



# Reducing uncertainty in environmental measurements using bayesian and adaptive moment estimation : study case Andean city of Quito

Ricardo Xavier Llugsi

## ► To cite this version:

Ricardo Xavier Llugsi. Reducing uncertainty in environmental measurements using bayesian and adaptive moment estimation : study case Andean city of Quito. Computation [stat.CO]. Université de Perpignan, 2023. English. NNT : 2023PERP0017 . tel-04215989

**HAL Id: tel-04215989**

**<https://theses.hal.science/tel-04215989>**

Submitted on 23 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Délivrée par  
**Université Perpignan Via Domitia**

Préparée au sein de l'école doctorale Énergie Environnement

Et de l'unité de recherche Espace Dev

Spécialité : **Sciences de l'Ingénieur**

Présentée par : **M. Ricardo Xavier LLUGSI CAÑAR**

**REDUCING UNCERTAINTY IN ENVIRONMENTAL  
MEASUREMENTS USING BAYESIAN AND  
ADAPTIVE MOMENT ESTIMATION  
STUDY CASE: ANDEAN CITY OF QUITO**

Soutenue le 17 juin 2023 devant le jury composé de

M. S. BEN YAHIA	PR, Tallinn University of Technology	Rapporteur
M. A. ZEMMARI	PR, Université de Bordeaux	Rapporteur
Mme. A. MOUAKHER	MCF, Université de Perpignan	Examinatrice
M. T. TALBERT	MCF, Université de Perpignan	Examineur
Mme A. LAURENT	PR, Université de Montpellier	Présidente
Mme S. EL YACOUBI	PR, Université de Perpignan	Directrice de thèse
Mme A. FONTAINE	MCF, Université de Guyane	Co-Directrice de thèse
M. P. LUPERA	PR, National Polytechnic School	Co-Directeur de thèse

2023-06-01

# Contents

<b>List of Figures</b>	<b>IV</b>
<b>List of Tables</b>	<b>V</b>
<b>List of Acronyms</b>	<b>VI</b>
<b>Abstract</b>	<b>1</b>
<b>Résumé</b>	<b>1</b>
<b>1 Introduction</b>	<b>4</b>
1.1 Motivation . . . . .	5
1.2 State of the Art . . . . .	6
1.3 Context . . . . .	9
1.4 Contribution . . . . .	10
1.5 Content . . . . .	12
<b>2 Generalities of Neural Networks &amp; Tools</b>	<b>14</b>
2.1 History . . . . .	14
2.2 Neural Network Operation . . . . .	15
2.3 Types of Neural Networks . . . . .	20
2.3.1 Recurrent Neural Networks . . . . .	20
2.3.2 Long short-term Memory (LSTM) . . . . .	21
2.3.3 Gated Recurring Units (GRU) . . . . .	23
2.3.4 Stacked LSTM Neural Networks . . . . .	24
2.3.5 Auto-regressive integrated moving average network (ARIMA) . . . . .	25
2.3.6 Convolutional encoder–decoder network . . . . .	26
2.4 Validation . . . . .	28
2.4.1 K-Fold Cross validation . . . . .	28
2.4.2 Walk-Forward Validation . . . . .	29
2.5 Optimizers . . . . .	29
2.5.1 Stochastic Gradient Descent (SGD) . . . . .	30
2.5.2 Adaptative Gradient Algorithm (ADAGRAD) . . . . .	31
2.5.3 ADAGRAD based on a moving window of gradient updates (ADADELTA) . . . . .	31
2.5.4 Root Mean Square Propagation (RMSprop) . . . . .	31
2.5.5 Adaptive Moment Estimation (ADAM) . . . . .	31
2.5.6 Adaptive Moment Estimation with infinite normalization (ADAMAX)	33

2.5.7	Adaptive Moment Estimation with Decoupled Decay Regularization . . . . .	34
2.6	Tools . . . . .	35
2.6.1	Error Metrics . . . . .	35
2.6.2	Correlation . . . . .	37
2.6.3	Dynamic Time Warping . . . . .	38
2.6.4	Methodology . . . . .	39
<b>3</b>	<b>Challenges of the Weather of Quito</b>	<b>40</b>
3.1	Quito Weather . . . . .	40
3.2	Distribution of the Stations . . . . .	41
3.3	Forecast Methodology . . . . .	44
3.4	Experimentation . . . . .	44
3.4.1	Discussion . . . . .	46
<b>4</b>	<b>The Bayesian approach and the Adaptive Moment Estimation to reduce uncertainty</b>	<b>49</b>
4.1	Definition of Uncertainty . . . . .	49
4.2	Bayesian Inference . . . . .	50
4.2.1	Uncertainty in Neural Networks and Dropout Technique . . .	52
4.2.2	Estimators for Uncertainty . . . . .	54
4.2.3	Experimentation . . . . .	54
4.2.4	Results from the experiments . . . . .	55
4.2.5	Discussion . . . . .	57
4.3	Adaptive Moment Estimation to reduce uncertainty . . . . .	57
4.3.1	The optimization process . . . . .	58
4.3.2	Weight Decay . . . . .	60
4.3.3	ADAML . . . . .	60
4.3.4	Proof of ADAML's convergence . . . . .	61
4.3.5	Experimentation . . . . .	72
4.3.6	Discussion of Results . . . . .	73
<b>5</b>	<b>Error detection and adjustment approach</b>	<b>74</b>
5.1	Error Detection . . . . .	74
5.1.1	Correlation analysis of the stations of the network . . . . .	74
5.1.2	24-Hour Forecast . . . . .	77
5.1.3	Discussion of results . . . . .	82
5.2	Auto-Adjustment . . . . .	84
5.2.1	Auto-Adjustment Process . . . . .	84
5.2.2	Relationship between series . . . . .	85

5.2.3	Discussion of results . . . . .	86
<b>6</b>	<b>General Conclusions and Future work</b>	<b>88</b>
6.1	Conclusions . . . . .	88
6.2	Limitations and future work . . . . .	90
	<b>References</b>	<b>92</b>
	<b>Appendix</b>	<b>98</b>
A	Published Articles . . . . .	98
B	Calibration reports . . . . .	98

## List of Figures

Figure 1:	Forecast Chain. . . . .	6
Figure 2:	Variations of the weather in Quito depending on the geography. . .	11
Figure 3:	Neural Network scheme. . . . .	15
Figure 4:	Activation functions. (a) Sigmoid function, (b) Tanh function, (c) ReLU function, (d) Leaky ReLU function, (e) Softmax function. . .	19
Figure 5:	Neural Network scheme. . . . .	20
Figure 6:	Structure of LSTM. . . . .	22
Figure 7:	Structure of GRU. . . . .	23
Figure 8:	Encoder-Decoder structure. . . . .	27
Figure 9:	SGD optimizer with (a) acceleration, (b) acceleration with Nesterov approach. . . . .	30
Figure 10:	(a) Euclidean Matching, (b) Dynamic Time Warping Matching. . .	38
Figure 11:	Geographic position of the AWS on a topographic map. Note: Node 2 was omitted because it was utilized for operational verification only. . .	41
Figure 12:	MAPE comparison for the Neural Network with LSTM structure (Scenario 1). (a) Without Bayesian Modelling, (b) With Bayesian Modelling. . . . .	56
Figure 13:	MAPE comparison for the Neural Network with LSTM structure (Scenario 2). (a) Without Bayesian Modelling, (b) With Bayesian Modelling. . . . .	57
Figure 14:	Comparison of error between models. . . . .	73
Figure 15:	Graphic analysis between AWS 1 and AWS 3 (Temperature). . . . .	76
Figure 16:	DTW comparison between AWS 1 & AWS 3. . . . .	77
Figure 17:	Comparison between real temperature and forecast. . . . .	79
Figure 18:	Comparison of forecasted temperature for the AWS Network. . . .	81
Figure 19:	Visualizing the distribution of observations in the dataset. . . . .	82
Figure 20:	Flow diagram of the auto-adjustment process. . . . .	85
Figure 21:	Result of the adjustment process for AWS 1. . . . .	86
Figure 22:	Result of the adjustment process for AWS 3. . . . .	87

## List of Tables

Table 1:	Geographical position of the AWS network. . . . .	43
Table 2:	Number of hidden units per layer (L1, L2 and L3) (I). . . . .	45
Table 3:	Number of hidden units per layer (L1, L2 and L3) (II). . . . .	46
Table 4:	Networks with best prediction accuracy (per error metric). . . . .	47
Table 5:	Parameters and forecast information for LSTM (without Bayesian Modelling) . . . . .	56
Table 6:	Exemplary Table . . . . .	56
Table 7:	Specifics of the Optimizer Algorithms ADAM and ADAML. . . . .	61
Table 8:	Networks with best prediction accuracy (per error metric). . . . .	73
Table 9:	Correlation Coefficient (r) and distance (d) in kilometres between the stations in the Network. . . . .	75
Table 10:	Correlation Coefficient per week. . . . .	76
Table 11:	Correlation Coefficient per week. . . . .	76
Table 12:	Error metrics and the associated Correlation for every station in the network. . . . .	78
Table 13:	Exemplary Table . . . . .	80
Table 14:	Determination of station neighbourhoods and thresholds based on correlation coefficients and MAE. . . . .	83
Table 15:	Exemplary Table . . . . .	84
Table 16:	Relationship between the stations . . . . .	86



## List of Acronyms

<b>AdaGrad</b>	Adaptive Gradient Algorithm
<b>Adaline</b>	Adaptative Linear Elements model
<b>ADAM</b>	Adaptive Moment Estimation
<b>ADAML</b>	ADAM Logger
<b>ALU</b>	Arithmetic Logic Unit
<b>AI</b>	Artificial Intelligence
<b>ANN</b>	Artificial Neural Network
<b>AWS</b>	Automatic Weather Stations
<b>AR</b>	Auto-Regressive
<b>ARIMA</b>	Auto-Regressive Integrated Moving Average
<b>BNN</b>	Bayesian Neural Network
<b>CPU</b>	Central Processing Unit
<b>DAC</b>	Civil Aviation Directorate
<b>CWS</b>	Conventional Weather Stations
<b>CNN</b>	Convolutional Neural Networks
<b>Ccal</b>	Correlation Coefficient Calculated
<b>CTyp</b>	Correlation Coefficient Typical
<b>k-fold cross validation</b>	Cross validation of k iterations
<b>DAC</b>	Civil Aviation Directorate
<b>DL</b>	Deep Learning
<b>DTW</b>	Dynamic Time Warping
<b>EWS</b>	Early Warning Systems
<b>EAP</b>	Economically Active Population
<b>FOSP</b>	First Order Stationary Points

<b>FBNNs</b>	Functional Variational Bayesian Neural Networks
<b>GOES</b>	Geostationary Operational Environmental Satellite
<b>GPU</b>	Graphics Processing Unit
<b>GRU</b>	Gated Recurring Units
<b>HMM</b>	Hidden Markov Mode
<b>KL</b>	Kullback-Leibler
<b>lr</b>	Learning rate
<b>LSTM</b>	Long Short Term Memory
<b>ML</b>	Machine Learning
<b>MAE</b>	Mean Absolute Error
<b>MAPE</b>	Mean Absolute Percentage Error
<b>MSE</b>	Mean Squared Error
<b>MASL</b>	Meters Above Sea Level
<b>MC</b>	Monte Carlo
<b>MA</b>	Moving Average
<b>INAMHI</b>	National Institute of Meteorology and Hydrology of Ecuador
<b>INEC</b>	National Institute of Statistics and Censuses of Ecuador
<b>NOAA</b>	National Oceanic and Atmospheric Administration
<b>NN</b>	Neural Network
<b>1D</b>	One-Dimensional
<b>OAQ</b>	Quito Astronomical Observatory
<b>RNN</b>	Recurrent Neural Networks
<b>ReLU</b>	Rectified Linear Unit
<b>RMSE</b>	Root Mean Squared Error
<b>RMSprop</b>	Root Mean Square Propagation

**SOSP** Second Order Stationary Points

**SUT** Station Under Test

**SGD** Stochastic Gradient Descent

**ZO-SGD** Stochastic ZO-GD

**VI** Variational Inference

**WMO** World Meteorological Organization

**ZO** Zero-Order Methods

**ZO-ADAMM** ZO adaptive momentum method

## Abstract

Given the unique topography of Quito, predicting climate change in this city is challenging. This thesis focuses on the study of meteorological data, specifically for the city of Quito. To achieve this goal, automated micro meteorological stations (AMS) were deployed at sites of interest and data was collected on an FTP server in the cloud using the available cellular network. The main objective of this study is to predict environmental parameters while identifying measurement errors in order to calibrate the stations and correct these errors.

In this thesis, we developed four different models to obtain accurate forecasts: an ARIMA (AutoRegressive Integrated Moving Average) model, an LSTM (Long Short-Term Memory) model, a stacked LSTM model, and a convolutional LSTM model with uncertainty error reduction. To detect errors in the automated micro meteorological stations (AMS), we use time series data from two highly correlated stations with the station under analysis to obtain a 24-hour forecast of the measured parameter (temperature in our experiments). This allows us to determine if the station under analysis is recording inaccurate measurements. To detect measurement errors, a comparison is iteratively performed with information from each micro station based on its neighboring stations. The difference between the initial correlation coefficients and those acquired at time  $t$  is calculated. If this difference exceeds a certain threshold, the algorithm signals an error and initiates the calculation of an adjustment for this error based on the calculated forecast for that station. Finally, we provide the sensor calibration equation parameters of the station using the proposed adjustment. To conduct this study, we applied various techniques, including correlation coefficient calculations, the use of a multilayer neural network, the design of a new version of the ADAM optimizer, and a Bayesian-based uncertainty reduction strategy.

## Résumé

Selon les statistiques générées par l'Institut national des statistiques et des recensements de l'Équateur (INEC), l'agriculture est l'un des points forts de l'économie équatorienne. En effet, en plus de produire la majorité des denrées alimentaires consommées localement et de permettre l'exportation de produits tels que les bananes, les brocolis et le cacao à l'international, on retrouve dans le secteur agricole de nombreux avantages sociaux. La zone de convergence intertropicale oscille autour de l'Équateur qui est lui-même traversé par la cordillère des Andes. Cette situation géographique joue sur le climat et affecte l'agriculture avec l'apparition de températures basses (autour de 0°C) et de températures élevées (autour de 36°C) dans des zones géographiques très proches les unes des autres. Le premier scénario donne lieu à l'apparition de

gelées agricoles qui, selon la physiologie de la plante, peuvent gravement affecter des cultures telles que la pomme de terre ou le maïs. Alors que le deuxième scénario indique une augmentation du phénomène d'évapotranspiration, ce qui implique un stress agricole sur les cultures et l'apparition de maladies telles que les tâches rouges ou la cercosporiose noire. À ce stade, il est pertinent de mentionner que le changement climatique frappant le monde entier affecte également l'Equateur dans les zones urbaines. Dans toutes les régions du pays (côtes, montagnes, Amazonie), des catastrophes naturelles telles que des glissements de terrain et des inondations arrivent fréquemment et ce aussi en zones peuplées entraînant la perte de vies humaines et des pertes économique de l'ordre de millions de dollars. Ainsi, une connaissance adéquate de la variation des paramètres météorologiques permettrait, par exemple, à travers les Systèmes d'Alerte Précoce (SAT) de prévoir des scénarios critiques permettant d'anticiper et donc d'assurer les bonnes pratiques agricoles mais aussi la sécurité des personnes. La collecte de données météorologiques ne peut pas être faite précisément dans tout le pays, c'est pourquoi nous nous cantonnons dans notre étude à l'analyse de la ville andine de Quito. La ville de Quito a une topographie unique ce qui rend difficile la prédiction des changements climatiques. Malgré certaines tentatives faites pour modéliser le climat de la ville de façon adéquate, les résultats sont restés infructueux et dans la plupart des cas, les équipements installés ont été extrêmement coûteux et leur fiabilité n'a pu être montrée en raison du manque de vérification de la qualité des données. En tenant compte de tous ces éléments, l'objectif de cette thèse est de prédire les paramètres environnementaux, en identifiant automatiquement les erreurs de mesure pour fournir un ajustement de la calibration des stations et ainsi améliorer la qualité des données. Pour résoudre ce problème, des micro-stations météorologiques automatiques (SMA) ont été installées sur les sites d'intérêt et les données ont été collectées sur un serveur FTP se trouvant dans le Cloud en utilisant le réseau cellulaire disponible.

