Adopting and Embedding Machine Learning Algorithms in Microcontroller for Weather Prediction

Petros Karvelis
Computer Technology Institute & Press "Diophantus"
Patras, Greece
pkarvelis@gmail.com

Theofanis-Aristofanis Michail
Computer Technology Institute & Press "Diophantus"
Patras, Greece
michail@westgate.gr

Daniele Mazzei
Computer Science Department,
University of Pisa,
Pisa, Italy
d.mazzei@zerynth.com

Stefanos Petsios
Computer Technology Institute & Press "Diophantus"
Patras, Greece
stefanos.petsios@gmail.com

Andrea Bau, Gabriele Montelisciani
TOI srl
Pisa, Italy
a.bau@zerynth.com,
g.montelisciani@zerynth.com

Chrysostomos Stylios
Computer Technology Institute & Press "Diophantus"
Patras, Greece
stylios@cti.gr

***Abstract*—Maritime journeys are significantly depending on weather conditions and so meteorology have ever had a key role in maritime businesses. Nowadays, the new era of innovative machine learning approaches along with the availability of a wide range of sensors and microcontrollers, creates increasing perspectives for providing onboard reliable short-range forecasting of main meteorological variables. The main goal of the current study is to propose a machine learning algorithm, which will be coded into a microcontroller and will be able to predict in short-term the wind speed weather conditions on board of the boat. A regression machine learning algorithm was chosen so that to require the smallest amount of resources (memory, CPU) and to be able to run in a microcontroller. The method was coded suing a powerful programming platform for microcontrollers namely the Zerynth studio. The proposed method was tested on real weather data recorded during a ship journey and its efficiency is proven based on a number of error metrics.**

Keywords—weather forecasting, Zerynth, machine learning, microcontroller.

I. Introduction

Meteorology is the study of weather events, having as major focus the forecasting of main weather variables. Prediction on weather conditions is performing either on historical data or mathematical models or a combination of them. There is great importance on quantifying meteorological variables such as temperature, humidity, air pressure, wind flow; their variations and interactions and how the change over time. There are proposed various spatial scales in order to describe and predict weather conditions on local, regional or global levels. On the other hand, weather conditions have had great effect on sea-transportation and all the maritime affairs.

Maritime activities are related to maritime commerce and maritime leisure but they also affect environment and consequently atmosphere and so the weather conditions. All are related and interconnected and seem to belong to a vicious circle that impact the quality of environment and the meteorological maritime conditions [1]-[6]. The safety of the maritime transportation is strictly related to weather information. International Maritime Organization (IMO) and World Meteorological Organization (WMO) have proposed and define regulation strategies on how to provide marine weather predictions so that to minimize accidents and economic losses. There is an essential need for maritime weather prediction, which is connected to sustainable development of the sea commerce, requiring developing of early warning systems and automatic real-time information exchange in order to achieve high weather forecast quality (especially about storms, heavy precipitation, wind, waves and extreme temperatures).

When it comes to forecasting meteorological parameters like wind speed special care must be taken in order to tackle the problem of the local environment and the complexity of these parameters which can seriously affect the performance of the forecasting methods. Hence, it is a common approach to test a number of machine learning techniques to solve the weather forecasting problem [7]-[8]. Such weather forecasting techniques either depend on the utilization of generative numerical methods [9], or use time-series analysis such as Auto Regressive Integrated Moving Average (ARIMA) models [10] or finally use advanced machine learning algorithms that are based on Artificial Neural Networks (ANNs) [11], [12] or Support Vector Machines [14], [15]. Nowadays the effective use of approaches relying in Deep Learning for a number of domains have persuaded its utilization for climate and weather prediction [16].

The paper is structured in the following way: Section II begins to describe the Zerynth platform that was used to write the code for the microcontroller and Section III presents the regression algorithm. Section IV, presents some technical details of the microcontroller used and Section V and VI describe the data used and the results achieved of our method respectively. Finally, Section VII concludes the paper with some final remarks, recommendations and suggestions for future work.

