Optimization and Automation of Multi-Layered EnsembleLearning for Short-Term Forecasting in Agro-Climatology

Rohit Annad Gupta, Xiaoyang Liu, Jade Eva Guisiano, Raja Chiky, Julien Orensanz, Shohreh Ahvar

Optimization and Automation of Multi-Layered Ensemble Learning for Short-Term Forecasting in Agro-Climatology

Rohit Annad Gupta*, Xiaoyang Liu**
Jade Eva Guisiano ** Raja Chiky**
Julien Orensanz***, Shohreh Ahvar**

*Skoleom, Paris, France
r.gupta@skoleom.com
**ISEP, Paris, France
{xiaoyang.liu, jguisiano, rchiky, sahvar}@isep.fr
***Cap2020, Gradignan, France
jorensanz@cap2020.fr

Abstract. Agriculture is one of the areas whose activities heavily depend on weather forecasts. This paper, using Spark, proposes OptMLEL, an improved version of our previous work (i.e., MLEL) for short-term forecasting to assist farmers in their decision-making. OptMLEL helps to resolve the computational bottleneck when training MLEL in a single computational node. It also applies new features in training, tunes the algorithm parameters automatically in case of adding/removing data source providers and selects the number of layers automatically based on the amount of data. The obtained results show the applicability and performance of the proposed method.

1 Introduction

The reliability of weather forecasts is necessary for many areas for which activities depend on weather conditions. Agriculture is one of this areas that can be particularly impacted by weather events such as extreme temperatures, wind, storm, rain, etc. These events can cause significant damage to harvests and the result can be the total or partial loss of production. In addition to the damage, certain climatic conditions can also affect cultural operations such as the limited possibility of treatment in windy conditions, but also the difficulty of access to the soil with agricultural machinery in case of rain.

The challenge for farmers is to learn early enough about future climate risks in order to put in place action plans to minimize potential damage. For this, they usually consult the weather forecasts several times a day. The forecasting providers have limited reliability, thus farmers consult 3, 4, sometimes more, sources of weather forecasts and arbitrate between these sources in a subjective way. Indeed, these sources do not necessarily always perform in the same way. Some providers of weather forecasts will sometimes over-perform by providing forecasts close to the values really observed, and others under-perform with forecasts further away from the values actually observed. The accuracy of the weather forecasts of each supplier may vary according to the period, the type of climate or the geographical area. Given the variability in the

reliability of weather forecasts and the multiple prediction providers available to farmers, their decision-making is not facilitated and therefore often remains unclear. One of the possibilities for improving farmer's decision-making is providing them a single source of forecasts that outperforms those they have. The latter should allow farmers to consult only one source of forecasts and thus no longer have to arbitrate between the various providers of usual forecasts. The ideal range of forecasts allowing farmers to prepare for certain climate events is 1 to 12 hours.

In our previous work, (Guisiano et al., 2020), we proposed a Multi-Layered Ensemble Learning (MLEL) for short-term weather prediction. MLEL goal is to provide more reliable forecasts than weather forecast providers for the next 1 to 12 hours. It has been tested for 2 agricultural sites a few kilometers apart. However, when amount of data may increase to train MLEL, the number of layers may need to be expanded to gain better results. In case of increase in amount data and MLEL layers and using combination of complex models in each layer of it, it is necessary for computer to have enormous computing processing power to run MLEL. This paper introduces OptMLEL which extends MLEL, with improvements in the execution time, scalability and accuracy of the model by using new features and methods.

The rest of the paper is organized as follows: Section 2 describes related work. Section 3 introduces the proposed model in our previous work. Section 4 and 5 describe the automation and feature engineering in OptMLEL. Finally, section 6 demonstrates the experiment setup as well as performance of OptMLEL and last section concludes the paper.

2 RELATED WORK

There are machine learning techniques such as recurrent neural networks (Zaytar and Amrani, 2016) and Bayesian networks (Cofiño et al., 2002) used for short-term forecasting of climatic variables. It is also possible to use clustering techniques such as an Enhanced K-nearest neighbor (Sharif and Burn, 2007). In addition, different methods that can predict the probabilities of meteorological variables such as the Numerical Weather Prediction (NWP) using a great computational power and the Deep Hybrid Model for Weather Prediction using less computational power were introduced in our previous work. They provide different differences between the predicted temperatures and the observed temperatures. However, one degree difference between the predicted temperatures and the observed temperatures can be decisive for decision making. Our method in (Guisiano et al., 2020), via an innovative architecture, tried to minimize the difference between the predicted temperatures and the observed temperatures. In this paper we improved MLEL by automating and optimizing the model.