De même, nous avons développé et analysé quatre modèles différents pour obtenir des prédictions fiables : un modèle ARIMA (auto-régression intégrée à moyenne mobile), un modèle LSTM (mémoire à court et long terme), un modèle LSTM empilé et un modèle convolutif LSTM avec une réduction des erreurs d'incertitude. Pour détecter les erreurs dans les micro-stations météorologiques automatiques, nous avons utilisé des séries temporelles de deux stations fortement corrélées avec la station analysée pour obtenir une prévision du paramètre mesuré (la température dans nos expériences) pendant 24 heures. Cela nous permet de déterminer si la station analysée enregistre des mesures erronées. Pour détecter les erreurs de mesure, une comparaison itérative est effectuée avec les informations de chaque micro-station en fonction de ses stations voisines. La différence entre les coefficients de corrélation initiaux et ceux acquis à l'instant  $t$  est calculée. Si cet écart dépasse un certain seuil, l'algorithme signale une

erreur et lance le calcul d'ajustement de cette erreur à partir de la prévision calculée pour cette station. Enfin, nous fournissons les paramètres de l'équation de calibration du capteur de la station en utilisant l'ajustement proposé. Pour mener à bien cette étude, nous avons utilisé différentes techniques, dont le calcul de coefficients de corrélation, l'utilisation d'un réseau de neurones multicouches, la conception d'une nouvelle version de l'optimiseur ADAM et une stratégie de réduction de l'incertitude basée sur une approche bayésienne.

# 1 Introduction

According to statistics generated in the National Institute of Statistics and Censuses of Ecuador (INEC), in 2018, the agricultural sector of Ecuador represented 8% of the Gross Domestic Product (GDP), and 95% of the food goods consumed internally in the country were generated in this sector (INEC, 2019). In addition, 29.3% of the total population Economically Active Population (EAP) was engaged in agricultural, hunting, forestry, and fishing activities in 2018. It is important to mention that agriculture is considered a fundamental source in the generation of foreign exchange (approximately 22 billion dollars). This is due to the export of traditional products such as bananas, coffee, and cocoa, as well as non-traditional products such as flowers, broccoli, among others. According to the National Institute of Meteorology and Hydrology of Ecuador (INAMHI), the climatic factors that mostly affect agriculture in Ecuador are low temperatures around 0°C and high temperatures around 36°C. The first scenario leads to the appearance of agricultural frosts, which, depending on the physiology of the plant (Maule, 2019), can severely affect the plant. Meanwhile, the second scenario, involves an increase in the phenomenon of evapotranspiration, which implies that crops suffer agricultural stress and wilting (INAMHI, 2016). Additionally, climate change in the country has produced phenomena in urban areas that imply a very high risk, especially in densely populated sectors. Landslides occur on the coast, in the mountains, and in the eastern region because Ecuador is a mountainous country. When the soil receives a large amount of water, it softens and breaks off to form mudflows that rush down the slope. Taking into account the above, an adequate knowledge of the variation of the meteorological parameters would allow, for example, through the Early Warning Systems (EWS) to forecast critical scenarios that may affect not only the correct maintenance of crops but also the safety of people. However, it is necessary to consider that the climate analysis cannot be proposed throughout the entire country since the work would be extremely extensive and costly. Based on the above, we propose the analysis for the Andean city of Quito, the capital of Ecuador, because it presents the ideal conditions to do the research. To get information about the environment, you need instruments that can get the right meteorological information. Instruments that meet the above requirements must be electronic so that they can automatically measure information and how it moves through the communication networks that are already in place. The lack of preventive maintenance of these Automatic Weather Stations (AWS) implies several problems, among which two stand out: the loss of accuracy and the complete failure of the equipment. Additionally, in current works, no emphasis is placed on the importance of the quality of the data registered by the stations (R. Sieber & Pudmenzky, 2022). Therefore, the lack of error detection in the measurement of the sensors would be an important inhibiting factor when mak-

ing a good weather forecast. However, the techniques used for error detection in data acquisition in AWS require further examination. In the present work, an approach to reducing the maintenance of AWS in rugged terrain through neural networks and using the city of Quito as an experimental site is presented. The error appears by comparing the actual data set of one station (called "under analysis") on a certain day against the predicted information of two neighbours (to be sure that the station under analysis shows errors) using previous information for that day. To ensure the similarity of the weather conditions in the neighbourhood of stations, the neighbour's selection follows two stages. First, the stations within the network whose real series show the highest correlation for air temperature compared to the Station Under Test (SUT) are isolated. Then the stations whose predicted series show the lowest error metrics compared to the actual series of the station under test conform to the other selections. The first step is to choose the stations with the best correlations, keeping in mind the total amount of data from the network of stations (75 days). Then, a weekly and daily analysis is done to check and confirm this relationship. The analysis was done with a set of 1440 records, since the information was collected every minute during the given time period. The dataset was divided into 24-hour groups to apply the walk-forward validation methodology and was subsequently divided into groups that approximated the 70% (training set), 15% (validation set), and 15% (test set). The second stage looks at a 24-hour air temperature forecast for each station in the network. The process repeats during the 11 days of continuous predictions to inspect the prediction quality through an error metric calculation between the real series (11 days) of the stations and their predictions. The error metrics are used to figure out which stations are near the station being looked at.

## 1.1 Motivation

As we get closer to the equator, it gets harder to predict the weather, especially in the Intertropical Convergence Zone (ITCZ), a low-pressure area at the equator where trade winds from the northeast and southeast meet. The ITCZ is a significant factor in the global climate system, and any changes in its weather patterns can significantly alter the physical model output of any forecasting system (M. P. Byrne & Wodzicki, 2018). Furthermore, if a city located in the Andes mountain range intersects the equator, the local weather forecast may be inaccurate due to the complex topography and weather patterns in the region.

An example of this phenomenon can be seen in the city of Quito, the capital of Ecuador, located in the Andean Region at an elevation of 2800 meters above sea level, which also happens to be where the equator runs through. The rough terrain of the Andes mountain range has shaped Quito's distinct landscape. This creates complex weather conditions and microclimates that make it even harder to predict the weather



(S. Serrano & et al., 2017). The presence of geographical features such as the Panecillo and the Guagua Pichincha Volcano also plays a crucial role in the city's weather patterns. These elevations have a big effect on Quito's convective processes, which cause temperature and weather changes that can be hard to predict.

To better understand and forecast the weather in cities like Quito, meteorologists must use advanced modeling techniques and sophisticated equipment to account for the complex atmospheric conditions and topography in the region. Meteorologists can deal with the unique weather patterns and physical features of equatorial cities by getting better at science and coming up with new ways to do things.

## 1.2 State of the Art

Ground stations, Automatic Weather Stations (AWS), Geostationary Operational Environmental Satellite (GOES) imagery, and forecast models based on artificial intelligence are just a few examples of modern components that have advanced the state-of-the-art of Weather Forecasting systems. As shown in Figure 1, the weather forecasting process covers several stages grouped together in the Forecast Chain (Rasp & Lerch, 2018).

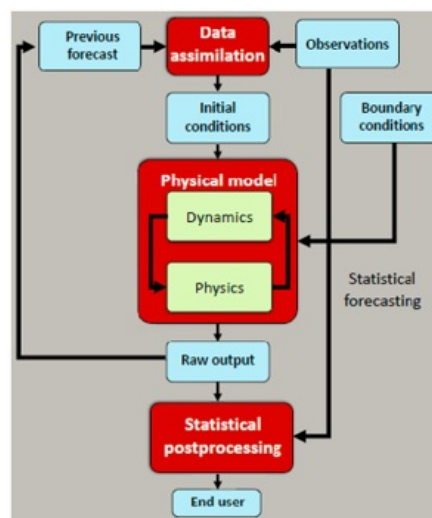


Figure 1: Forecast Chain.

The blue field in the picture shows the data acquisition and processing, while the red blocks indicate methods. The data acquisition and processing constitute the first stage of the chain. The "Observations" block comprises the information from Conventional Weather Stations (CWS), Automatic Weather Stations (AWS) and satellite imagery, while the "Data assimilation" comprises the combination of data from previous model forecasts to avoid errors related to a lack of information or incompatibilities from indirect measurements. Then the data obtained from the stage above is

sent to the physical model using initial conditions. The "physical model" focuses on the formulation and analysis of a mathematical approach applied to approximate the physical processes governing the weather evolution and the atmospheric state (dynamics and physics) in a certain place. The output called "raw output" encompasses the data predicted from the physical model that is re-entered into the model or is adjusted or corrected in the next stage. The data re-entry looks for adjustments to the model through a feedback strategy, while the statistical post-processing block corrects the forecasted data. The post-processing stage implements a short-term weather forecast system to carry out the statistical forecasting process. To achieve the post-processing, there are several alternatives, such as the use of Markov chains (Khiatani & Ghose, 2017) or other mathematical models. However, this is where neural networks show up due to their proven ability to obtain short-term forecasts. A Hidden Markov Mode (HMM) is a statistical Markov model that follows the Markov chain with hidden (secret) states criteria. A Markov chain (named after Andrei Andreyevich Markov, a Russian mathematician) is a prediction model that uses the probabilities of a sequence of random variables and states to predict a new unknown state. This ability allows a HMM to predict weather using the Markov chain property. That is, the method predicts the probability of occurrence of a future state from the current observation state of the system. It is for this reason that the evolution of the Markov process in the future depends only on the present state. To use this methodology, it is necessary to make use of historical data to train the model and compute the probability of the occurrence of an event. It is essential to elaborate on the initial states of the Markov chain, so abundant historical information should be available. For instance, (Khiatani & Ghose, 2017) uses weather data for a period of 21 years. The first step is to categorize the data based on standard values. Experimentally, the model works well for short-term weather forecasting (5 days) based on the weather pattern of the day chosen as the forecast point. Experimentally, it was determined that the use of the Markov model of low order is not reliable to make short-term forecasts and that as the Markov order increases, the accuracy of the forecasts made improves.

Nevertheless, the HMC method assumes that transition probabilities don't change over time, which isn't always the case in meteorology. Other approaches, such as the non-homogeneous Markov chain (NHMC) approach, can improve the HMC approach, but it involves the use of Bayesian nonparametric estimation to describe time-varying transition probabilities (E. Pope & Jackson, 2020). Consequently, it is necessary to use a modified Bayes rule that gradually forgets transitions in the distant past to update the transition probabilities. The above implies that Markov chains are not only simple to understand but can also be combined with other strategies. However, for the formulation of the Markov chains, it is necessary to assume that the data universe is of a homogeneous type. Considering the climate change suffered by the planet, or in the

case of Ecuador, due to its proximity to the intertropical convergence zone, there are no guarantees that the data are of a homogeneous type. This implies that in climatological studies, Markov chains model heterogeneous information as if it were homogeneous. In other words, the homogeneity of the meteorology of a place that the Markov Chains consider does not fully reflect its reality.

On the other hand, Artificial Intelligence (AI) methods have become increasingly important as ways of developing forecasting tasks. Three main approaches are available: Machine Learning (ML), Neural Network (NN) and Deep Learning (DL). The advantages of each one often result in debates as to which method best suits the need for research. This section highlights how to select a methodology to better suit research focused on weather forecasting. Machine learning is a set of algorithms that first analyze data to find relationships and patterns. Secondly, learn from what they have found. Thirdly, apply what they have learned from the behaviors that they have defined. Neural networks, also known as Artificial Neural Network (ANN) are a subset of machine learning. Thus, neural networks are a subset of algorithms used in machine learning for modeling the data using structures called neurons. Therefore, the main difference is that a neural network uses a collection of neurons that simulates the complexity of a human's brain to analyze the data. On the other hand, deep learning is a subfield of machine learning and neural networks that works on multiple layers of neural networks to extract more precisely the relationships and patterns of the data. Machine learning methods will be present in many components of modern climate models and numerical weather prediction thanks to trends in new computer clusters (Bochenek & Ustrnul, 2022). This happens due to the rapid growth of high-speed clusters thanks to Graphics Processing Unit (GPU) accelerators in recent years. As seen in recent years, the field of machine learning in weather and climate science has grown rapidly as architectures that are more sophisticated become available in modern computing systems (M. Schultz & et al., 2021). So working with neural networks and time series or imagery processing has become the most widely used methodology to be able to implement weather forecasts because of its reliability and development facilities. Now, considering the data acquisition stage, nowadays the use of ground-based automatic stations to acquire environmental information is common in developing countries, although their maintenance and operation are still costly. One alternative to solving this problem consists of the use of satellite imagery. Especially for member countries of the World Meteorological Organization (WMO), this information is freely accessible. It represents several advantages, especially the acquisition of infrared and visible information with good spatial resolution that is freely accessible thanks to National Oceanic and Atmospheric Administration (NOAA) repositories. Although there are several advantages in the use of this type of information, satellite imagery implies the acquisition of an image of Earth with a 1 km resolution taken from an altitude of 36000 km above

the surface (O’Carroll & Leslie, 2009). The above leads to two problems: firstly, clouds could absorb, emit, and reflect radiation, so the modification of the planetary boundary layer limits produced by the cold pool dynamics of the clouds generates convective systems that are not registered by the satellite imagery. Secondly, the size of countries like Ecuador compared to the size of the grid means that the quality of the image does not provide the respective advantages. Both problems lead to inaccuracies in weather forecast systems strongly based on satellite imagery (H. Bluestein & et al, 2022). Therefore, it is necessary to work with ground information to complement the satellite imagery information.

### 1.3 Context

The National Institute of Meteorology and Hydrology of Ecuador (INAMHI) has three stations (S. Serrano & et al., 2017) in the city, while the Quito Astronomical Observatory (OAQ) and the Civil Aviation Directorate (DAC) each supervise the other two stations (Villacis & Marrero, 2017). The WMO regional models use the information from the INAMHI and DAC stations to produce regional weather forecasts. Importantly, the information from the above institutions was not available for this research because the corresponding permissions were not currently available. However, the information from these stations has served to verify the acquired environmental values. As mentioned before, the rugged geography of Quito creates several microclimates along the city, and bearing in mind that for the physical model, the establishment of initial conditions is fundamental, the lack of surface information could lead to an erroneous forecast for the city. To solve any inaccuracies in the model prediction, the statistical post-processing uses the original data from the CWS, AWS, and satellite imagery to adjust the output of the physical model. Statistical post-processing adjusts the inaccuracies in the physical model by making use of a weather forecast obtained through statistical methods. An alternative to implementing a statistical prediction process is the use of time series and neural networks. These networks extract the features obtained from the Times Series and compute them to implement a statistical model to obtain a reliable prediction of the observed values. Such a non-dynamical approach can be helpful for short-term forecasts, up to a few hours (Rasp & Lerch, 2018). This methodology has proven to be effective, especially for obtaining reliable short-term weather forecasts.

Considering that Ecuador has only recently started using AWS networks and that information from CWS and satellite images is often used, combining the three sources of information would be a better way to predict the future. Considering that the physical model uses information from the CWS and satellite imagery to generate weather forecasts, an interesting strategy to carry out the adjustment of the output of

the physical model could be the use of neural networks working with the information extracted from the AWS. The two main problems in obtaining a proper weather forecast for the city of Quito depend on the rugged geography of the city and its proximity to zero latitude. Quito encompasses a wide range of natural formations, among which the Guagua Pichincha volcano stands out with an altitude of 4776 m.a.m.s.l. That is, there is a difference of about two thousand meters between the volcano and the city. The microclimates generated by the rugged geography of the city and the convective systems on the surface produced by the altitude difference between the city and its surroundings deeply affect the atmosphere's dynamics around the city. An alternative to this predicament is the deployment of low-cost AWS, implemented with free software, in strategic points of the city.