978-1-5386-7097-2/18/$31.00 ©2018 IEEE

II. Zerynth Platform

Zerynth [17] is a software suite that provides to the developers with a number of tools in order to program 32bit microcontrollers using Python or hybrid C/Python. Zerynth has a very light footprint while guaranteeing real-time performances thanks to the RTOS (e.i. FreeRTOS or Chibios can be chosen). The main component of the Zerynth suite is the Zerynth Virtual Machine (VM), which is a real time multithreaded Operating System that provides hardware abstraction making any Zerynth project hardware independent. The Virtual Machine provides high-level features of Python like the following: classes, multi-threading, modules, callback, timers and finally exceptions. It supports the use of hardware features e.g. PWM, interrupts and digital Input - Output. In Zerynth, threads are written in C programming language and can live along the VM enabling a high-performance mixed C/Python real-time environment.

Zerynth also includes a dedicated development environment. Zerynth Studio which is a free, cross-platform Integrated Development Environment and a Toolchain for developing hybrid C or Python applications and managing Zerynth supported boards. It includes a compiler, debugger and an editor, alongside tutorials and example projects for an easy learning experience. The Zerynth ADM (Advanced Device Manager) is a cloud service that enables a fast and easy to use cloud provisioning of Zerynth powered devices while also allowing Firmware Over-The-Air (FOTA) updates and Remote Procedure Calls. Finally, the Zerynth App is a general purpose Android/iOS mobile app that allow an easy building of HTML/JS Graphical User Interfaces for Zerynth powered devices.

There are various hardware abstraction layer libraries, RTOSs and interpreted language engines for microcontroller available on the market. However, considering the need of developing a Machine Learning model for weather prediction we preferred to choice a complete and reliable developing framework able to support Python and in particular hybrid C/Python coding. In this context, a valid alternative to Zerynth is Micropython. Micropython is a Python interpreter executable by MCUs that allows a quick and easy development of embedded applications in Python language. Micropython scripts can be directly written on the board's serial port terminal giving developers the possibility to easily test embedded devices features and peripherals from a serial port emulated terminal console. However, having the entire interpreter executed by the microcontroller dramatically increase the footprint of the OS reducing the RAM and Flash memory available for the ML application. Moreover, Micropython doesn’t include any ready to use service for FOTA and RPC that are both required in order to allow an easy debug and upgrade of already installed remote devices.

III. Regression Methods

Regression is a type of problem when a prediction is needed for some output from a number of inputs using a number of data. In order to establish notation we will use $x \in \Re^{m}$, to describe the input features, and $y \in R$ to denote the target value of a parameter. Thus $\{x, y\}$ in our case will be the training data and the list of the training data it is often called the dataset. The dataset $\{x_i, y_i \mid i = 1, \ldots, n\}$ consists of a number n of training data where $\Re^{n \times m}$ denotes the input space, and $\Re^{n}$ the output space of n values.

Thus, the regression problem is the task of approximating a function like $X^{n\times m} \rightarrow Y^{n\times 1}$, for which we minimize error between the real wind speed value and the predicted wind speed value. In this study, the available features are: Temperature, Dew point temperature, Humidity, Wind direction, Pressure, Precipitation and Wind Speed. The goal of this study is to predict the value of the Wind speed parameter.

One of the simpler algorithms that can be used is the Linear Regression (LR) [18], which seeks to predict the wind speed as a linear combination of the features. In order to approximate y (wind speed) as a linear function of x_i:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_m x_m = \sum_{i=1}^{m} \theta_i x_i \,, \tag{1}$$

where $x_0 = 1$ and $\boldsymbol{\theta} = \{\theta_i \mid \boldsymbol{i} = 1, \ldots, \mathrm{m}\}$ are the parameters of the linear function mapping from $X^{n\times m}$ to $Y^{n\times 1}$.

