1}
σ	Boltzmann constant, $MJK^{-4}m^{-2}$
ρ_a	The actual vapour pressure derived from relative humidity, KPa
ρ_s	The mean saturation vapour pressure derived from air temperature, KPa
Φ	Latitude, radian

Abbreviations

CE	Cross-Entropy
CNN	Convolutional Neural Network
EP	Forecasted precipitation
ETCR	Real Crops Evapotranspiration
FAO	Food and Agriculture Organization
GP	Gaussian Process
K-L	Kullback-Leibler
LR	Learning Rate
ML	Machine Learning
MLE	Maximum Likelihood Estimation
NN	Neural Network
NOF	Need Of Water
P_{ET}	Potential Evapotranspiration
RELU	Rectified Linear Unit
R_{ET}, ET_0	Reference Evapotranspiration
SD	Standard Deviation
SWAP	Soil Water Plant Atmosphere
U	Uncertainty

References

1. Chen, Y.; Chang, H.; Meng, J.; and Zhang, D. (2019). Ensemble neural networks (ENN): A gradient-free stochastic method. *Neural Networks*, 110, 170-185.
2. Singer, S.; and Nelder, J. (2009). Nelder-mead algorithm. *Scholarpedia*, 4(7), 2928.
3. Nicholas, P.J. (2015). A dividing rectangles algorithm for stochastic simulation optimization. 14*th informs computing society conference*. Virginia, USA, 47-61.
4. Wang, D.; Tan, D.; and Liu, L. (2018). Particle swarm optimization algorithm: An overview. *Soft Computing*, 22, 387-408.
5. Larson, J.; Menickelly, M.; and Wild, S.M. (2019). Derivative-free optimization methods. *Acta Numerica*, 28 (2019), 287-404.
6. Zhang, F.; Martinez, C.M.; Clarke, D.; Cao, D.; and Knoll, A. (2019). Neural network based uncertainty prediction for autonomous vehicle application. *Frontiers in Neurorobotics Journal*, 12(13), 1-17.
7. Tran, D.; Dusenberry, M.W.; Van Der Wilk, M.; and Hafner, D. (2019). Bayesian layers: A module for neural network uncertainty. 33*rd Conference on Neural Information Processing Systems*(*NeurIPS* 2019). Vancouver, Canada.
8. Shrestha, A.; and Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, 7, 53040-53065.
9. De Jong, K.A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. Ph.D. Dissertation. University of Michigan, USA.
10. Odili, J.B.; and Romli, A. (2017). Implementation evaluation of Cuckoo search for the benchmark Rosenbrock test function. *Proceedings of the* 8*th International Conference on Information Technology* (*ICIT*). Amman, 334-337.
11. Taheri, J.; Lee, Y.C.; Zomaya, A.Y.; and Siegel, H.J. (2013). A bee colony based optimization approach for simultaneous job scheduling and data replication in grid environments. *Computers and Operations Research*, 40(6), 1564-1578.
12. West, P.; Sculli, G.; Fore, A.; Okam, N.; Dunlap, C.; Neily, J.; and Mills, P. (2012). Improving patient safety and optimizing nursing teamwork using crew resource management techniques. *Journal of Nursing Administration*, 42(1), 15-20.
13. Odili, J.B.; Kahar, M.N.M.; and Anwar, S. (2015). African buffalo optimization: A swarm-intelligence technique. *Procedia Computer Science*, 76, 443-448.
14. Odili, J.B. (2013). Staff development programs and job performance: Implications for productivity in Lagos state ministry of education. *Journal of Business Administration and Management Sciences Research*, 2(12), 330-334.
15. Shang, Y.; and Qiu, Y. (2006). A Note on the extended rosenbrock function. *Evolutionary Computation*, 14(1), 119-126.
16. Jensen, F.V.; and Nielsen, T.D. (2007). *Bayesian network and decision graph* (2nd ed.). New York: Springer-Verlag.
17. Ian, G.; Yoshua, B.; and Courville, A. (2016). *Deep Learning*. MIT Press.
18. Rubenstein, R.Y.; and Kroese, D.P. (2004). *The cross-entropy method*. New York: Springer-Verlag.