Any method for forecasting requires high volumes of information to be able to determine the particular characteristics of the information and, thus, forecast future outcomes. To acquire enough surface information, it is necessary to transmit temperature, humidity, and pressure data in real time. Low-cost single-board computers and low-cost sensors are the building blocks of the AWS. The environmental data acquired comes from sensors connected to the single-board computer and is fed into a neural network to implement the weather forecast. As mentioned earlier, for the purpose of carrying out the measurement of weather parameters in the city of Quito, the government institutions INAMHI, OAQ, and DAC have installed five AWS along Quito each to transmit information through the cellular network (Villacis & Marrero, 2017) that is available online. Nevertheless, such information was not available for the realization of this research because the corresponding permissions were not currently available. However, the information from these stations has served to verify the acquired environmental values.

## 1.4 Contribution

To install the new low-cost AWS network, the rugged geography of the city is analyzed. Quito is a city located at 2850 m.a.m.s.l. divided in its central part by the hill of El Panecillo (3035 m.a.m.s.l.) to the east by the hills of Puengasí (2897 m.a.m.s.l), Guanguiltagua (2890 m.a.m.s.l) and Itchimbí (2910 m.a.m.s.l) and to the west by the Pichincha volcano (4776 m.a.m.s.l), see Figure 2.

As shown in Figure 2, Quito's geography complicates the generation of accurate weather forecasts due to the formation of microclimates throughout the city. Based on the above, five locations were strategically selected for the AWS installation. The installation followed the WMO recommendations for weather station installation in urban spaces (WMO, 2008) and the data acquisition started in February of 2020. Once the installation of the AWS network of stations has been completed, the condi-



Figure 2: Variations of the weather in Quito depending on the geography.

tioning and processing necessary to obtain a weather forecast begin. At this point, it's important to figure out how uncertain the predicted data are so that the output of the neural network can be changed to make sure the statistical modeling works. Bayesian modeling is a very interesting way to figure out and even change a neural network's uncertainty because it uses changes in the weight distribution of the network to make a model behave in a Bayesian way. Several works written recently show how Bayesian analysis is a good way to figure out how uncertain a model is. In (L. Cardelli & et al., 2019) a method to obtain a reliable image recognition system is described for estimating the probabilistic robustness of a Bayesian Neural Network (BNN) with adequate confidence limits and a priori error. The use of Bayesian modeling has aroused interest even at "unit level" analysis (M. Vladimirova & et al., 2019) namely to characterize the marginal prior distribution of the units in deep learning. Finally, to extrapolate large datasets over various structures with reliable uncertainties, we can use new kinds of Bayesian neural networks (S. Sun & et al., 2019) for instance, Functional Variational Bayesian Neural Networks (FBNNs).

Despite the apparent advantages offered by Bayesian modeling, some authors state that approximate Bayesian inference methods struggle to capture true posterior probabilities (J. Yao & et al., 2019). Therefore, it is important to properly and carefully evaluate the methods that, when working together with uncertainty reduction through Bayesian methods, allow obtaining a better response from the neural network model used. We did an analysis using estimator-based posterior predictive distribution fitting (see Chapter 3) to find a way to reduce the forecast error variance. Bearing the above in mind, we developed a methodology based on a Bayesian approach using the weight decay rate ( $\lambda$ ). This strategy allowed us to reduce the uncertainty obtained from the neural networks used. Consequently, we obtained a short-term weather forecast with good precision for the Andean city of Quito. Taking into account that we think the weight decay rate works with the process of reducing uncertainty, we made a version of the well-known ADAM optimizer that we called ADAM Logger (ADAML). By adding the weight decay rate, we can separate the weight decay from the gradient-based update. This makes it easier to find the minimum of the learning function of the

model. Also, a walk-forward validation is used to test the optimization on a subset of the training set whose size changes with each iteration of the validation process. This makes sure that the optimization will be applied to all the data sets.

So that an accurate analysis of the proposed method can be made, different error metrics like root mean squared error (RMSE), mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are shown for each test scenario. Lastly, we come up with a way to figure out how wrong the weather stations' measurements are by using data from "neighboring" stations. There are two ways to choose a station's neighbors. The first is to look at the 75-day data and choose the stations that are most similar to the one being studied. The second approach consists of obtaining the neighbors by considering the lowest error metric values between the forecast series of 11 extra days for the stations and their actual values. The study reveals that the north-central part of the Andean city of Quito experiences the most variable climatic conditions of all the areas selected for monitoring. Despite this, a relationship is established between the stations of the network that allows us to determine from information from a neighborhood of stations whether a station presents measurement errors and thus ensure the quality of the data. To find out if the above-mentioned method is a good way to implement a method for auto adjustment to prevent maintenance in AWS for the Andean city of Quito, we came up with the following goals:

- To build and install (in the Andean city of Quito) a network of micro-AWS with open source hardware for data acquisition.
- To develop a methodology to reduce the uncertainty in a neural network.
- To develop a method for detecting errors in the weather data of an AWS using correlation coefficients and a neural network with multiple layers.
- To develop an algorithm (a perceptron) to adjust internal sensor parameters in a micro AWS to obtain the correct value of the weather parameter measured.
- To determine a proper distribution of the nodes bearing in mind a weather analysis (Isomaps).

## 1.5 Content

We divided this manuscript as follows: First, we talk about the big picture of the different meteorological ideas and the limits of getting a good weather forecast because of the weather conditions in the Intertropical Convergence Zone. We also explain how weather forecasts are made, including the different steps in the chain of forecasts and the different ways to get a weather forecast. In the first chapter, we review the generalities of the neural network model and the tools utilized to verify the

feasibility of the methodology through the analysis of error metrics and correlation. We discuss the differences between artificial intelligence (AI) methods and the types of neural networks. While the rest of the chapter shows up the validation strategies adopted and the types of optimizers to provide detailed information on the state of the art of neural networks. The second chapter is dedicated to analyzing the characteristics of Quito's weather. This section describes the geographical conditions of the city and the strategy adopted through exploratory research to obtain short-term weather forecasts in the mentioned sector. This part corresponds to the published article Deep Learning to implement a Statistical Weather Forecast for the Andean City of Quito, DOI: 10.1109/ANDESCON50619.2020.9272106, published in the IEEE ANDESCON 2020 International Conference held in Ecuador, and incorporates some additional details. In the third chapter, a review is given of the Bayesian strategy used to figure out a neural network's uncertainty and the estimators used to change it. Next, we talk about the theory and development of the optimizer that was used in the research's neural networks. There is both a mathematical proof of the optimizer's convergence and the experimentation that went along with it. This part corresponds to the published articles: 1) Uncertainty Reduction in the Neural Network's Weather Forecast for the Andean City of Quito Through the Adjustment of the Posterior Predictive Distribution Based on Estimators, DOI: 10.1007/978-3-030-62833-8\_39, published in the TICEC 2020 International Conference held in Ecuador. 2) A novel encoder-decoder structure for time series analysis based on Bayesian uncertainty reduction, DOI: 10.1109/LA-CCI48322.2021.9769850, published in the IEEE LACCI 2021 International Conference held in Chile. 3) A novel ADAM approach related to decoupled weight decay (ADAML), DOI: 10.1109/LA-CCI48322.2021.9769816, published in the IEEE LACCI 2021 International Conference held in Chile and incorporates some additional details. The fourth chapter talks about the strategy for finding mistakes and the approach for fixing them. This section shows the correlation analysis of the AWS network stations and the proposed forecast to figure out the error in their data acquisition and how to fix it. This part corresponds to the journal article: A novel approach for detecting error measurements in a network of automatic weather stations, DOI: 10.1080/17445760.2021.2022672, published in the International Journal of Parallel, Emergent, and Distributed Systems, Taylor & Francis, and incorporates some additional details. In the general conclusions, the best parts of the work and results of this study are emphasized. At the end of this manuscript, a set of possible answers to the unanswered questions is suggested.



## 2 Generalities of Neural Networks & Tools

This chapter provides an overview of the history of machine learning and neural networks. The types of networks, structures, and configurations as well as their applications based on the forecasting of meteorological parameters are presented. In each case, the state of the art is described, including the most recent works and experiments. The chapter concludes by revealing the type of validation adopted for the investigation and the optimizers utilized to conduct the experiments. Today, neural networks permit the resolution of a variety of issues, including facial recognition, social media, aerospace, defense, healthcare, signature verification and handwriting analysis, stock market prediction, and environmental prediction, among others. In a few words, the application of neural networks has revolutionized traditional mathematical modeling and programming approaches. For neural networks to perform the aforementioned tasks, their ability to perform tasks with infinite combinations based on the analysis of data sets and time series is crucial. In terms of information forecasting, it has been determined through experimentation that neural networks permit accurate short-term forecasts. This is due to the ability to learn from trends, particularly in the description of nonlinear, complex relationships present in time series. Configurations of neural networks permit the development of statistical models that fully or partially capture the characteristics of the time series. However, experimentation is required to determine the optimal network configuration for producing the best approximation of the actual data. Moreover, given that the forecast task entails the ability to predict the future data of a time series, it is essential for any project to rely on historical data of particular values in order to execute such a process.

### 2.1 History

Although it is true that in 1936 Alan Turing developed an applied version of a computer that allowed the messages of the German Enigma decoding machine to be decoded during World War II, the first theorists who modeled a simple neural network using electrical circuits were Warren McCulloch and Walter Pitts in 1943 (Matich, 2001). Despite these advances, it was not until 1949 that Donald Hebb first described learning processes (a basic element of human intelligence) from a psychological point of view, developing a rule for how learning occurred. This concept states that learning occurs when certain changes in a neuron are activated. On the other hand, Karl Lashley in 1950 found that information was not stored centrally in the brain but on top of it, which implies a revolutionary concept in the study of learning and memory. In 1957, Frank Rosenblatt began the development of the perceptron. This is the oldest neural network, whose function today focuses on pattern identification. This

model was capable of generalizing, that is, after having learned a series of patterns, it could recognize other similar ones even if they had not been presented to it during the training stage. However, it had a number of limitations, for example, its inability to solve the exclusive-OR function problem, and, in general, it was unable to classify non-linearly separable classes. In 1960, Bernard Widrow and Marcian Hoff developed the Adaptive Linear Elements model (Adaline). This was the first neural network applied to a real problem: adaptive filters to remove echoes on telephone lines. In 1974, Paul Werbos developed the basic idea of the backpropagation learning algorithm. In 1977, Stephen Grossberg proposed the Adapted Resonance Theory. Which is a network architecture focused on simulating brain abilities such as long- and short term memory. In 1982, John Hopfield modeled an approach to creating machines using bidirectional lines. While in the same year, Reilly and Cooper used the hybrid network with multiple layers to create a neural model for category learning (Matich, 2001). In 1986, neural networks started to work with multiple layers using the Widrow-Hoff rule, and at the same time, David Rumelhart, a former fellow in the psychology department at Stanford, modeled the backpropagation network. The disadvantage of the backpropagation network at the time was that it learned slowly, requiring thousands of iterations to become efficient.

## 2.2 Neural Network Operation

Neural networks receive a series of input values that are used to connect neurons, which are nodes. The layers formed by the network's neurons constitute the neural network. Each neuron in the network has a weight, a numeric value that modifies the significance of the incoming input. As shown in Figure 3, the newly-obtained values leave the neurons and continue their journey through the network.

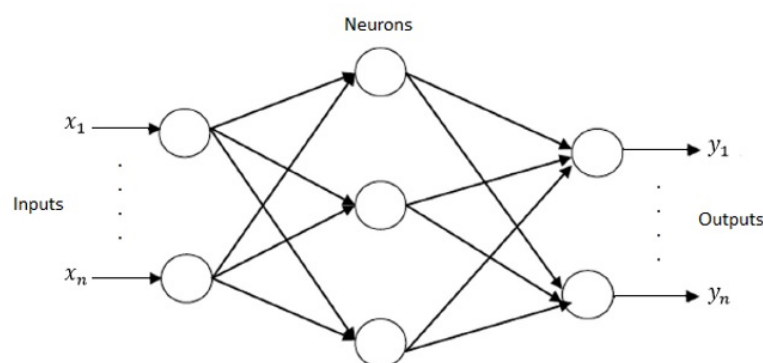


Figure 3: Neural Network scheme.

When you get to the end of the network, you get an output, which is the prediction that the network made. The more layers the network has and the more complex it is, the more complex the functions it can perform. The network must be trained to per-

form the desired functions for it to operate properly. The way to train a neural network is to change the weights of its neurons so that it can get the results you want. To train the network, a set of training data is fed into it, and the weights of the neurons change based on the error and how much each neuron contributed to the result. This is the origin of the backpropagation or backward propagation approach. With this method, the network learns, achieving a model capable of obtaining very accurate results even with data that is very different from that used during its training. It is precisely thanks to the creation of the backpropagation algorithm that the use of GPU<sup>1</sup> and the greater number of data available for training that neural networks have now resurfaced and gained prominence in various fields. Thanks to these improvements, the appearance of deep learning has been made possible. The backpropagation algorithm is the most common way for neural networks to learn, but because of how it works, it has some major flaws, such as a slow learning process. To make sure that a neural network's predictions are as accurate as possible, the weights of each neuron must be changed. The backpropagation algorithm shows how much each neuron contributes to the overall error. The backpropagation method uses a two-step cycle of propagation and adaptation that is based on the gradient. First, when we put a pattern into the network's input, it moves from the first layer through the rest of the network's layers until it comes out of the network's output. The neural network then compares the output signal to the output it wants and figures out an error signal for each output. The algorithm sends the error outputs from the output layer back to all the neurons in the hidden layers. Each neuron's contribution determines how much of the total error signal is sent back. The algorithm repeats the process layer by layer until all of the neurons in the network have received the error signal. In this way, the neurons learn to recognize different parts of the patterns that are fed into the network. So, the neurons in the hidden layer of the network will respond with an active output if the new input has a pattern that looks like a trait that the individual neurons have learned to recognize during their training.

The cost function tries to figure out how far off the estimated value is from the real value. It is a type of function, also called in some scenarios objective Objective Function. The cost function measures the error between the output value and the real value in order to optimize the parameters of the neural network (Aggarwal, 2018). It seeks to reverse the transmission of the error by constantly adjusting the weight and the threshold in the network so that the gap between the predicted value and the real value is reduced. Among the best-known cost functions are the root mean square and the categorical cross-entropy. The root mean square error provides a measure of precision and is calculated as the root mean square (RMSE) of the residuals. Residuals

---

<sup>1</sup>A graphics processing unit Central Processing Unit (CPU) is a processor with a large amount of memory that allows for floating-point operations and quick graphics rendering. The main difference between GPU and CPU is that GPUs use more transistors to implement Arithmetic Logic Unit (ALU) tasks and less for caching and flow control compared to CPUs.

constitute the difference between the predicted (correct) value and the actual value obtained. Among the most notable characteristics of this approach, one can note that it penalizes very large values, is not easily interpretable, and works very well to optimize regressions in general (see Eq. (1) (Jurafsky & Martin, 2021)).

$$\text{RMSE} = \sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n} \quad (1)$$

Where:

$\hat{y}_i$  : Predicted values.

$y_i$  : Dependent variable.

$n$  : Number of observations.

On the other hand, categorical cross entropy  $L(\theta)$  is a measure of precision for categorical variables. One of the most notable characteristics of this approach is that its differentiation and convergence are more difficult to obtain, present a univariate scale, are symmetric and are easier to interpret, see Eq. (2) (Jurafsky & Martin, 2021).

$$L(\theta) = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m y_{ij} \log(p_{ij}) \quad (2)$$

Where:

$y_{ij}$  : Observed random variable (class).

$p_{ij}$  : Variable (class) to predict.

$c$  : Number of classes in a multilevel classification.

$n$  : Number of observations.

In order to assign weights in the backpropagation process, it is necessary to calculate the gradient of the loss function. To carry out this process, the so-called "activation functions" are used. An activation function is a function that seeks to help a neural network learn complex patterns in data by taking the output signal from the previous cell and converting it into the input to the next cell. Additionally, activation functions determine how to transform the weighted sum of the inputs at the neural network to obtain an output at the end layer of the network. Activation functions make backpropagation possible, as gradients are present along with the error to update weights and biases throughout the network. The most popular activation functions are mentioned below.

*Sigmoid function* The sigmoid function transforms the entered values into a scale between 0 and 1 (Aggarwal, 2018), see Figure 4(a), where high values asymptotically tend to 1 and very low values asymptotically tend to 0, see Eq. (3) (Jurafsky & Martin, 2021).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

The sigmoid function tends to saturate and reduce the gradient; it presents a slow convergence; it is not centered at zero; and it is bounded between 0 and 1. However, it presents a good performance in the last layer.