A most common error function between the output $h_\theta(x)$ and the actual value y_i is the least square error function defined in terms of:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} \left(h_\theta(x_i) - y_i \right)^2 , \tag{2}$$

The previous equation (2) is the familiar least-squares cost function which defines ordinary least squares regression model. When it comes to minimize the error $J(\theta)$ a most common algorithm would be the Gradient Descent (GD) algorithm which start from some “initial guess” for the parameters $\boldsymbol{\theta}$ and repeatedly changes the values of the parameters in order to make $J(\theta)$ smaller, until converge to minimum value for $J(\theta)$. This is an algorithm that repeatedly takes a step in the direction of steepest decrease of $J(\theta)$. In detail, the GD algorithm performs the following update on the parameters $\boldsymbol{\theta}$:

$$\theta_j^{new} = \theta_j^{old} - a \frac{\partial J(\theta)}{\partial \theta_j} , \tag{3}$$

where a the learning rate which is usually set to $10^{-3} \le a \le 10^{-2}$ and smaller values for the learning rate leads to slower convergence.

The previous update is simultaneously performed for all values of $j = 1, \ldots, m$. However, when it comes to program embedded environments, where processing cycles, memory and power are all in short supply is really hard. Thus deploying machine learning algorithms for regression could be considered extremely difficult and one has to keep in mind some basic limitations that had to be overcome a) CPU limitations: the low operating frequency of the microcontrollers, and b) Memory limitations: the small memory capacity of such type of microcontrollers.

In order to overcome the above two limitations for regression problems one can choose to update the parameters $\boldsymbol{\theta}$ as the new data are collected from the weather station. This type of method is called Stochastic Gradient Descent (SGD) [19], [20] and is currently gaining interest in Deep Learning applications [21]. For example SGD has been used in the case of large volume datasets [22] instead of using the costly GD since one has to do a single step for one pass for the whole training set. Furthermore, the larger the training set, the slower the GD algorithm will update the parameters since the longer it may take until it converges to the global minimum.

Stochastic Gradient Descent (SGD) simply computes the gradient of the parameters using only a single or a few training examples thus the new update is given by:

$$\theta_j^{new} = \theta_j^{old} - a\frac{\partial J(\theta; x_i . y_i)}{\partial \theta_j}, \tag{4}$$

for a pair $(x_i . y_i)$ from the training set. In SGD the learning rate a is typically much smaller than a corresponding learning rate in batch gradient descent because there is much more variance in the update.

IV. Microcontroller - ESP-32

The used microcontroller was the ESP-32 DevKitC. the official development board is displayed in the figure below and a technical description of the microcontroller is shown in the following Table.

Fig. 1. The microcontroller ESP8266 ESP-32 μC.

TABLE I. ESP32 SPECIFICATIONS

Microcontroller:	**Tensilica 32-bit Single-/Dual-core CPU Xtensa LX6**
Operating Voltage	3.3V
Power Supply Voltage:	7-12V
Digital I/O Pins (DIO)	28
Analog Input Pins (ADC)	8
UARTs	3
SPIs	2
I2Cs	3
Flash Memory	4MB
SRAM	520 KB
Wi-Fi and Bluetooth	IEEE 802.11 b/g/n/e/i

V. Data

The data was collected using a Tugboat (Wikipedia, Tugboat) which is a type of vessel that maneuvers other vessels by pushing or pulling them either by direct contact or by means of a tow line. The trip of the tugboat was from Chios island (Greece) to Athens (Greece) and the data collection lasted from 2018-01-22 11:40:00 to 2018-02-09 12:50:00. The planned trip of the ship is displayed in the figure below:

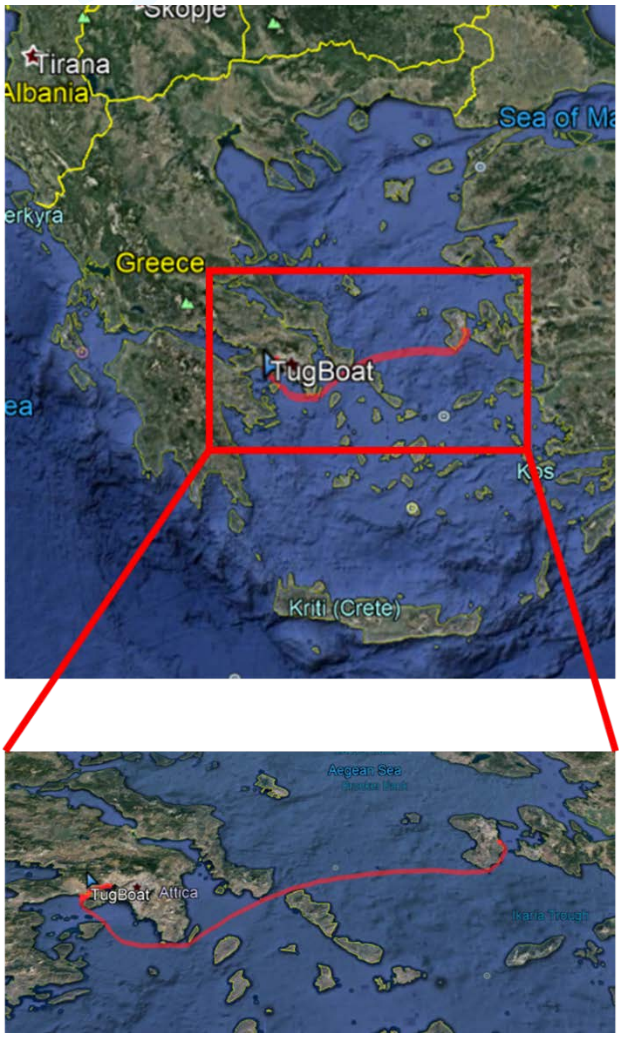

Fig. 2. The recorded trip of the tugboat from Chios Island, Greece to Athens, Greece.

On board of the tugboat we have installed a weather station (Airmar 220WX, Airmar Technology) which send the data to a data logger using the NMEA 2000 channel [23]. During the trip of the tug boat the following weather parameters were recorded Temperature, Dew point temperature, Humidity, Wind direction, Pressure, Precipitation and Wind Speed with a 15 minutes interval. The technical details and specifications of the weather station are presented in the following table.

TABLE II. WEATHER STATION BASIC CHARACTERISTICS.

Airmar 220WX	
Operating voltage range of 12-24 DC.	Waterproof.
Meteorological Parameters: Air Temperature, True/Apparent Wind Speed, True/Apparent Wind Direction, Barometric Pressure.	Three-axis solid-state compass with dynamic stabilization (better than 1° static compass accuracy).
Three-axis accelerometer for pitch and roll.	Three-axis rate gyros provide rate-of-turn data.
Internal GPS.	Current Time.

As far as the meteorological parameters that were measured from the Weather Station a detailed description can be displayed in the following Table.

TABLE III. DESCRIPTION OF THE THE METEOROLOGICAL PARAMETERS RECORDED.

Measurement	Unit of measure	Typical operation range
True Wind Speed	m/s	[0, 60] m/s
True Wind Direction	deg	[0, 360°]
Air temperature	ºC	[-30, +60°C]
Barometric Pressure	mbar	[540, 1100] hPa
Relative humidity	%	[0 %, 100 %]

VI. RESULTS

In the current study, we partitioned the dataset to an initial training and the rest of the dataset was used as a testing set. The initial training set was composed of 4 days of the trip of the tugboat and the rest of the data for testing. In order to evaluate our method, we have used the Median Absolute Error (MAE) and the Root Mean Square Error ($RMSE$) which are defined below as [24], [25]:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i - y_i\right|, \quad (4)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2}, \quad (5)$$

where N is the number of the testing samples, $\hat{y}_i$ is the predicted value and y_i the true wind speed value.

Finally, in Table IV we present our results with respect to the MAE and $RMSE$ error rates.