3 Model presentation

Our model, represented by the figure 1, is mainly inspired by the stacking technique. Indeed, each layer of the model is composed of a set of heterogeneous weak learners R_k, where $k = 1, \ldots, 3$. We find in each layer a L_k set of different supervised learning algorithms such as Multilayer Perceptron (MLP), Random Forest (RF) and Gradient Boosting Regressor (GBR).

Definition of the input/output data format as well as each layers details are explained in our previous work (Guisiano et al., 2020).

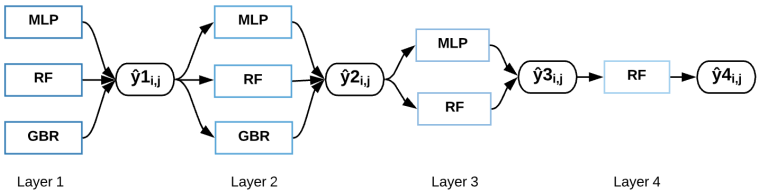

FIG. 1: Architecture of the Multi-Layered Ensemble Learning model.

4 Automation

MLEL includes data processing, training the model and prediction. This section explains how each of these steps are automated in OptMLEL.

4.1 The automation of data processing

In OptMLEL, the automation of data processing can be divided into three parts as follows: getting the data from API, removing redundant data and formatting into ideal csv format.

4.1.1 Get data from API

To get the data, three parameters are set at the beginning: Station or Point of Interest(POI), start time (date) and stop time (date).

There is a need of determining the forecast period as well. In our case, the forecast period is 6 hours for each provider. So the time at 1 a.m,7 a.m,13 a.m, 19 a.m is chosen to get the forecast data of each provider for the future 6 hours. For example, at 1 a.m, we collect the forecast data from 1 a.m. to 6 a.m.

4.1.2 Remove redundant data

Some providers have more than 6 hours' forecast data, and it is limited into 6 hours to make sure that our forecast data is the latest.

In addition, sometimes some weather providers cannot predict the data, and blank lines will appear on the data table. To make sure that the weather dataset are supplemented with new and complete data, the forecast data of all providers in a certain period is averaged or taken the median. This matter is explained in the section 4.1.3.

4.1.3 Turn data format into mean/median/all individual.

To automate the entire training process, there should not be any time gap in all data. However, due to various reasons, the free weather providers often have missing data, which leads to the obstruction of the code automation process. As mentioned above, in order to avoid data loss, the following data formats have been adopted to solve this problem.

— mean format (take average value for every hours' prediction data)
— median format (take median value for every hours' prediction data)
— The best provider (select the best provider's data for every hours' prediction data)

— All combine data of every individual provider (for comparison and test)
After testing different methods, results obtained from mean had the best results.

4.2 The automation of Machine Learning model

The automation of Machine Learning model includes the automation of tuning the model and layer adaption. Auto-selection of the best parameters and training the model in every layer is achieved using GridSearchCV in this project. Moreover, regarding the layer adaption, the amount of data is considered. The amount of data determines the accuracy of the predictions. We conducted different training methods for different amounts of data.

If we do not have a lot of data, we select the third layer test results and in the case of increase in the amount of data, another layer including RF and MLP is added.

5 Feature Engineering

New features added to MLEL are listed in equations 1-8.

Observed Humidity/Temperature relative difference: This is the relative humidity or temperature difference of current observed value from the last hour observed value.

$$Observed\_Temperature\_relative\_difference = observed\_temp(t) - observed\_temp(t-1) \quad (1)$$

$$Observed\_humidity\_relative\_difference = observed\_humidity(t) - observed\_humidity(t-1) \quad (2)$$

Humidity/Temperature relative difference: This is the relative humidity or temperature difference of current prediction from the last hour prediction of providers.

$$Temperature\_relative\_difference = temp(t) - temp(t-1) \quad (3)$$

$$Humidity\_relative\_difference = humidity(t) - humidity(t-1) \quad (4)$$

Humidity/Temperature absolute squared error: absolute squared error of the individual provider prediction with observed values.

$$Temperature\_absolute\_squared\_error = (temp(t) - observed\_temp(t))^2 \quad (5)$$

$$Humidity\_absolute\_squarederror = (humidity(t) - observed\_humidity(t))^2 \quad (6)$$

Humidity/Temperature tendency error (dynamicity): Tendency error is the difference of relative difference with respect to the relative difference of observed temperature/humidity.

$$Temperature\_tendency\_error = (temperature\_relative\_diff(t) - observed\_Temp\_relative\_diff(t))^2 \quad (7)$$

$$Humidity\_tendency\_error = (humidity\_relative\_diff(t) - observed\_humidity\_relative\_diff(t))^2 \quad (8)$$

6 Performance Evaluation

In this section, the trained data, validation methods and results are explained.