*Hyperbolic Tangent (Tanh) function* The hyperbolic tangent function transforms the entered values into a scale between -1 and 1 (Aggarwal, 2018), see Figure 4(b), where high values asymptotically tend to 1 and very low values asymptotically tend to -1, see Eq. (4) (Jurafsky & Martin, 2021).

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (4)$$

The hyperbolic tangent function behaves very similarly to the sigmoid function since it tends to saturate and reduce the gradient of the function to zero. On the other hand, it shows a slow convergence, nevertheless, it can be said that it is a function centered at 0 and bounded between -1 and 1. It is a function used to decide between two options and it performs well in recurrent neural networks (RNN).

*Rectified Linear Unit (ReLU) function* The ReLU function transforms the entered values by nulling out negative values and leaving positive ones (Aggarwal, 2018), see Figure 4(c), where high values asymptotically tend to 1 and very low values asymptotically tend to -1, see Eq. (5) (Jurafsky & Martin, 2021).

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (5)$$

The ReLU function only works against positive values (Sparse Activation) and it cannot be bound. By its nature, it is a function that causes several neurons to stop working when detecting negative values. However, it behaves well when working with images. Finally, it presents a good performance in convolutional neural networks (CNN).

*Leaky Rectified Linear Unit function (ReLU)* The Rectified Linear Unit (ReLU) function transforms the entered values by multiplying the negative ones by a rectifying coefficient and leaving the positive ones when the data at the neuron are entered; see Figure 4(d) and Eq. (6) (Aggarwal, 2018).

$$f(x) = \max(ax, x) = \begin{cases} a \cdot x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (6)$$

Where:

a: Small slope with negative values to avoid the dying neuron problem. The Leaky ReLU function has similar performance to the ReLU function. It is a function that penalizes negative values by means of a rectifier coefficient but is not bound, works well with images and presents a good performance in CNN.

*Softmax function* The Softmax function transforms the outputs into a representation in the form of probabilities, such that the sum of all the probabilities of the outputs equals 1, see Figure 4(e) and Eq. (7) (Aggarwal, 2018).

$$\sigma(z) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \text{for } i = 1, \dots, K \quad (7)$$

Where:

$z_i$  : Elements of the input vector  $z = (z_1, \dots, z_K) \in \mathbb{R}^K$ .

The Softmax function applies the exponential function to each element of the vector  $z$ , normalizing them by the sum of their own exponential values. With this, the function ensures that the sum of the output vector is equal to one. We can consider it a very differentiable function, so we can utilize it to normalize type values in multiclass scenarios. It presents a good performance in the last layers of a network.

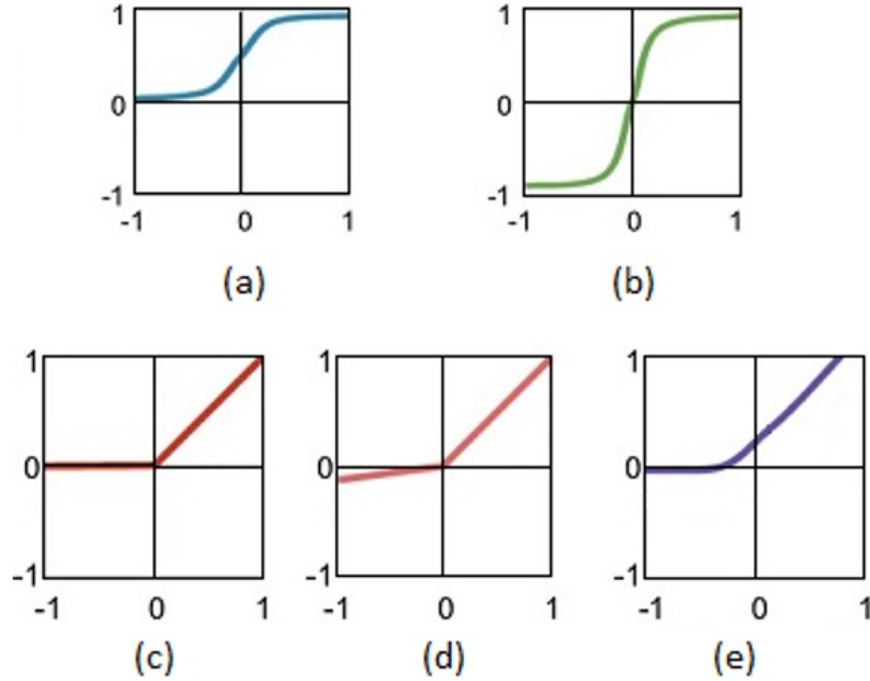


Figure 4: Activation functions. (a) Sigmoid function, (b) Tanh function, (c) ReLU function, (d) Leaky ReLU function, (e) Softmax function.

The most important feature of an activation function is its ability to add non-linearity to a neural network. However, an activation function should also accomplish some desirable features such as being zero-centered, computationally inexpensive, differentiable, and overcoming the Vanishing Gradient problem. The zero-centered property consists of forcing the output of the activation function to be symmetric about zero. That is, the gradients must not change in a particular direction. Meanwhile, the computationally inexpensive approach should be accomplished because the activation functions are applied after every layer, namely several times (depending on the

structure). On the other hand, the differentiable property refers to the fact that, for the optimization approach, it is necessary to obtain the gradient of the function. Finally, the Vanishing Gradient problem refers to the multiple calculation stages for the weights during the backward pass. The following scenario describes the Vanishing Gradient problem (Aggarwal, 2018).

Considering a network of two layers represented as  $f_1(x)$  and  $f_2(x)$  the general network will be  $g(x) = f_2(f_1(x))$ . Then the algorithm calculates the gradient to retrieve the weights for the entries as  $g'(x) = f_2'(x)f_1'(x)$ . We can replace  $f_1(x)$  with  $f(W_1x_1 + b_1)$ , where  $W$  is the weight matrix,  $x$  the input vector and  $b$  the bias vector. Applying the chain rule, we obtain  $f_1'(x) = f'(W_1x_1 + b_1)x_1$ . The above shows a strong relationship between the backpropagation process and the activation function. If the value of the function is between 0 and 1, this reduces the value of the gradient for the initial layers. Moreover, bearing in mind that the network could comprise several layers of neural networks, their gradients tend to disappear. Consequently, the gradients tend to vanish and the layers affected by this calculation are not able to learn properly. This is the so-called vanishing gradient problem.

## 2.3 Types of Neural Networks

In the following, we illustrate the types of neural networks that are now the most widely recognized. In this section of the document, the typical configurations of the networks and their related mathematical descriptions are presented.

### 2.3.1 Recurrent Neural Networks

We have seen networks up until this point whose activation function only functions in one direction, forward, from the input layer to the output layer. According to what has been stated above, they do not remember the prior settings. Recurrent Neural Networks (RNN) is designed with "backward-pointing" connections in some places. Figure 5 illustrates a form of feedback that can occur between neurons that are located inside the same layer.

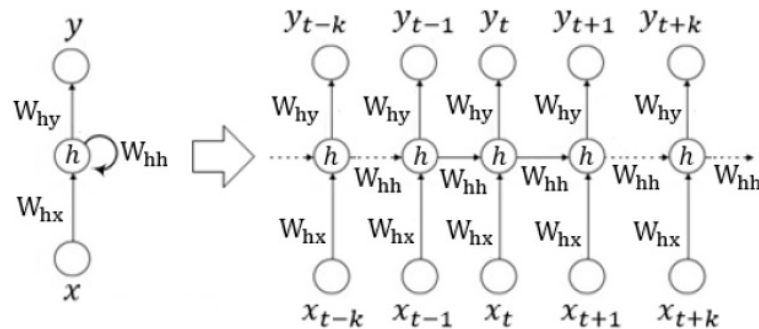


Figure 5: Neural Network scheme.

In their most basic form, the outputs of a RNN  $y_{t-k}, \dots, y_{t-1}, y_t, y_{t+1}, \dots, y_{t+k}$  are determined by the inputs  $x_{t-k}, \dots, x_{t-1}, x_t, x_{t+1}, \dots, x_{t+k}$  (Géron, 2017). The present states of  $x$  and  $y$  are identified with  $t$  and the previous or next states of  $x$  and  $y$  with  $t-k$  or  $t+k$  respectively. In this recurring structure, at each instant "time step", the neurons not only receive the input from the previous layer (with the  $h$  hidden layer vector), but also their own output at the same time to generate their output. Additionally, it has to be said that  $W$  is the weight matrix.

Given the fact that the output of a recurrent neuron at every given time instant is a function of the input it received at earlier time instants, the discussion above suggests that recurrent neurons have a certain amount of "Internal Memory" capacity. The component of the neural network that maintains a state throughout the course of time is referred to as a memory cell (or just a cell for short). Because of its potential for memorizing, this form of network is ideally suited for applying machine learning techniques to problems in which time series are involved (A. Alzahrani & et al., 2017). In spite of the benefits that were discussed earlier, the performance of the recurrent network quickly declines as the length of the input stream grows longer. This is due to the fact that increasing the number of activation functions in the structure of the neural network brings the gradients of the loss function closer to zero, which is known as a vanishing gradient. This makes it more difficult to train the network, as stated by Shuyang. Exploding gradients and vanishing gradients are two common yet significant problems that arise with RNNs, as well as with other types of networks that have a big number of parameters. In general, these problems arise with any sort of network that has a lot of parameters. Exploding gradients are caused when an algorithm gives a disproportionately high value to the weights for little to no discernible reason; as a result, this creates a challenge for the training process. It is essential to keep in mind that the faster a model can learn, the higher the gradient, the steeper the slope, and the more information it can take in. Nevertheless, if there is no slope at all, the model will stop learning.

### 2.3.2 Long short-term Memory (LSTM)

Adding memory blocks to the recurring structure addresses the memory issue, enabling the learning of long-term dependencies. Long Short Term Memory (LSTM) are an extension of recurrent neural networks that use an extension to the memory to learn from important experiences that have happened a long time ago. An LSTM neuron can read, write, and delete information from its memory. LSTM is a type of RNN that combines memory stages (cells) into special structures known as gates to store information (V. Athira & et al., 2018), see Figure 6. Three types of gates are present in the LSTM structure: "forget" ( $f_t$ ), "input" ( $i_t$ ) and "output" ( $o_t$ ). The



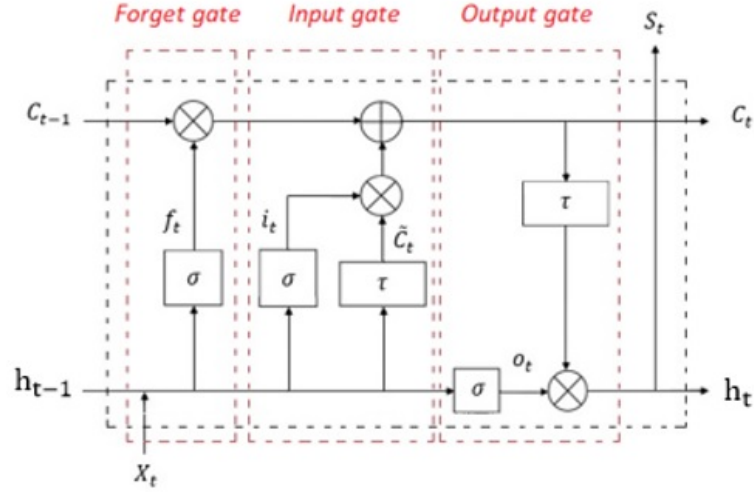


Figure 6: Structure of LSTM.

operation of these gates is mathematically described below (I. Kamal & et al., 2020).

$$\text{Forget gate: } f_t = \sigma \left( W_{xf}^T \cdot X_t + W_{hf}^T \cdot h_{t-1} + b_f \right) \quad (8)$$

$$\text{Input gate: } i_t = \sigma \left( W_{xi}^T \cdot X_t + W_{hi}^T \cdot h_{t-1} + b_i \right) \quad (9)$$

$$\text{Output gate: } O_t = \sigma \left( W_{xo} \cdot X_t + W_{ho} \cdot h_{t-1} + b_o \right) \quad (10)$$

$$\text{New Candidate: } \tilde{C}_t = \tanh \left( W_{xc} \cdot X_t + W_{hc} \cdot h_{t-1} + b_c \right) \quad (11)$$

$$\text{Cell state: } C_t = C_{t-1} \otimes f_t + i_t \otimes \tilde{C}_t \quad (12)$$

$$\text{Hidden state: } h_t = O_t \otimes \tanh(C_t) \quad (13)$$

Where:

$X_t$ : Input vector.

$\sigma$ : Sigmoid function.

$\tau$ : Tanh function.

$b_f, b_i, b_c, b_o$ : Bias vectors for each gate and block input.

$W_{xi}, W_{xf}, W_{xc}, W_{xo}$ : Weight matrices of each gate and block input for their connection to the input vector  $X_t$ .

$W_{hi}, W_{hf}, W_{hc}, W_{ho}$ : Weight matrices of each gate and block input for their connection to the previous short-term state  $h_{t-1}$ .

$\tilde{C}_t$ : Hidden state called "candidate" which is calculated based on the information of the current input and the previous hidden state.

$C_t$ : Internal memory of the unit, which is the combination of the previously stored information, the forgetting gate, the input gate and the recently calculated hidden state.

With the inclusion of this type of architecture, LSTM networks can handle extensive

temporal data dependencies, thus overcoming the gradient fading problem in RNNs. The dimensions of the arrays (matrices) are determined considering the batch size ( $B$ ), the number of features ( $F$ ) entered into the network and the number of units in an LSTM cell ( $U$ ) as follows: For  $X_t$  the dimensions will be  $B \times F$ , on the other hand for  $h_{t-1}$ ,  $h_t$  and  $C_{t-1}$  the dimensions are  $B \times U$ , for the weights matrices  $W_{xi}$ ,  $W_{xf}$ ,  $W_{xc}$  and  $W_{xo}$  the dimensions are  $F \times U$ , while for the weight matrices  $W_{hi}$ ,  $W_{hf}$ ,  $W_{hc}$  and  $W_{ho}$  the dimensions are  $U \times U$ . In the case of the weight matrices  $W_i$ ,  $W_c$ ,  $W_f$  and  $W_o$  the dimensions are  $F + U \times U$ , while for the bias vectors  $b_i$ ,  $b_c$ ,  $b_f$  and  $b_o$  the dimensions are  $U$ . Finally, the dimensions for the parameters  $i_t$ ,  $f_t$ ,  $C_t$ ,  $h_t$  and  $O_t$ , are  $B \times U$ .

### 2.3.3 Gated Recurring Units (GRU)

Gated Recurring Units (GRU) is another type of recurring network that uses memory cells to manage temporal dependencies, but in a simplified way. It synthesizes the forget door and the input gate into a single update door and mixes the cell state and the hidden state (I. Kamal & et al., 2020). Consequently, there are only two gates in the GRU model: the update gate ( $z_t$ ) and the reset gate ( $r_t$ ), see Figure 7.

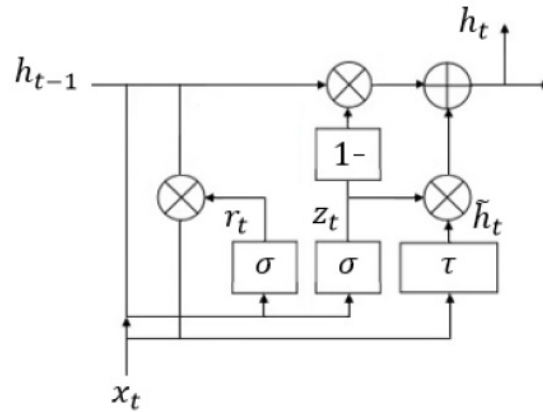


Figure 7: Structure of GRU.

Then, the following equations describes the mathematics related to the management of this type of structures.

$$\text{Update gate vector: } z_t = \sigma_g (W_z x_t + U_z h_{t-1} + b_z) \quad (14)$$

$$\text{Reset gate vector: } r_t = \sigma_g (W_r x_t + U_r h_{t-1} + b_r) \quad (15)$$

$$\text{Candidate action vector: } \tilde{h}_t = \tanh (W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \quad (16)$$

$$\text{Output vector: } h_t = z_t \odot \tilde{h}_t + (1 - z_t) \odot h_{t-1} \quad (17)$$

Where:

$x_t$  : Input vector.