TABLE IV. WEATHER STATION BASIC CHARACTERISTICS.

	MAE	*RMSE*
Our Method	1.591	2.165

In the following figure we demonstrate that the resulting forecasting model that was derived from the platform achieve a high accuracy at predicting future wind speed in real time situations.

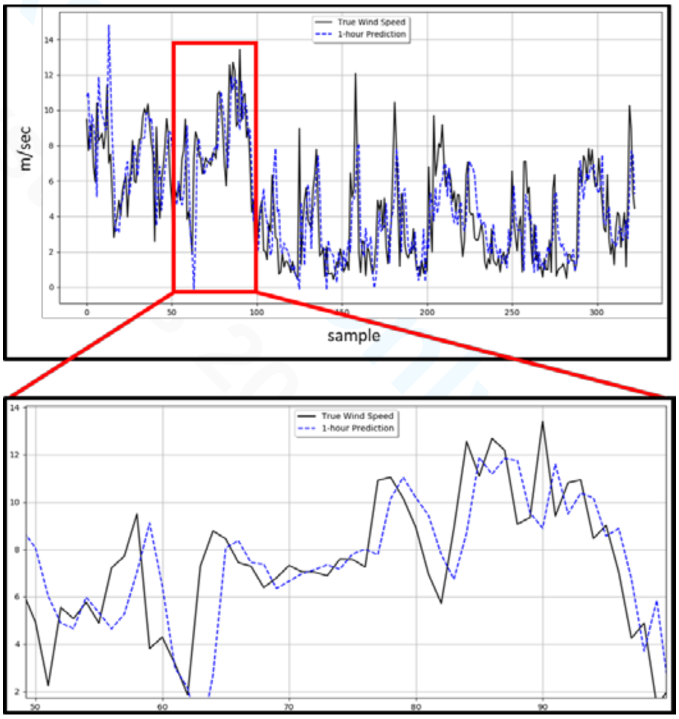

Fig. 3. Performance of our model over test data. Wind speed predictions for 1 hour.

VII. CONCLUSIONS

The aim of this study was to predict the wind speed on short-term weather conditions using an on-board vessels weather station. We have used the Zerynth platform and the ESP-32 DevKitC microcontroller in order to code the regression method. The efficiency of our method is proven using real weather data recorded from a ship. As for future work we are going to test our method using more data from the tugboat.

ACKNOWLEDGMENT

This study is supported by LINCOLN” (Lean Innovative Connected Vessels) Project (www.lincolnproject.eu) Horizon

2020 research and innovation program (Grant Agreement: 727982).

REFERENCES

[1] S. Kolios, A. Vorobev, G. Vorobeva and C. Stylios, GIS and Environmental Monitoring: Application in the Marine, Atmospheric and Geomagnetic Fields, Springer ISBN 978-3-319-53084-0 (2017).

[2] A. K. Gupta, S. K. Gupta and R. Patil, “Environmental management plan for port and harbour projects,” Clean Technologies and Environmental Policy, vol.7(2), pp. 133–141, 2005.

[3] V. Eyring, , I. S. A., T. IsaksenBerntsen, W. J. Collins, J. J. Corbett, O. Endresen, R. G. Grainger, J. Moldanova, H. Schlager and D. S. Stevenson, “Transport impacts on atmosphere and climate: Shipping,” Atmospheric Environment, vol. 44(37), pp. 4735–4771, 2010.

[4] M. Grifoll, G. Jordà, M. Espino, J. Romo and M. García–Sotillo, “A management system for accidental water pollution risk in a harbour: the Barcelona case study,” Journal of Marine Systems, vol. 88(1), pp. 60–73, 2011.

[5] S. Kolios, P. Karvelis, C.D. Stylios and P. Tagkas, “Cloud top Height Estimation from the Meteosat Water Vapor Imagery Using Computational Intelligence Techniques: Preliminary Results,” In: *Proceedings of the 5th International Conference on Advances in Satellite and Space Communications SPACOMM 2013*, 21-26 April 2013, Venice, Italy, pp. 65-70, (2013).