19. Kwon, Y.; Won, J.; Kim, B.J.; and Cho Paik, M. (2018). Uncertainty quantification using Bayesian neural networks in classification: Application to ischemic stroke lesion segmentation. *Proceedings of the* 1*st conference on Medical Imaging with Deep Learning*. Amsterdam, The Netherlands.
20. Kendall, A.; and Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*. California, USA, 5574-5584.
21. Kwon, Y.; Won, J.; Kim, B.J.; and Cho Paik, M. (2020). Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics and Data Analysis*, 142, 106816.
22. Peng, R.D. (2020). Advanced statistical computing. Retrieved May 13, 2020, from https://bookdown.org/rdpeng/advstatcomp/laplace-approximation.html.
23. Shridhar, K.; Laumann, F.; and Liwicki, M. (2018). Uncertainty estimations by softplus normalization in Bayesian convolutional neural networks with variational inference. Retrieved April 20, 2020, from https://arxiv.org/pdf/1806.05978.
24. Graves, A. (2011). Practical variational inference for neural networks. *Proceedings of the* 24*th International Conference on Neural Information Processing Systems*. Vancouver, Canada, 2348-2356.
25. Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. (2015). Weight uncertainty in neural networks. *Proceedings of the* 32*nd International Conference on Machine Learning*. Lille, France, 1613-1622.
26. Kullback, S.; and Leibler, R.A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), 79-86.
27. Odan, F. K.; and Reis, L.F.R. (2012). Hybrid water demand forecasting model associating artificial neural network with Fourier series. *Journal of Water Resources Planning and Management*, 138(3), 1287-1305.
28. First, M.; Turan, M.E.; and Yurdusev, M.A. (2010). Comparative analysis of neural network techniques for predicting water consumption time series. *Journal of Hydrology*, 384(1-2), 46-51.
29. Torres, A. F.; Walker, W.R.; and McKee, M. (2011). Forecasting daily potential evapotranspiration using machine learning and limited climatic data. *Agricultural Water Management*, 98(4), 553-562.
30. Srivastava, G.; Panda, S.N.; Mondal, P.; and Liu, J. (2010). Forecasting of rainfall using ocean-atmospheric indices with a fuzzy neural technique. *Journal of Hydrology*, 395(3-4), 190-198.
31. Delgoda, D.; Malano, H.; Saleem, S.K.; and Halgamuge, M.N. (2016). Irrigation control based on model predictive control (MPC): Formulation of theory and validation using weather forecast data and AQUACROP model. *Environmental Modelling and Software*, 78, 40-53.
32. Mishra, A.; Siderius, C.; Aberson, K.; Van Der Ploeg, M.; and Froebrich, J. (2013). Short-term rainfall forecasts as a soft adaptation to climate change in irrigation management in North-East India. *Agricultural Water Management*, 127, 97-106.

33. Azhar, A.H.; and Perera, B.J.C. (2011). Prediction of rainfall for short term irrigation planning and scheduling-case study in Victoria, Australia. *Journal of Irrigation and Drainage Engineering*, 137(7), 435-445.
34. Bai, Y.; Wang, P.; Li, C.; Xie, J.; and Wang, Y. (2014). A multi-scale relevance vector regression approach for daily urban water demand forecasting. *Journal of Hydrology*, 517, 236-245.
35. Perera, K.C.; Western, A.W.; Nawarathna, B.; and George, B. (2014). Forecasting daily reference evapotranspiration for Australia using numerical weather prediction outputs. *Agricultural and Forest Meteorology*, 194, 50-63.
36. Monteith, J.L. (1965). Evaporation and environment. *Symposia of the Society for Experimental Biology*, 19, 205-234.
37. Allen, R.G.; Pereira, L.S.; Raes, D.; and Smith, M. (1998). Crop evapotranspiration - Guidelines for computing crop water requirements. Retrieved March 20, 2020, from http://www.fao.org/3/X0490E/x0490e06.htm.
38. Doorenbos, J.; and Pruitt, W.O. (1975). Guidelines for predicting crop water requirements. Retrieved August 9, 2020, from http://www.fao.org/3/f2430e/f2430e.pdf.
39. Gal, Y.; and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of the* 33*rd International Conference on International Conference on Machine Learning*. New York, USA, 1050-1059.
40. Shoemaker, W.B. (2019). Daily reference (RET) and potential (PET) evapotranspiration data over Florida. Retrieved May 1, 2020, from https://water.usgs.gov/GIS/dsdl/Florida_2018.zip.
41. Meteo Maroc (Moroccan weather), Inc. (2002). Weather forecast. Retrieved May 15, 2020, from http://www.meteomaroc.com/.

Appendix A

A short review on FAO evapotranspiration (ET) theoretical calculation based on Penman-Monteith equation

In this Appendix, we review the theoretical calculation of FAO Evapotranspiration (FAO-ET) using the Penman-Monteith equation and based on the following environmental parameters: Air temperature, solar radiation, relative humidity, wind speed, environmental aspects, crops characteristics, cultivation practices.

The formula of ET_0 given below refers to the reference evapotranspiration that was derived by FAO from the Penman-Monteith equation:

$$ET_0 = \frac{0.408\Delta(R_n - G) + \delta_d(\frac{900}{T_m+273})\mu_2(\rho_s - \rho_a)}{(\Delta + \gamma(1+0.34\mu_2))} \quad (11)$$

Now let us express each term of the equation ET_0 with its elementary equation.

$$\Delta = \frac{4098\left[0.6108\, exp\left(\frac{12.27T_m}{T_m+237.3}\right)\right]}{(T_m+237.3)^2} \quad (12)$$

T_m refers to the mean daily air temperature at 2m height, it is calculated as follows:

$$T_m = \frac{T_{max} - T_{min}}{2} \quad (13)$$

T_{max} and T_{min} consecutively are the height and the low temperatures measured during the experience.