- $W_z$  : Update gate Weight matrix.
- $W_r$  : Reset gate Weight matrix.
- $W_h$  : Candidate action Weight matrix.
- $U$  : Number of units at the GRU cell.
- $b$  : Bias vector.
- $\sigma_g$  : Sigmoid function.

In this case, the update gate controls the degree to which state information from the previous time is brought into the current state and the restart gate controls how much information from the previous state is written to the current candidate set  $\tilde{h}_t$ . With this structure, a larger value of the update gate indicates that more state information was entered at the previous time. Bearing in mind the above, the smaller the reset gate, the less information is written to the previous state. In comparison to LSTM, this structure is able to carry out a "more agile" learning process as a result of its faster information processing and, subsequently, its ability to learn new things.

To define the dimension of  $h_0$ , it is necessary to consider, first, that it is a hyperparameter and that its value will depend on experimentation and bibliographic review.  $W_z x_t$  is a vector of the same size as  $h_t$ , since it obeys the embedding dimension. The sigmoid operation occurs as an element-wise product and  $1 - z_t$  appears as a vector subtraction, where the unity is a vector of ones of the same size as  $h_t$ . Finally, bearing in mind that  $W_z x_t$  is a matrix multiplication, the number of columns of  $W_z$  should be equal to the dimension of  $x_t$  and the number of rows of  $W_z$  should be equal to the dimension of  $z_t$ .

### 2.3.4 Stacked LSTM Neural Networks

The original LSTM model consists of a single hidden LSTM layer and a conventional forward output layer. Stacked LSTM is an extension of this model that has multiple hidden LSTM layers where each layer contains multiple memory cells. Stacking hidden layers of LSTM makes the model deeper, which makes the description of deep learning more accurate. Since LSTMs operate on sequence data, the addition of layers adds levels of abstraction from the input observations over time. In fact, it fragments the observations over time or represents the problem on different time scales. Because of the creation of a more complex feature representation of the input, this approach constitutes a very useful alternative to solve a wide range of prediction problems. LSTM stacking has proven to be very useful in obtaining good weather forecasts based on historical records (D. Kreuzer & et al., 2020). For instance, in (Zaytar & Amrani, 2016) it is mentioned that the meteorological data collected from nine stations installed at the airports of nine cities was used to perform a 12-Hours forecast using a multi-layer model with RNN. In this case, the error per hour was in the range of  $0.01^\circ\text{C}$  to  $3^\circ\text{C}$ .

### 2.3.5 Auto-regressive integrated moving average network (ARIMA)

Auto-Regressive Integrated Moving Average (ARIMA) model is one of the most widely used approaches for time series forecasting. The model is defined by modeling the expected value of a random variable at time  $t$  using its own lags and error terms in past periods. This happens because ARIMA allows for both Auto-Regressive (AR), differencing, and Moving Average (MA) components. The ARIMA  $(p, d, q)$  makes reference to the order of the auto-regressive component ( $p$ ), the order of differencing ( $d$ ) and the order of the moving average (MA) component. With this structure, the most recent changes in the time series can be easily modeled (D. Kreuzer & et al., 2020) as well as capture smoothed trends in the data (T. Dimri & Sharif, 2020; M. Alsharif & et al., 2019). For stationary time series, the ARIMA  $(p, d, q)$  model can be written in terms of past temperature data, residuals and prediction errors (Zhou et al., 2020) as follows:

$$Y_t = -\left(\Delta^d Y_t - Y_t\right) + \phi_0 + \sum_{i=1}^p \phi_i \Delta^d Y_{t-i} - \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (18)$$

Where:

$d$ : Differences that are necessary to convert the original series to stationary.

$\phi_1, \dots, \phi_p$ : Autoregressive parameters of the model.

$\theta_1, \dots, \theta_q$ : Moving averages parameters of the model.

$\phi_0$ : Constant.

$\varepsilon_t$ : Error term.

$\Delta Y_t$ :  $Y_t - Y_{t-1}$

If the time series is non-stationary, we perform differencing to transform the series into a stationary model. We express a first-order differencing as follows:

$$y_t = x_{t-i} - x_t \quad (19)$$

If  $y_t$  is not stationary when  $d=1$ , then it is needed to difference using  $d-1$  times until it becomes stationary.

Among the main advantages of the ARIMA model is that it allows the user to choose the models it implements. It is a reliable model and is considered easy to implement. Even though we make extensive use of the ARIMA model for the purpose of prediction, this model does have a few significant drawbacks. One of these is the fact that the ARIMA model presupposes that input data is linear. Therefore, the ARIMA model tends to perform well only in short-term forecasting. Because of this, it is necessary to note that this model fails to capture the unexpected changes in the behavior patterns of the series due precisely to its linearity. Additionally, there is no established technique to find the parameters of each model, so the adequacy of the model is determined by testing the parameters of the equation of each model.

Multiple studies currently use this model for weather forecasting. For instance, in (M. Murat & et al., 2018) it is mentioned that a 30-year record of air temperature and precipitation in the cities of Jokioinen, Dikopshof, Lleida and Lublin were utilized to implement a weather forecast through an ARIMA model. In this case, the error per hour falls in the range of approximately 3.02 °C and 4.5 °C.

### 2.3.6 Convolutional encoder–decoder network

Convolutional Neural Networks (CNN) consist of multiple layers of convolutional filters of one or more dimensions. After each layer, a function perform non-linear causal mapping. Despite convolutional networks usually performs image processing however these structures can perform an interesting time series analysis. The feature extraction phase of this type of network resembles the stimulating process in the cells of the visual cortex. In other words, it seeks to recognize from colour or similar patterns or lines (first convolution) to identify complex composite elements. The distinctive processing of CNNs consists of performing close data convolutions, for example groups of pixels in an image, or information close to a value in time series, and mathematically operating them against groups of kernels (matrix of different sizes). This phase implies the use of alternating layers of convolutional neurons and down sampling neurons. As the data progresses through this phase, its dimensionality decreases, with neurons in far layers being much less sensitive to disturbances in the input data, but at the same time more capable to recognize complex features. In the field of weather forecasting, we can work with a One-Dimensional (1D) Convolutional Neural Network utilizing different approaches such as the arrangement called Encoder–Decoder (Sagheer & Kotb, 2018), see Figure 8. This approach allows to the Neural Network first, to learn the spatial correlations between the input data and second to interpret these correlations obtaining a pattern in the output of the arrangement. The CNNs automatically learn the representation of the features in the time series by assigning the correspondent weights to the elements in the image or in the time series in order to differentiate one from another. That is, the CNNs apply several convolutions independently on each of the network inputs  $x_1, x_2, \dots, x_{t-1}$  with  $x_t \in \mathbb{R}^n$  to learn the interactions between the different components of  $x_t$  (Y. Tao & et al., 2018). We successfully tested the Encoder–Decoder structure for time series analysis in two test scenarios. First using Bayesian uncertainty reduction (see section 4.2) and a novel ADAM approach (see section 4.3) related to decoupling weight decay with ADAML (Llugsi & et al., 2021b). Second, when we carried out a comparison between ADAM, ADAMAX and ADAMW (see section 4.3) optimizers to implement a reliable weather forecast based on Neural Networks for the Andean city of Quito in (Llugsi & et al., 2021a).

In Eq. (20) we present the 1D One-Dimensional discrete convolution process.

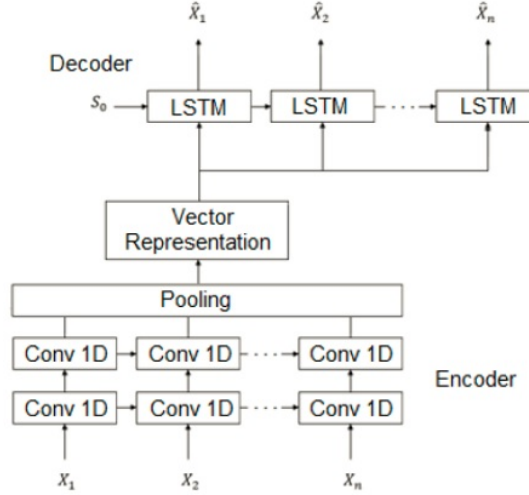


Figure 8: Encoder-Decoder structure.

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \quad (20)$$

It is necessary to take into account that the name encoder-decoder refers to the fact that a layer encodes the information in the form of a vector of fixed length. Once the feature detection process ends, another layer decodes the data to generate a prediction (Sagheer & Kotb, 2018), see Eqs. (21) and (22):

$$h(x) = f(W_1x + b_1) \quad (21)$$

$$\hat{x} = g(W_2h(x) + b_2) \quad (22)$$

Where:

$h(x)$ : Encoded (hidden) vector of input  $x$ .

$\hat{x}$ : Decoded (reconstructed) vector.

$f$ : Encoding function.

$g$ : Decoding function.

$W_1$ : Encoder weight matrix.

$W_2$ : Decoder weight matrix.

$b_1$ : Correction vector (Bias) of the encoder.

$b_2$ : Correction vector (Bias) of the decoder.

It is important to point out that there are additional methods for making predictions, such as the ARIMA technique. However, it is essential to keep in mind that ARIMA operates under the presumption that the data that is input into the model is linear. As a consequence of this, ARIMA models are unable to perform better than LSTM in situations in which the data change in a manner that was not anticipated, such as when

making a weather forecast under unusual circumstances (as discussed in (Y. Tao & et al., 2018) and in (I. Kamal & et al., 2020)).

## 2.4 Validation

You can estimate the error using the hold-out method when the quantity of data for training and testing is limited. This methodology aims to reserve a certain proportion of data for assessment and the remainder for training. As a recommendation, one-third of the data should be used for testing, while the remaining two-thirds should be used for training. There may be variations in the estimation of the error and the training and test data using this methodology. Cross-validation is an enhancement to the hold-out technique.

### 2.4.1 K-Fold Cross validation

Cross validation of  $k$  iterations (k-fold cross validation) is a widely used model evaluation method that aims to make effective use of all data for both training and testing. It allows using all available records for training and at the same time, uses many records as separate test suites. With cross-validation, first, a fixed number of partitions of the data is decided. Then the data is divided into the approximate number of partitions desired, and each in turn is used for testing, while the rest are used for training. Again, you can use two-thirds of the data for training and one-third for testing. The process repeats as many times as the size of the partitions and at the end, the methodology uses each partition once for testing.

The cross-validation of  $k$  iterations, or k-fold cross-validation, consists of dividing the original data into  $k$  subsets. At the time of training, the methodology uses each  $k$ -subset as the test set for the model and the rest of the  $k$ -subsets ( $k - 1$ ) as the training set. The cross-validation process will be repeated  $k$  times, and in each iteration, the methodology takes a different test set, with the rest of the data being the training set. At the end of the  $k$  iterations, the average of the precision and error results obtained for each test subset is calculated (Raschka, 2018). In k-fold cross-validation, the initial data is randomly partitioned into  $k$  mutually exclusive subsets,  $D_1, D_2, \dots, D_k$  each approximately the same size. Training and testing are performed  $k$  times. In general, some authors recommend at least a 10-fold stratified cross-validation to estimate precision due to its relatively low bias and variance (Witten & Frank, 2005). The main advantage is that the process uses all the observations for both training and validation, while the disadvantage is that K-fold does not work properly for time series data because it ignores their temporal relationship. For instance, it is not useful to train a model with temperature data from Wednesday and Friday to predict Thursday's weather.

### 2.4.2 Walk-Forward Validation

Adopting the classical approach of dividing the available data into training, validation and testing in order to validate the model implicitly biased the data validation towards the most recent period under analysis. This schema is not convenient if you want to investigate in depth meteorological information, for example, for the analysis of climate change. This happens because the environmental data are non-stationary time series. In meteorology, this is the typical scenario for regions of the planet close to the intertropical convergence zone. For instance, maybe the last day in Quito will bring a good forecast for the next day, but maybe it does not. On the other hand, maybe the first week of the month would hold some important patterns that would affect future days, but these would be lost in the approach mentioned above. The name "Walk Forward" refers to the use of a moving window that slowly walks through the entire period of historical data at a preset pace. The main goal of Walk Forward is to minimize over-optimized parameters throughout the model optimization process. Walk Forward Analysis does optimization on a subset of the training set that changes in size every iteration during the process. Namely, the goal of this technique is to minimize the curve fitting on the out-of-sample data by shifting a moving window. The train set is expanding each time step (tWL samples) and we fixed the test set one time step ahead (P. Ladyzynski & et al., 2013), see Eq. (23). Consequently, there are multiple out-of-sample periods and the process looks at these results combined at the end of every iteration (Narayanaa & Turhan, 2018).

$$TW = \sum_i^n tWL_i \quad (23)$$

Where:

TW: Testing Window.

tWL<sub>i</sub>: Window length for test per iteration.

The main advantage is that the model is updated at each time step with new data received. It implies that the algorithm provides a robust estimation. The main disadvantage of this schema is the additional computational cost generated for all the subset predictions. To carry out proper data analysis, we adopted a period of 24 hours.

## 2.5 Optimizers

As mentioned above, the goal of neural network training is to minimize the cost function by finding the minimum points of the function. To carry out this reduction, the training algorithm calculates the weights for the different inputs/characteristics entered into the network. Likewise, the numerical algorithm called backpropagation discovers these weights. The optimizer is in charge of generating better and better weights, that is, weights that allow responses with low levels of error (Perin & Picek, 2021).



The optimizer calculates the gradient of the cost function (partial derivative) for each weight (parameter/dimension) of the network. In addition, since we want to minimize the error, we will modify each weight in the (negative) direction of the gradient. To speed up the convergence of the cost function towards its minimum, we multiply the gradient vector by a factor called the Learning rate ( $\text{lr}$ ). Finally, we call the set of iterative methods for reducing the error function (search for a local minimum) optimization methods based on the descending gradient and batch size selection. The batch size consists of the number of examples that we introduce to the network in each iteration of the training process. If the number is small, it means that the network has a small amount of data in memory and trains faster. We have presented the theory related to the best-known optimizers until now, but it is important to mention that to stay focused on the research, we place emphasis on the theory related to ADAM.

### 2.5.1 Stochastic Gradient Descent (SGD)

The calculation of the partial derivative of the cost function with respect to each of the weights of the network for each observation is very complicated given the number of different weights and observations. Therefore, a first optimization consists in the introduction of a stochastic (random) behavior, (Perin & Picek, 2021). Stochastic Gradient Descent (SGD) does something as simple as limiting the derivative calculation to just one observation per batch. There are some variations based, for example, on selecting several observations instead of one (mini-batch SGD). A very useful variation is the introduction of momentum. The momentum accelerates the descent in directions similar to the previous ones. That is, it obtains the direction where the error decreases; see Figure 9(a). To do this, we are going to save a vector that represents the window mean of the previous descent vectors. If the new vector is similar to the momentum vector, we accelerate its descent. Some works present another variation, Nesterov's accelerated gradient optimizer. To operate this variation, we first calculate the descent, trusting the momentum vector, and once it has descended in its direction, we compute the new gradient from that point, see Figure 9(b).

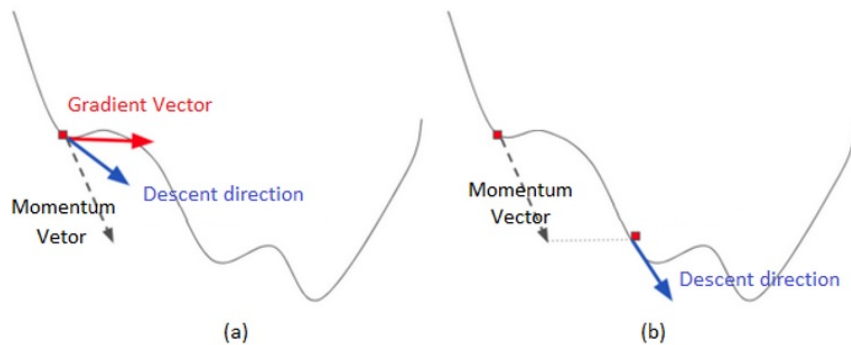


Figure 9: SGD optimizer with (a) acceleration, (b) acceleration with Nesterov approach.