[6] J. Dinwoodie, S. Tuck, H. Knowles, J. Benhin, and M. Sansom, “Sustainable development of maritime operations in ports,” Business Strategy and the Environment, vol. 21(2), 111–126, 2012.

[7] P. Karvelis, S. Kolios, G. Georgoulas, and C. D. Stylios, “Ensemble Learning for Forecasting Main Meteorological Parameters,” In: Proceedings of 2017 IEEE International Conference on Systems, Man and Cybernetics (SMC)), 5-8 October 2017, Banff Center, Banff, Canada, pp. 3711-3714 (2017).

[8] G. Georgoulas, P. Karvelis, S. Kolios and C. D. Stylios, “Examining nominal and ordinal classifiers for forecasting wind speed,” In: Proceedings of 2016 IEEE 8th International Conference on Intelligent Systems (IS'16), 3-5 September 2016, Sofia, Bulgaria. pp. 504-509 (2016).

[9] G. Marchuk. Numerical methods in weather prediction. Elsevier, 2012.

[10] H. Saima, J. Jaafar, S. Belhaouari, T. Perak, and T.A. Jillani, “Intelligent methods for weather forecasting: A review,” In Proceedings of the National Postgraduate Conference, (Tronoh Perak, Malaysia) Sept. 19-20, 2011.

[11] V. Krasnopolsky, and M. FoxRabinovitz. “Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction.” Neural Networks, vol. 19, pp. 122-134 , (2006):.

[12] Lai, Loi Lei, et al. ”Intelligent weather forecast.”Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on. Vol. 7. IEEE, 2004.

[13] Y. Radhika and M. Shashi “Atmospheric temperature prediction using support vector machines,” International Journal of Computer Theory and Engineering, vol. 1, pp. 1793–8201, 2009.

[14] N. I. Sapankevych and R. Sankar “Time series prediction using support vector machines: a survey,” Computational Intelligence Magazine, IEEE, vol. 4, pp. 24–38, 2009.

[15] Y. Radhika and M. Shashi. Atmospheric temperature prediction using support vector machines. International Journal of Computer Theory and Engineering, 1(1):1793–8201, 2009.

[16] J. N. K Liu, Y. Hu, Y. He, P. W. Chan and I.. Lai. “Deep Neural Network Modeling for Big Data Weather Forecasting,” Information Granularity, Big Data, and Computational Intelligence. Springer International Publishing, pp. 389-408, 2015.

[17] D. Mazzei, G. Baldi, G. Montelisciani and G. Fantoni, “A full stack for quick prototyping of IoT solutions,” Annals of Telecommunication, springer, pp1-11, 2018.

[18] G. Seber and A. Lee, Linear Regression Analysis, Whilley, 2003.

[19] Léon Bottou and Yann LeCun: Large Scale Online Learning, Advances in Neural Information Processing Systems 16, 2016.

[20] T. Zheng, “Solving large scale linear prediction problems using stochastic gradient descent algorithms,” Proceedings of the Twenty-First International Cconference on Machine Learning (ICML), 2004.

[21] G. Cong, O. Bhardwaj and M. Feng, "An Efficient, Distributed Stochastic Gradient Descent Algorithm for Deep-Learning Applications," 2017 46th International Conference on Parallel Processing (ICPP), Bristol, pp. 11-20, 2017.

[22] Kaggle, https://www.kaggle.com, (accessed July 2018)

[23] NMEA 2000, https://www.nmea.org, (accessed July 2018).

[24] A. Saxena, J. Celaya, B. Saha,S. Saha and K. Goebel, “Metrics for offlineevaluation of prognostic performance,” International Journal of Prognostics and Health Management, vol. 1, no. 1, pp.4-23, 2010.

[25] S. Makridakis, S. Wheelwright and R. Hyndman, Forecasting Methods and Applications, 3rd Edition, Wiley, 2012.