Now the solar declination δ_d is calculated using Eq. (14):

$$\delta_d = 0.409 \sin\left[\frac{2\Pi J}{365} - 1.39\right] \quad (14)$$

With J refers to the day number in the year between 1 and 365 or 366.

The net solar radiation parameter is divided into two components, the net shortwave solar radiation R_{ns}, and the net outgoing longwave solar radiation $R_{nl}(MJm^{-2}day^{-1})$, so:

$$R_n = R_{ns} - R_{nl} \quad (15)$$

Given that R_{ns} can be found with; $R_{ns}=(1-a)\ R_s$, where R_s is the mean daily solar radiation and a is the reflection coefficient, it equals 0.23 for hypothetical grass reference crop. As for R_{nl}, it is obtained using Boltzmann law:

$$R_{nl} = \frac{\sigma\left[(T_{\max} + 273.16)^4 + (T_{\min} + 273.16)^4\right]}{2}(0.34 - 0.14\sqrt{\rho_a})\left[\frac{1.35R_s}{R_{s0}} - 0.35\right] \quad (16)$$

Note that σ = 4.903E10^{-9} (MJK^{-4}m^{-2}) is a constant of Boltzmann, and $R_{s0}=(0.75+2E10^{-5}\ Z)\ R_a$, with Z is the elevation above the sea level and R_a is the extra-terrestrial radiation (MJm^{-2}day^{-1}) given by:

$$R_a = \frac{24(60)}{\Pi} G d_r[(\omega_s \sin\vartheta \cos\theta) + (\sin\omega_s \cos\vartheta \cos\delta)] \quad (17)$$

where ω_s is the sunset hour angle in radian and φ and δ respectively are the latitude and the longitude in radian, if their values are given in decimal degree, they can be converted to radian using this formula: $\varphi_{rad}=\pi/180\ \varphi_{dec}$.

The average daily wind speed μ_2 is measured at 2 m height, sometimes wind speed is calculated in a height different from 2 m, in that case the calculated value is converted using the following equation:

$$\mu_2 = \frac{4.87\mu_h}{ln(67.8h - 5.42)} \quad (18)$$

With h is the measurement height of the value.

In order to calculate the mean saturation vapour pressure ρ_s, it is first necessary to measure maximal and minimal saturation vapour pressures and apply this formula: $\rho_s=(\rho_{max}+\rho_{min})/2$, where ρ depends on the temperature, thus:

$$\rho = 0.6108\, exp\frac{17.27T}{T+273} \quad (19)$$

The actual vapour pressure ρ_a derived from relative humidity and temperature can be found using Eq. (20):

$$\rho_a = \frac{\rho_{\min}\left(\frac{H_{\min}}{100}\right) + \rho_{\max}\left(\frac{H_{\max}}{100}\right)}{2} \quad (20)$$

With H_{min} and H_{max} respectively are the maximal and minimal values of the humidity measured during the experience.

The psychrometric constant γ is: γ=0.00065P, with P is the atmospheric pressure in Kilopascal, if this pressure is measured in a specific height Z, the value can be adjusted using this formula:

$$P = 101.3\left[\frac{293-0.0065Z}{293}\right]^{5.26} \quad (21)$$

Another parameter called Delta Term used as an auxiliary for radiation calculation:

$$\Delta T = \frac{\Delta}{\Delta+\gamma(1.0.34\mu_2)} \quad (22)$$

and Psi Term PT used as an auxiliary for wind speed calculation:

$$PT = \frac{\gamma}{\Delta+\gamma(1+0.34\mu_2)} \quad (23)$$

The temperature term TT is also an auxiliary calculation for wind speed:

$$TT = \frac{900\mu_2}{T_{min}+273} \quad (24)$$

The R_{ng} factor represents the equivalent evapotranspiration of R_n, it is obtained using R_{ng}=0.408R_n, and ET_{rad} is the evapotranspiration radiation term (mmd^{-1}), it can also be calculated using this equation: RT_{rad}=ΔT*R_{ng}, where ΔT is the delta term.

Calculation of reference evapotranspiration need another factor called evapotranspiration wind term ET_{wind}, which can be found using this formula:

$$ET_{wind} = PT \times TT(\rho_s - \rho_a) \quad (25)$$

Finally, the evapotranspiration is the sum of ET_{wind} and ET_{rad} using Eq. (26):

$$ET_0 = ET_{rad} + ET_{wind} \quad (26)$$