### 2.5.2 Adaptative Gradient Algorithm (ADAGRAD)

Adaptive Gradient Algorithm (AdaGrad) is a modification of stochastic gradient descent. The AdaGrad algorithm introduces a variation in the training process. Instead of considering a uniform learning rate value for all weights, use a specific factor for each of them. In other words, it uses different learning rates for the variables, taking into account the accumulated gradient in each iteration (M. J. Uddin & et al., 2022).

### 2.5.3 ADAGRAD based on a moving window of gradient updates (ADADELTA)

Adadelata is a variation of AdaGrad in which a moving window of gradient updates is used. Instead of calculating the scaling of the learning rate of each variable taking into account the gradient accumulated from the beginning of the execution, it is restricted to a window of fixed size of the last  $n$  gradients (M. J. Uddin & et al., 2022).

### 2.5.4 Root Mean Square Propagation (RMSprop)

Root Mean Square Propagation (RMSprop) is an algorithm similar to Adadelata, namely, it maintains a different training factor for each variable. The core idea of RMSprop is to keep the moving average of the squared gradients for each weight and then divide the gradient by the square root of the mean square (Postalcioğlu, 2020). That is why we called it RMSprop (root mean square). Namely, instead of keeping an accumulation of the gradients, we use the window concept to consider only the most recent gradients. Consequently, RMSprop maintains that gradient squared estimate, but instead of letting that estimate continuously accumulate during training, it maintains a moving average of it.

### 2.5.5 Adaptive Moment Estimation (ADAM)

One of the optimization algorithms most commonly used today is Adaptive Moment Estimation (ADAM). This algorithm combines the benefits of AdaGrad and RMSProp. The learning rate structure is maintained per parameter, but in addition to calculating RMSProp, each rate is also affected by the mean gradient momentum. We summarize the ADAM operation as follows: First, we calculate the gradient at a certain place in the loss function to produce a vector. The magnitude of the vector is the maximum rate of change of the function at the point of analysis. The direction of the vector points at the maximum point of the function. Then the algorithm iteratively takes measurements in the opposite directions of the calculated gradients to find the lower area of the loss function, namely to implement the gradient descent operation (Kingma & Ba, 2015), see Eq. (24) and (25):

$$\Delta = -\gamma \nabla \quad (24)$$

$$\theta_+ = \Delta \quad (25)$$

Where:

$\gamma$ : Step size.

$\nabla$ : Gradient operation.

$\theta$ : Weight of a neuron-to-neuron connection in the network to be optimized.

$\Delta$ : Rate of change for  $\theta$  after each iteration of the algorithm.

Secondly, to improve the process of finding the minimum of the function, the algorithm adds the concept of momentum. This implies that in each iteration of the algorithm, the momentum adds the direction of the previous step. To avoid any problem related to an endless processing loop, the authors add a “decay rate” to slow the progression of the momentum calculation until it stops (Kingma & Ba, 2015), see Eqs. (26) and (27).

$$g_{\text{actual}} = g + \beta g_{\text{previous}} \quad (26)$$

$$\Delta = -\beta g_{\text{actual}} \quad (27)$$

Where:

$g_{\text{actual}}$ : Current sum of gradients.

$g_{\text{previous}}$ : Previous sum of gradients.

$\beta$ : Decay rate.

To compute the decaying averages of past and past squared gradients  $m_t$  and  $v_t$ , the algorithm performs the following, see Eqs. (28) and (29):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (28)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (29)$$

Where:

$m_t$ : First moment of gradients (mean).

$v_t$ : Second moment of gradients (uncentered variance).

$\beta_1, \beta_2$ : Decay rates.

The algorithm performs the biases by computing the bias-corrected first and second moments, as shown in Eqs. (30) and (31):

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (30)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (31)$$

Where:

$\hat{m}_t$  : First moment of gradients (biased).

$\hat{v}_t$  : Second moment of gradients (biased).

Finally, the core of the ADAM approach, the update rule, is established in Eq. (32):

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t \quad (32)$$

Where:

$\theta$  : Model weights.

$\eta$  : Learning rate.

$\varepsilon$  : Small scalar used to prevent division by 0 (typically  $10^{-8}$ ).

### 2.5.6 Adaptive Moment Estimation with infinite normalization (ADAMAX)

ADAMAX is an extension of the ADAM optimizer based on infinity normalization. Reviewing the gradient calculation process for ADAM and L2<sup>2</sup>, some researchers have detected that despite their differences, we can exploit their work together. In ADAM's update process,  $v_t$  scales the gradient inversely proportional to how it is done in L2 ( $v_{t-1}$  and  $g_t^2$ ), see (1.28). In (Kingma & Ba, 2015) the authors have adopted the process applied in L2 in ADAM, generalizing the process in the following way, see Eq. (33):

$$v_t = \beta_2^p v_{t-1} + (1 - \beta_2^p) |g_t|^p \quad (33)$$

The authors of this approach propose the convergence to the infinite normalization for the above in order to obtain a more stable value in the following way, see Eq. (34):

$$v_t = \beta_2^\infty v_{t-1} + (1 - \beta_2^\infty) |g_t|^\infty \quad (34)$$

Finally, in order to reach the convergence of the optimization process, the algorithm modifies the ADAM rule by adding the square of the biased second moment of gradients,  $\sqrt{\hat{v}_t} + \varepsilon$  to the equation of the second moment of gradients,  $v_t$ . Thus the new ADAMAX rule is, see Eq. (35):

$$\theta_{t+1} = \theta_t - \frac{\eta}{v_t} \hat{m}_t \quad (35)$$

---

<sup>2</sup>In the L1 (Lasso) regularization, the complexity  $C$  is measured as the mean of the absolute value of the model coefficients, while in the L2 (Ridge) regularization, the complexity is measured as the mean of the square of the model coefficients. A low dense solution is sought in L1 regularization. That is, it is desired that some of the coefficients end up having a value of 0. On the other hand, L2 causes the coefficients to end up smaller. The reduction of the coefficients minimizes the effect of the correlation between the input attributes and makes the model generalize better (X. ZongBen & et al., 2010).

### 2.5.7 Adaptive Moment Estimation with Decoupled Decay Regularization

The Optimizer with Decoupled Decay Regularization (ADAMW) optimizer seeks to insert a regularization based on the decay of weights in the optimization process of neural networks. The regularization consists of adding a penalty to the loss function to produce simpler models. This process occurs when new data is available. The authors of the methodology propose this approach, considering that regularization techniques such as L2 and weight decay behave adequately for standard stochastic gradient descent but not for adaptive gradients like ADAM (Loshchilov & Hutter, 2019). The weight decay approach is used to prevent the weights  $\theta_t$  from growing too much during their update process, this is achieved by multiplying the parameter by a decay rate  $\omega_t$ , as presented in Eq. (36) (Loshchilov & Hutter, 2019).

$$\theta_{t+1} = \omega_t \theta_t \quad (36)$$

We can commonly implement weight decay by regularizing or modifying the value of the calculated gradient. However, this process is not valid for adaptive gradient algorithms since the regularization tends to vary. We can verify the above by analyzing the update of weights when an optimizer runs on the Loss Function  $f_t(\theta)$  with and without decay of weights. In the first case, the weight is updated considering  $\theta_{t+1} \leftarrow (1 - \lambda)\theta_t - \alpha M_t \nabla f_t(\theta_t)$ , while in the second case, the calculated weight is updated with  $\theta_{t+1} \leftarrow \theta_t - \alpha M_t \nabla f_t(\theta_t)$ . Therefore, for the optimization process, there is no coefficient  $\lambda$  such that when executed without weight decay, it is equivalent to executing the optimizer in  $f_t(\theta_t)$  with the mentioned decay.

We can implement the decoupled weight decrease technique (Loshchilov & Hutter, 2019) to solve the problem described above and generalize the gradient update process in ADAM, see Eq. (37):

$$\theta_t \leftarrow \theta_{t-1} - \eta_t \left( \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \lambda \theta_{t-1} \right) \quad (37)$$

Taking the above into account, we can modify the ADAM optimizer to consider the decay of weights (ADAMW) as follows, see Eqs. (38) to (42):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (38)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (39)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (40)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (41)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t - \eta \omega_t \theta_t \quad (42)$$

Where:

$m_t$  : Update (with adjustment) of the 1st moment estimate.

$v_t$  : Update (with adjustment) of the estimate of the 2 nd moment (without processing).

$\hat{m}_t$  : Estimate calculated from the 1st moment, with corrected adjustment.

$\hat{v}_t$  : Estimate calculated from the 2 nd moment (without processing) with corrected adjustment.

$\beta_1$  : Decay rate of the 1 st moment.

$\beta_2$  : Decay rate of the 2 nd moment.

$g_t^2$  : Dot product (elementwise square)  $g_t \odot g_t$ .

## 2.6 Tools

In the following, the tools and methodology that are used for the automatic adjustment of the measurements of environmental parameters are broken down in detail. Analyses are performed on the instruments that are used to estimate the performance of a particular model and evaluate its fit. The methodology for determining the strength of a link between two variables is dissected here. At long last, the process that can be used to mitigate the unpredictability of a neural network model has been uncovered.

### 2.6.1 Error Metrics

When using a neural network for predictive modeling, we implicitly predict a true or false numerical value. This approach is different from classification, which involves predicting a class label. Unlike classification, we cannot use the accuracy of classification to evaluate the predictions made by the model. Instead, we should use error metrics specifically designed to evaluate the predictions made by the model, for instance, to quantify the difference between the model prediction and the actual value acquired by a sensor. We can analyze the ability or performance of a predictive model by considering the error in its predictions. However, keeping in mind that if we are predicting a numerical value, we do not want to know if the model predicted the exact value, as this can be incredibly difficult. Instead, we want to know how close the predictions were to the expected values. Error analysis addresses exactly this and summarizes on average how close the predictions were to their expected values. The most commonly used error metrics are mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

It is necessary to say that in this work no confusion matrices or precision and delay graphs were included. This decision was made because, although it is true that confusion matrices are considered a tool to determine if the predictions were correct, we do not work with classification models. To verify the correct operation of the neural networks, we verify the correct performance of the model with learning function curves. This parameter was not included in the work because experimentally obtaining the curves and calculating the precision of the model slowed down the simulation.

*Mean Squared Error (MSE)* One of the most commonly used error metrics is the Mean Squared Error (MSE). It represents the average of the squared errors, namely the difference between the actual value and the estimated value. It is also an important loss function for algorithms fitted or optimized using the least-squares framework of a regression problem. Since this metric considers the squares of the estimated and actual values, the MSE is the second moment of the error and therefore incorporates the variance of the estimator as the basis for calculating the error. Consequently, the higher this value, the worse the model; see Eq. (43) (Naser & Alavi, 2021).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2 \quad (43)$$

Where:

$\hat{x}_i$  : Estimated time series.

$x_i$  : Current time series observations.

$n$ : Non-missing data point number.

*Root Mean Squared Error (RMSE)* Similar to the square root of a variance, the Root Mean Squared Error (RMSE) can be interpreted as the standard deviation of the unexplained variance, and it possesses the advantageous property of having the same units as the response variable. Consequently, the RMSE represents the square root of the mean square distance between the actual and estimated values. Consequently, it permits determining how concentrated the data are around the line of best fit and, consequently, whether or not the model used to estimate a particular parameter is functioning correctly. See Eq. (44) for the relationship between RMSE and fit (Naser & Alavi, 2021).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2} \quad (44)$$

*Mean Absolute Error (MAE)* Mean Absolute Error (MAE) is the average of the absolute errors resulting from a comparison of two data series. The mean absolute error is a linear score, meaning that all individual deviations contribute equally to the average. MAE is therefore more resistant to outliers and penalizes errors less severely than MSE. The mean absolute error employs the same scale as the measured data;

therefore, this error metric is a scale-dependent precision metric, see Eq. (45) (Naser & Alavi, 2021).

$$\text{MAE} = \frac{\sum_{i=1}^n |x_i - \hat{x}_i|}{n} \quad (45)$$

*Mean Absolute Percentage Error (MAPE)* The Mean Absolute Percentage Error (MAPE) error metric expresses precision as a percentage. MAPE can therefore be simpler to comprehend than other error metrics for measuring accuracy. It is essential to note that even though the model appears to adequately suit the data, occasionally we can observe high MAPE values. This is the result of the calculation of percentages that scale the resulting value relative to 100. In general, values close to zero can significantly increase the MAPE; see Eq. (46) (Naser & Alavi, 2021).

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^n \left| \frac{x_i - \hat{x}_i}{x_i} \right| \quad (46)$$

### 2.6.2 Correlation

Correlation is a type of association between numerical variables where we evaluate the trend (increasing or decreasing) in the data. Two variables are associated when one variable gives information about the other. On the contrary, when there is no association, the increase or decrease of one variable does not say anything about the behavior of the other variable. Among the well-known types of correlation, we can mention the following (P. Schober & et al., 2018):

- Pearson correlation: Evaluates the linear relationship between the raw data values of two continuous variables (in a range from  $-1$  to  $+1$ ) (Sedgwick, 2012).
- Spearman correlation: Works with rank-ordered variables.
- Kendall rank correlation: Works with rank-ordered variables and presents a low error sensitivity and a smaller asymptotic variance (AV).
- Point-Biserial correlation: We use this correlation to measure the strength and direction of the association between one continuous variable and one dichotomous variable.

When we compare two time series, the Pearson correlation is the best option because the value is independent of whatever unit we use to measure the variables. Additionally, if the sample is large, the accuracy of the estimate is likely to be higher.

$$r_{xy} = \frac{n \sum x_i y_i - \sum x_i - \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \quad (47)$$

Where:



$r_{xy}$  : Pearson  $r$  correlation coefficient  $x$  and  $y$ .  
 $n$ : Number of observations.  
 $x_i$  : Value of  $x$  (for  $i$ th observation).  
 $y_i$  : Value of  $y$  (for  $i$ th observation).

### 2.6.3 Dynamic Time Warping

Dynamic Time Warping (DTW) is an algorithm to measure the similarity between two temporal sequences that can vary in speed or length. The technique is useful when one is willing to find a low distance score between signals (P. Tormene & et al., 2008), see Figure 10. This technique is widely applied to deal with temporal variation

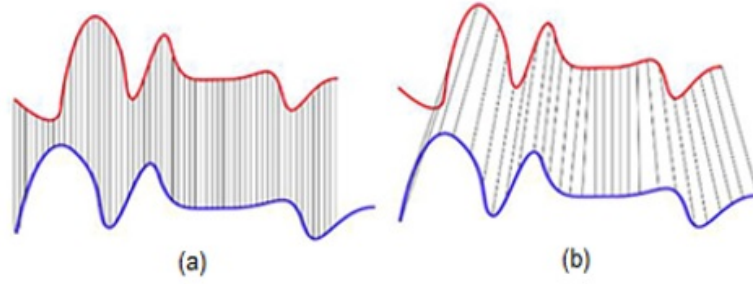


Figure 10: (a) Euclidean Matching, (b) Dynamic Time Warping Matching.

(speech recognition). Because the voice signal has considerable randomness. Even if the same person sends the same sound at different times, it is impossible to have a full length. In addition, the speed of pronunciation of the same word is also different. Since DTW-spaces present mathematically less structure than metric spaces (Giorgino, 2009), we cannot consider DTW-distance as a metric. At the core of the technique lies the warping curve  $\varnothing(k), k = 1 \dots T$ .

$$\varnothing(k) = (\varnothing_x(k), \varnothing_y(k)) \quad (48)$$

Given  $\varnothing$  we can calculate the average cumulative distortion between the  $X$  and  $Y$  Warped time series as follows:

$$d_{\varnothing}(X, Y) = \sum_{k=1}^T \frac{d(\varnothing_x(k), \varnothing_y(k)) m_{\varnothing}(k)}{M_{\varnothing}} \quad (49)$$

Where:

$m_{\varnothing}(k)$  : Per-step weighting coefficient.

$M_{\varnothing}$  : Normalization constant of  $m_{\varnothing}(k)$ .