6.1 Data and Validation Methods

Our dataset focuses on a small area located in France, where we have 2 weather Stations. One of these Stations (Station 1) was used in our previous work and we will have some new tests still on those data in this paper. Regarding the free weather forecast providers, we have 4 providers that have forecasts for each Station ranging from 1 to 3 hours depending on the supplier (aerisweather, [1], Forecast.io [2], METEO-CONCEPT [3], and OpenWeatherMap [4]). The data from stations and the forecasters are from February till April. The tested results are validated using 3 validation periods considering 6 sequentially data points for each period.

6.2 Results

6.2.1 Comparison of program running time

To have an idea about effect of Spark on the execution time, Table 1 shows the running time of MLEL under the distributed system (i.e., spark) and traditional machine learning framework. (The training data comes from Station 1 and contains temperature, humidity, and wind speed data provided by all weather forecasters mentioned in (Guisiano et al., 2020).)

	Layer1		Layer2		Layer3	
	MLEL	MLELSpark	MLEL	MLELSpark	MLEL	MLELSpark
RF	3	13	2	10	1	10
MLP	572	14	40	13	47	13
GBR	887	21	736	19		

TAB. 1: Execution time for every layer

It shows that the distributed system based on the spark framework can greatly shorten the running time of the entire code. The total time to run all the program for MLEL is 2288 seconds and 113 second for MLEL in spark.

6.2.2 RMSE error

Using the data from New Station, the RMSE result for individual forecast providers is shown in Table 2. RMSE for OptMLEL is shown in Table 3 for the last layer of the model using mean methods. As the results show, OptMLEL is providing better results in comparison to the individual providers.

	Aeris	Forecast.io	METEO-CONCEPT	OpenWeatherMap
RMSE	2.28	1.70	1.97	2.01

TAB. 2: RMSE for forecasters

1. https://www.aerisweather.com/
2. Forecast.io
3. https://www.meteo-concept.fr/
4. https://openweathermap.org/

	val1	val2	val3
RF	0.34	0.39	0.34
MLP	0.25	0.32	0.30

TAB. 3: New Station, RMSE for mean method

7 Conclusion

In this paper, we successfully combined machine learning and big data closely, and used Spark, a distributed processing tool for big data, to significantly increase the speed of the our previous proposed previous work (i.e., MLEL) for short-term forecasting to assist farmers in decision-making. On the other hand, we put free weather forecast data into the machine learning algorithm, and accurately obtained more accurate forecast results than other free weather forecasts. We believe the short-term forecasting system will provide more accurate weather forecast information for agricultural production.

References

Cofiño, A. S., R. Cano, C. Sordo, and J. M. Gutiérrez (2002). Bayesian networks for probabilistic weather prediction. In *In Proceedings of the 15th European Conference on Artificial Intelligence (ECAI)*, pp. 695–700.

Guisiano, J. E., R. Chiky, S. Ahvar, and J. Orensanz (2020). Multilayered ensemble learning for shortterm forecasting in agroclimatology. In *6th International Conference on Computer and Technology Applications, Turkey*.

Sharif, M. and D. Burn (2007). Improved k-nearest neighbor weather generating model. *Journal of Hydrologic Engineering - J HYDROL ENG 12*.

Zaytar, M. A. and C. E. Amrani (2016). Sequence to sequence weather forecasting with long short-term memory recurrent neural networks.

Résumé

L'agriculture déépend fortement des prévisions météorologiques. En effet, afin d'optimiser leur production, les agriculteurs doivent être capables d'anticiper des conditions climatiques favorables ou non à leurs activités en déployant les plans d'actions appropriés. Pour cela, ils consultent quotidiennement les données de différents fournisseurs de prévisions météorologiques. Cependant, la fiabilité des prévisions de chaque fournisseur est variable selon la période, le climat ou la zone géographique. Les agriculteurs doivent donc arbitrer quotidiennement entre les différents fournisseurs. Cet article propose une version optimisée de MLEL (une approche de prédiction agro-climatique à court terme) en utilisant SPARK, nous l'avons appelé OptMLEL. OptMLEL permet de distribuer les traitements sur plusieurs noeuds de calcul. Elle permet également d'ajuster automatiquement les paramètres de l'algorithme en cas d'ajout ou suppression d'un fournisseur météo. Enfin, elle sélectionne automatiquement le nombre de

couches en fonction de la quantité de données disponibles en entrée. Les résultats des experimentations montrent l'applicabilité et les bonnes performances de notre approche