Finally, the core of the DTW process is to find the optimal alignment as follows:

$$D(X, Y) = \min d_{\varnothing}(X, Y) \quad (50)$$

#### 2.6.4 Methodology

For the present study, we will work with time series and neural networks. We will collect the numerical data through sensors to rank, measure, or categorize it through statistical analysis. We will use all of this information to predict patterns and relationships by reducing uncertainty through Bayesian estimators. Regarding the calibration of the stations, it is necessary to comment that a laboratory-certified station was built after the initial acquisition of information from the network of AWS in the period from March to May 2020. This decision was made due to the high cost of calibrating the stations, which was around USD 400 per station. To guarantee the correct acquisition of information in the period mentioned above, the calibration of the stations was carried out with a portable, certified device both during the assembly of the stations, as well as during the periodic maintenance of the stations. Considering that the traceability carried out was not used for the publications, we did not include the calibration reports for the paper publications, but these reports are added in the Appendix II section.

Since we seek to implement a predictive method, it is necessary to generate a process that involves the use of neural networks, statistics and mathematics in order to quantify the research problem. To carry out the above, it is necessary to structure the collection and analysis of information, considering that the acquisition of information will be of a meteorological nature. Therefore, it is necessary to consider the uncertainty present in the information acquired from the automatic station network. Uncertainty always appears to be associated with the measurement of magnitudes. Consequently, we can say that uncertainty is a quantitative measure of the quality of the measurement result. Additionally, it allows us to compare the measurement results with other results, references, specifications, or standards. To determine the uncertainty of the model generated in the investigation, we will use Bayesian inference. Bayesian inference is a type of statistical inference in which we use evidence or observations to update or infer the probability that a hypothesis might be true. The name Bayesian comes from the frequent use of Bayes' theorem during the inference process. Finally, considering that neural networks are parametric algorithms, it is essential to work with the appropriate optimizer. An optimizer seeks to optimize parameter values to reduce the error made by the network. Bearing in mind the above, the quantitative research methodology has been chosen to offer generalizable conclusions about the prediction process of meteorological variables. This research is useful to find how much, how often, or to what extent the variation of, for example, the temperature affects the weather of the Andean city of Quito.

### 3 Challenges of the Weather of Quito

This chapter provides a general overview of the climate that can be expected in the city of Quito. The methodology that was supposed to be used to carry out the distribution of stations in order to get environmental information is presented here. In addition to this, we describe the technique that was utilized in the implementation of the stations as well as the handling of the information that was gathered. Finally, we present the exploratory research that was conducted in order to develop the weather forecast models and the conclusions that were drawn from that research. This section corresponds to the published article *Deep Learning to Implement a Statistical Weather Forecast for the Andean City of Quito*, DOI: 10.1109/ANDESCON50619.2020.9272106, published in the IEEE ANDESCON 2020 International Conference held in Ecuador (see Appendix) and incorporates some additional details.

#### 3.1 Quito Weather

Quito, located on two slopes of the Pichincha volcano, blends into the Andean landscape, at an altitude of 2800 m.a.s.l. Its approximate dimensions are 50 km long in a south-north direction and 4 km wide from east to west (Unesco, 2020). It is located at the foot of the active volcano Guagua Pichincha and is populated with approximately 2.7 million inhabitants, thus becoming the most populous city in Ecuador (Comercio, 2020). The city is divided in its central part by the hill of El Panecillo (3035 m.a.s.l.) and to the east by the hills of Puengasí, Guanguiltagua and Itchimbía (Unesco, 2020). This geography makes it almost impossible to properly forecast the weather in the city due to the creation of microclimates. This is almost the same scenario for the rest of the cities located in the Andean region of South America. Currently, government institutions, such as the National Institute of Meteorology and Hydrology of Ecuador (INAMHI), acquire environmental data in the Andean city of Quito using CWS and AWS. Since its creation in 1961, INAMHI has been responsible for the establishment, operation and maintenance of the hydrometeorological station network of the country (Universitario, 2020). With the purpose of carrying out a proper measurement of climate parameters, INAMHI has installed three conventional stations (WMO, 2008) in Quito: Iñaquito, Izobamba and La Tola. These stations are expensive; kilometers apart from each other and in consequence do not allow the acquisition of proper data for further applications (i.e., forecast).

The deployment of these stations is done in large areas, which implies that the weather behavior in the zone is not represented accurately (S. Serrano & et al., 2017). Mathematical models running in regional offices of the World Meteorological Organization (WMO) (J. Cullman & et al., 2019) process the meteorological data acquired. How-

ever, it is crucial for these models to work with environmental data in real time to implement a proper weather forecast. New approaches such as low-cost small single-board computers configured as AWS are becoming more popular because of specific advantages such as portability, real-time data acquisition/ transmission and internet connectivity (Muck & Homam, 2018). Additionally, new weather forecast techniques currently use machine learning for time series (E. Abrahamsen & et al., 2018; Zaytar & Amrani, 2016) and satellite imagery (J. Frnda & et al., 2019; Scher & Messori, 2019). Thus, the fast development of neural networks and deep learning paradigms combined with the new low-cost hardware enhancements allows the creation of powerful processing platforms. We can use this strategy to obtain an accurate and inexpensive weather forecast that was unthinkable 12 years ago (G. Zhang & et al., 1998).

### 3.2 Distribution of the Stations

As mentioned at the beginning of this work, the country's proximity to the Intertropical Convergence Zone and the rugged geography of the Andean region make weather forecasts in Ecuador difficult to obtain. The foregoing is fully exemplified in the case of the Andean city of Quito (2,800 m.a.s.l.), since the Andes Mountain Range crosses exactly latitude  $0^\circ$  (R. Llugsí & et al., 2020a), see Figure 11.

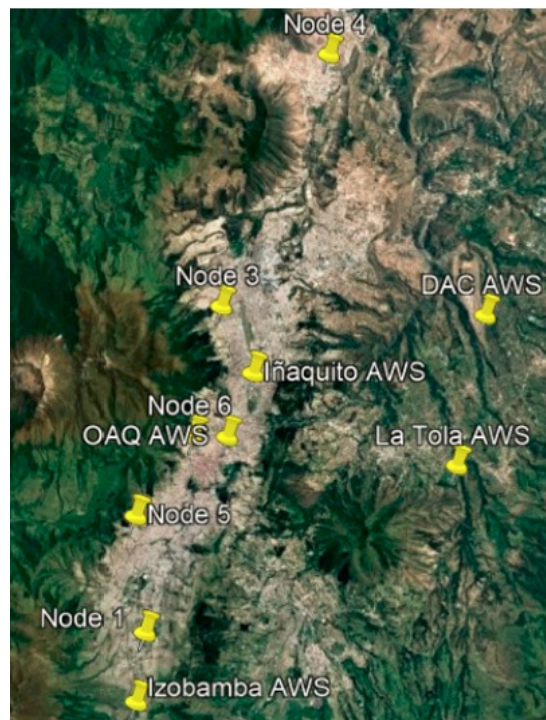


Figure 11: Geographic position of the AWS on a topographic map. Note: Node 2 was omitted because it was utilized for operational verification only.

INAMHI has installed three meteorological stations to measure environmental parameters in the city of Quito. Additionally, the Quito Astronomical Observatory

(OAQ) and the Directorate of Civil Aeronautics (DAC) have also installed stations to carry out this task (Villacis & Marrero, 2017). The three stations that INAMHI has installed in Quito are Iñaquito, Izobamba and La Tola, located at the north-central part of the city, the southern part of the city, and the Tumbaco Valley at the south-eastern part of Quito respectively. OAQ and DAC operate two more AWS installed in the central part of the city and at the city airport. For this work, we did not consider the information from these stations because there is currently no inter-institutional framework agreement that allows this interaction. However, the information from these stations has served, apart from a secondary standard sensor, to verify the acquired values of temperature and humidity. It is reasonable to think that there is a high probability that the number of stations installed by government institutions in Quito is not enough to analyze the weather in the city. Thus, for this research, we have implemented and installed a network of five low-cost automatic stations along the Andean city of Quito, considering the areas where there is a high probability of finding maximum weather effects (R. Llugsí & et al., 2020a). AWS 1 is located in the southern part of the city. In this location, there is a strong exchange of moisture by the vegetation, so a high tendency to rain is expected. The AWS 3 and AWS 5 stations are located on the north-east and southeast sides of Quito, exactly on the slopes of the Pichincha volcano. It is expected that the convective process formed by the difference in temperature between the city and the Pichincha volcano (4784 m.a.s.l.) will allow us to analyze to what extent the climatology of the city is affected by this geographical elevation. AWS 6 is located on the east side of the Pichincha volcano in the central part of the city, near the elevation called El Panecillo (3000 m.a.s.l.). Station AWS 4 is located in the farthest part of the city, to the north of Quito. We have selected this location because it is predominantly dry and allows us to analyze in a basic way the similarities between the meteorology of this area and that of other sectors of the city. The AWS 4 station is located 400m lower in altitude than the average of the other stations installed throughout the city. This precisely confirms the unevenness of the geography of the city of Quito. Finally, we have excluded AWS 2 from the analysis, because we installed the station near AWS 6 for reasons of operational verification, so it does not provide relevant information for the research. The location and installation of the stations were carried out taking into account the official recommendations generated by the United Nations specialized agency for meteorology, the World Meteorological Organization (WMO, 2008; Oke, 2006). We selected the location with the aim of demonstrating the strong climatic variation experienced in the city (see Table 1).

We implemented the AWS with Raspberry Pi modules, which can be considered low-cost small single-board computers. AWS collects temperature, humidity, and pressure information every 20 seconds. The program loaded on the Raspberry Pi averages the information every minute as recommended by the WMO in (WMO, 2008). The pro-

Table 1: Geographical position of the AWS network.

AWS	Latitude	Longitude	Altitude (m)
1	0°19'34.40"S	78°32'59.88"W	3014.5
3	0°8'25.32"S	78°30'20.32"W	2917.6
4	0°0'14.42" N	78°26'35.60"W	2408.4
5	0°15'34.89"S	78°33'18.17"W	2909.9
6	0°12'50.66"S	78°31'18.38"W	2976.1
AWS Ñaquito	0°10'42.06"S	78°29'15.66"W	2792.0
AWS La Tola	0°13'54.38"S	78°22'13.50"W	2495.0
AWS Izobamba	0°21'57.09"S	78°33'18.69"W	3064.0
AWS OAQ	0°12'52.92"S	78°30'9.39"W	2822.0
AWS DAC	0°8'43.58"S	78°21'12.65"W	2414.0

gram saves the information in the internal memory of the Raspberry Pi. To work with the data acquired by the stations, the program transmits the information stored in internal memory to an FTP server located in the cloud using the Raspberry Pi's WIFI (Wireless Fidelity) card. In order to corroborate the quality of the information acquired, we have calibrated the stations using the secondary calibration standard methodology (that is, we reference the measurements to a sensor calibrated in a laboratory) with the PEAKMETER MS6508 Digital Temperature and Humidity Meter. Then, we analyze and correct the data in a stage prior to entering the neural network. The tasks carried out in this stage were polynomial interpolation and correction of abnormal measurements to ensure the length of the time series and the validity of the data acquired. It is necessary to highlight that the stations do not have an internal battery because we considered that we were going to connect them to safe power outlets and because of the increase in the cost of implementing the stations. Taking the above into account, we applied the interpolation to obtain data for, in some cases, hours due to failure of operation due to power outages or failure in the execution of the information acquisition program. In addition, the interpolation task was carried out in order to correct abnormal meteorological values (peaks) produced at the time of storage.

### 3.3 Forecast Methodology

To develop the short-term weather forecast model with neural networks, we used the data collected by the AWS network we installed in Quito. The information used was the one collected by the AWS in the period between March 18, 2020 and May 18, 2020 (temperature and humidity). Bearing in mind that the temperature corresponds to a stationary time series, the tasks performed at this stage were: polynomial interpolation, carried out to predict the lack of sampled points in the time series; and detection and suppression of spikes. When analyzing the information on temperature and humidity, we can say that it tends to be stationary since it shows "relatively" constant variations. By this, we mean, for instance, the variation in temperature during the day and at night. We present the seasonality analysis in detail in Chapter 5.

We do not apply any type of transformation to the data series, such as Fourier, to extract frequency components, since we do not seek to truncate the series and determine a cyclic behavior in the data collected. In the same way, we did not utilize normalization since in the initial tests carried out, there was no major difference in the learning time of the network with or without normalization.

We apply the following conditions to carry out the simulations with the networks: 1) The walk-forward technique for validation 2) A down sampling stage to resample the data in 1-hour steps (instead of 1-minute steps). Sigmoid, Tangent Hyperbolic (Tanh) and Rectified Linear Unit (ReLU) activation functions were tested and from these tests, it was found that the ReLU function presents the best behavior of the neural network (low error values and successful prediction). We can verify the proper performance of this activation function in several works related to forecasting models (E. Abrahamsen & et al., 2018; Huang & Kuo, 2018).

Next, it looks for the best way to make use of the data that AWS has collected to produce the most accurate temperature forecast. To meet this objective, we developed four test scenarios. The first scenario (Case 1) evaluates the temperature prediction for an AWS under analysis considering the past temperature values of the same station. In the second scenario (Case 2), we implement the temperature prediction using the historical temperature values of all the stations in the network. The third scenario (Case 3) seeks to implement the temperature forecast for the AWS under analysis, considering the temperature and humidity values of the same station. Finally, in the fourth scenario (Case 4), we sought to implement the temperature prediction for the AWS based on the temperature and humidity values of all the stations in the network.

### 3.4 Experimentation

With the purpose of implementing models that are efficient and that do not excessively consume computational resources, we use a CPU device with eight cores,

2.10 GHz core clock, and 16 GB of memory size. We use the Python programming language in conjunction with the Keras library to create a simulation of the neural networks that we design. During the simulation of the models, the simulation jobs utilized 70% of the capabilities of the central processing unit (CPU) and 45% of the RAM available. We configured the following parameters for the ADAM optimization algorithm: Learning rate = 0.001,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 1e-08$ . Additionally, we use a data distribution of 70%, 15%, and 15% for the training, validation and test data, respectively. We have investigated and assessed a variety of frameworks for the purpose of putting together time series-based weather forecasts. These structures were divided into three distinct sections:

- Stacked LSTM/GRU: Composed of two layers stacked LSTM (NN1), two layers stacked GRU (NN2), three layers stacked LSTM (NN3) and three layers stacked GRU (NN4).
- Encoder-Decoder LSTM/GRU: To detail the structures Encoder-Decoder LSTM (NN5) and Encoder-Decoder GRU (NN6).
- Encoder-Decoder Convolutional LSTM/GRU: To detail the structures Encoder-Decoder Convolutional LSTM (NN7) and Encoder-Decoder Convolutional GRU (NN8).

Once, we have specified the simulation parameter, then, the appropriate number of hidden units for every Network is determined. In order to do so, we have conducted several trials considering the four scenarios described above and different models. In Tables 2 and 3, we present the description of the Neural Networks evaluated for the temperature forecast.

Table 2: Number of hidden units per layer (L1, L2 and L3) (I).

NN 1 / NN2	L1: 25 – L2: 50, L1: 50-L2: 100, L1: 75-L2: 150, L1: 100-L2: 200,
	L1: 25 – L2: 25, L1: 50 – L2 : 50, L1: 75 – L2 : 75, L1 : 100 – L2 : 100,
	L1: 200 – L2 : 200, L1: 300 – L2 : 300, L1 : 400 – L2 : 400, L1 : 500 – L2 :
	500, L1: 200 – L2 : 100, L1 : 200 – L2 : 100, L1 : 150 – L2 : 75, L1 : 100 – L2: 50, L1: 50 – L2 : 25.

We have computed the RMSE, MSE, MAE and MAPE to establish the efficiency of the models through the lowest error (I. Kamal & et al., 2020). It must be



Table 3: Number of hidden units per layer (L1, L2 and L3) (II).

NN 3 / NN4	L1 & L2: # units of NN1 / NN2 that obtain the lowest MAPE value.
	L3: 25, 50, 75, 100, 200, 300, 400, 500.
NN 5 / NN6	L1: 25 – L2: 50, L1: 50-L2: 100, L1: 75-L2: 150, L1: 100-L2: 200,
	L1: 25 – L2: 25, L1: 50 – L2: 50, L1: 75 – L2: 75, L1: 100 – L2: 100,
NN 7 / NN8	L1: 200 – L2: 200, L1: 300 – L2: 300, L1: 400 – L2: 400, L1: 500 – L2: 500,
	L1: 200 – L2: 100, L1: 200 – L2: 100, L1: 150 – L2: 75, L1: 100 – L2: 50, L1: 50 – L2: 25.
Filters: 64, Layer: 25, 50, 75, 100, 200, 300, 400, 500.	

clarified that since the walk-forward validation was applied, the error metrics were calculated between the predicted and untrained test data for every step forward (namely, hourly) in order to find which hours were easier to forecast than others. In general, stacked LSTM/GRU with three layers were the neural networks that spent the most time processing. This is justified by the usually high number of units utilized in each layer of these models. We present the particularities of the trials for all the AWS of the network with lower error results in Table 4.

### 3.4.1 Discussion

We obtain the best prediction accuracy with the encoder-decoder and encoder-decoder convolutional GRU and stacked LSTM (with two layers). We can explain this outcome for the encoder-decoder structures thanks to their better capabilities to catch the features in the time series with a more complex structure. While for the stacked LSTM, the LSTM's memory capabilities play an important role in the prediction. Curiously, the AWS 4 presents a higher usage of hidden units (300 units) when it works with a convolutional stage. This is due to the location of AWS 4, which is closer to latitude 0° and probably the convective behavior of the zone, which affects the weather severally.

The AWS 3 was the station that shows the best prediction accuracy in the network with the lowest error values of RMSE (0.86 °C), MSE (0.74 °C), MAE (0.75 °C) and MAPE (5.17%). This is indeed contradictory because it is located at the base of the volcano Pichincha. However, this shows intrinsically that there are areas near the volcano that, despite the convective processes due to the height difference, have a stable climate. Since the error appears to be approximately between 0.01°C and 4°C, the outcomes presented in this work resemble the hourly error values obtained in (E. Abrahamsen & et al., 2018; Zaytar & Amrani, 2016). Namely, we obtain similar results with the

Table 4: Networks with best prediction accuracy (per error metric).

AWS	TYPE	ERROR ( °C )	CASE (NN)	UNITS PER LAYER (UL)
1	RMSE	1.37	2 (5)	UL1: 75, UL2: 150
1	MSE	1.88	2 (5)	UL1: 75, UL2: 150
1	MAE	1.06	2(5)	UL1: 75, UL2: 150
1	MAPE	6.11	2(7), 3(1)	Filters: 64, UL: 75, UL1: 50, UL2: 100
3	RMSE	0.86	2(6)	UL1: 25, UL2: 50
3	MSE	0.74	2(6)	UL1: 25, UL2: 50
3	MAE	0.75	3(6)	UL1: 150, UL2: 150
3	MAPE	5.17	1 (1)	UL1: 500, UL2: 500
4	RMSE	1.45	2(6, 8)	Filters: 64, UL: 300, UL1: 50, UL2: 100
4	MSE	2.10	2(6, 8)	Filters: 64, UL: 300, UL1: 50, UL2: 100
4	MAE	1.20	2(8)	Filters: 64, UL: 300
4	MAPE	5.41	2(8)	Filters: 64, UL: 300
5	RMSE	1.05	2 (1)	UL1: 150, UL2: 150
5	MSE	1.11	2 (1)	UL1: 150, UL2: 150
5	MAE	0.92	1(2), 2(1), 2(6)	UL1: 100, UL2: 200, UL1: 150, UL2: 150, UL1: 25, UL2: 50
6	RMSE	2.24	1 (8)	Filters: 64, UL: 50
6	MSE	5.03	1 (8)	Filters: 64, UL: 50
6	MAE	1.62	1 (3)	UL1: 50, UL2: 100, UL3: 50
6	MAPE	8.90	1 (3)	UL1: 50, UL2: 100, UL3: 50

application of other techniques for weather forecasting.

## **4 The Bayesian approach and the Adaptive Moment Estimation to reduce uncertainty**

In this chapter, we discuss Bayesian inference as a basis for understanding the process of calculating, adjusting uncertainty through estimators and implementing an optimizer for the learning task of the neural network. We show how to utilize the dropout technique to implement the calculation of the Bayesian uncertainty in any type of neural network. To verify the effectiveness of the uncertainty approach, we propose some test scenarios, conduct the respective experimentation, analyze the results and generate the respective conclusions. Next, we carry out an explanation of non-convex optimization as an introductory mechanism to introduce the weight decay approach to perform the optimization process. We present the weight decay methodology assumed to carry out the investigation and the respective mathematical demonstration of the convergence of the optimizer. Finally, we present the experimentation carried out to verify the operation of the optimizer, the discussion of results and the conclusions. This section corresponds to three published articles: 1) A novel ADAM approach related to decoupled weight decay (ADAML), DOI: 10.1109/LA-CCI48322.2021.9769816, published in the IEEE LACCI 2021 International Conference held in Chile and incorporates some additional details. 2) Uncertainty Reduction in the Neural Network's Weather Forecast for the Andean City of Quito through the Adjustment of the Posterior Predictive Distribution Based on Estimators, DOI: 10.1007/978-3-030-62833-8\_39, published in the TICEC 2020 International Conference held in Ecuador. 3) A novel encoder-decoder structure for time series analysis based on Bayesian uncertainty reduction, DOI: 10.1109/LA-CCI48322.2021.9769850, published in the IEEE LACCI 2021 International Conference held in Chile and incorporates some additional details.

### **4.1 Definition of Uncertainty**

Uncertainty is a concept that expresses the degree of ignorance about a future condition and may imply an imperfect predictability of the facts. That is an event for which the probability of its occurrence is unknown. From a statistical point of view, this means that it is impossible to determine with complete certainty the causes that lead to a specific effect. Therefore, we can only study it through randomness and probabilities. Uncertainty has negative implications for different types of activities, for example, the correct prediction of investment processes or an adequate weather forecast. We can treat the uncertainty also from the discipline that is responsible for considering decision-making. Indeed, this type of circumstance is extremely relevant when following one path or another in a given project. In our research, stochasticity has been the key to determining the uncertainty of the models (Neal, 1996; Gal &

Ghahramani, 2016). This approach allows us to improve the learning of the neurons in the network since the injection of stochastic noise into the model constitutes itself as a regularization technique, in this case of the stochastic type. Dropout is currently one of the most popular stochastic techniques in neural networks since it avoids overfitting in parameterization by eliminating neurons within the network learning process. We can check the effectiveness of the dropout approach by reviewing the behavior of a feed-forward neural network (with a single hidden layer). In (Gal & Ghahramani, 2016) the authors suggest that we can utilize the dropout as a basis to adopt a Bayesian behavior and therefore, its uncertainty can be determined. We can verify this approach experimentally by comparing the optimization for the cost function of a Bayesian neural network and the optimization of a network using dropout regularization.

## 4.2 Bayesian Inference

We can model uncertainty through probability. Bayes' theorem (Willink & White, 2012) describes a relationship between conditional probabilities that is very useful to obtain a probabilistic approach for the determination of uncertainty.

In Bayesian regression, given the training inputs  $x_1, \dots, x_N$  that produces the outputs  $y_1, \dots, y_N$ , the aim is to infer parameters  $\omega = (W_{xi}, W_{xf}, W_{xc}, W_{xo}, W_{hi}, W_{hf}, W_{hc}, W_{ho})$  (in the case of LSTM) of a function  $y = f^\omega(x)$  that are likely to generate the outputs. As a result, if a dataset  $X$  and a function  $Y$  are given, the function's posterior distribution can be described as follows in Eq. (51) (Gal & Ghahramani, 2016):

$$p(\omega | X, Y) = \frac{p(Y | X, \omega)p(\omega)}{p(Y | X)} \quad (51)$$

Where:

$p(\omega)$  : Prior distribution of the degree of belief of the parameter  $\omega$  before dataset acquisition.

$p(Y | X, \omega)$  : Likelihood function, which indicates the probability of obtaining certain results in the data set with the value  $\omega$ .

$p(\omega | X, Y)$  : Posterior distribution representing the state of knowledge of  $\omega$ , after new information has been acquired.

We can rewrite the Eq. (52) assuming that the probability distributions follow a continuous function as (Gal & Ghahramani, 2016; Willink & White, 2012):

$$p(\omega | X, Y) = \frac{p(Y | X, \omega)}{\int p(Y | X, \omega)p(\omega)d\omega}p(\omega) \quad (52)$$

By evaluating the integral in Eq. (53), first, we can say that the likelihood is marginalizing over  $\omega$  (marginal likelihood). Consequently, we can consider the denominator as a normalizing factor (model evidence). Therefore, we can rewrite the above equation

as follows (Gal & Ghahramani, 2016):

$$p(\omega | X, Y) \triangleq p(Y | X, \omega)p(\omega) \quad (53)$$

Finally, we can obtain a prediction of the outputs through the integration of the above relation every time we enter new data in the model, as follows (Gal & Ghahramani, 2016):

$$p(y^* | x^*, X, Y) = \int p(y^* | x^*, \omega) p(\omega | X, Y) d\omega \quad (54)$$

Where:

$x^*$  : New input point.

$y^*$  : New output point.

The process that we described above is mathematically known as Bayesian inference. Now, although it is true, we can appreciate that Bayesian probability allows us to model the uncertainty of a model; this process is associated with a high computational cost. This occurs because, up to this point, we can evaluate  $p(\omega | X, Y)$  analytically. To solve this problem, it is necessary to carry out the calculation of the integral using approximations.

Bearing in mind that the posterior probability of the Bayesian process over random variables  $\omega$  is rather complex to compute, (Gal & Ghahramani, 2016) propose to use the Variational Inference (VI) with an approximate distribution  $q_\theta(\omega)$  to approximate it. We can plot this distribution over  $\theta$  in order to evaluate the posterior probability of the Bayesian process. In this approach, the set of unknown numbers is labeled as  $\theta \in \mathbb{R}^n$ , namely  $\theta$  is a list of  $n$  unknown numbers, while the known numbers are listed as  $x \in \mathbb{R}^m$ . To carry out the above, first, (Gal & Ghahramani, 2016) use the Kullback-Leibler (KL) divergence (M. Jordan & et al., 1999) to measure the similarity between two distributions and thus best fits the result of the variational distribution to the result of the original model.

$$\text{KL}(q_\theta(\omega) || p(\omega | X, Y)) = \int q_\theta(\omega) \log \frac{q_\theta(\omega)}{p(\omega | X, Y)} d\omega \quad (55)$$

Then, by minimizing the KL divergence, we can approximate the predictive distribution as described in (Gal & Ghahramani, 2016) as follows:

$$p(y^* | x^*, X, Y) \approx \int p(y^* | x^*, \omega) q_\theta^*(\omega) d\omega \quad (56)$$

Where:

$q^*(\omega)$  : Minimum of the optimization objective (often a local minimum).

Finally, the minimization of the KL function is equivalent to the maximization of the evidence lower bound, which refers to the variational parameters defining  $q_\theta(\omega)$  in the

following manner:

$$\int q_{\theta}(\omega) \log p(Y | X, \omega) d\omega - \text{KL}(q_{\theta}(\omega) \| p(\omega)) \leq \log p(Y | X) \quad (57)$$

In the previous relationship, we can see two terms: in the first one, the expected log-likelihood appears, and in the second, the prior KL divergence. When the expected log likelihood is maximized, then we force  $q_{\theta}(\omega)$  to describe the data well, while minimising the prior KL and we obtain  $q_{\theta}(\omega)$  as close as possible to the prior. We can join the above terms to establish the cost function for the inference. We know this procedure as Variational Inference (VI) (Gal & Ghahramani, 2016). The Variational Inference is a machine learning method that approximates probability densities through the optimization process and, due to its versatility, is widely applied today.

$$\hat{L}_{\text{VI}}(\theta) = -\frac{N}{M} \sum_{i \in S} \int q_{\theta}(\omega) \log p(y_i | f^{\omega}(x_i)) d\omega + \text{KL}(q_{\theta}(\omega) \| p(\omega)) \quad (58)$$

Where:

S: Randomly sampled set of  $M$  indices from  $\{1, \dots, N\}$ .

Now, to compute the integral (namely, to estimate the expected log-likelihood), we apply the Monte Carlo (MC) estimator as follows (Gal & Ghahramani, 2016; J. Yao & et al., 2019):

$$\hat{L}_{\text{MC}}(\theta) = -\frac{N}{M} \sum_{i \in S} \log p(y_i | f^{\omega}(x_i)) d\omega + \text{KL}(q_{\theta}(\omega) \| p(\omega)) \quad (59)$$

Finally, to find the smallest value relation of the divergence between  $q_{\theta}(\omega)$  and  $P(\omega | X, Y)$ , we optimize  $\hat{L}_{\text{MC}}(\theta)$  as follow (Gal & Ghahramani, 2016):

$$\widehat{\Delta\theta} \leftarrow -\frac{N}{M} \sum_{i \in S} \frac{\partial}{\partial \theta} \log p(y_i | f^{\omega}(x_i)) + \frac{\partial}{\partial \theta} \text{KL}(q_{\theta}(\omega) \| p(\omega)) \quad (60)$$

#### 4.2.1 Uncertainty in Neural Networks and Dropout Technique

As mentioned in (Neal, 1996), the analysis of the stochasticity in a Bayesian multilayer perceptron allows to determine its uncertainty. It is necessary to remember that the stochastic theory allows the analysis of the uncertainty associated with certain parameters of a deterministic problem as long as certain information about these random variables is available. In this case, the injection of stochastic noise into the model through regularization techniques modifies the behavior of the network in such a way that we can utilize Bayesian modeling to determine its uncertainty. We can consider this an advantage today, as stochastic regularization techniques are widely used to improve the learning of neurons in a neural network. One of the best-known regularization techniques today is the dropout. As mentioned above, we utilize the dropout

in neural networks to randomly turn off neurons in a model to avoid overfitting in the learning process. To analyze the behavior of a network with dropout, we can consider that the stochastic noise of the drop can be transformed from the feature space to the parameter space as follows (Gal & Ghahramani, 2016):

$$\hat{y} = \sigma(x(\text{diag}(\hat{\epsilon}_1)M_1) + b)(\text{diag}(\hat{\epsilon}_2)M_2) \quad (61)$$

Where:

$\hat{\epsilon}_1, \hat{\epsilon}_2$ : Binary vectors to represent the dropout process (layer 1 and 2 (Gal & Ghahramani, 2016)).

$M_1, M_2$ : Weight matrix for the outputs of the layers 1 and 2.

$b$ : Bias vector.

$\sigma(\cdot)$ : Sigmoid function.

Considering that:  $\widehat{W}_1 := \text{diag}(\hat{\epsilon}_1)M_1$  and  $\widehat{W}_2 := \text{diag}(\hat{\epsilon}_2)M_2$  we can rewrite the above equation in the following way (Gal & Ghahramani, 2016):

$$\hat{y} = \sigma(x\widehat{W}_1 + b)\widehat{W}_2 =: f^{\widehat{W}_1, \widehat{W}_2, b}(x) \quad (62)$$

Bearing in mind the above, we can describe the cost function for dropout as follows (Gal & Ghahramani, 2016):

$$\widehat{L}_{\text{drop}}(M_1, M_2, b) := -\frac{1}{M} \sum_{i \in S} E^{\widehat{W}_1, \widehat{W}_2, b}(x_i, y_i) + \lambda_1 \|M_1\|^2 + \lambda_2 \|M_2\|^2 + \lambda_3 \|b\|^2 \quad (63)$$

With:

$$E^{W_1, W_2, b}(x, y) = \frac{1}{2N} \sum_{i=1}^N \|y_i - \hat{y}_i\|^2$$

Where:

$\lambda_i$ : Weight decay rates in the weight matrices and the bias vector in a Feed-forward Neural Network (with a single hidden layer).

$E^{W_1, W_2, b}(x, y)$ : Euclidean loss applied in a network for regression.

Additionally, we have to notice that:

$$E^{\widehat{W}_1, \widehat{W}_2, b}(x, y) = \frac{1}{2} \|y - f^{\widehat{W}_1, \widehat{W}_2, b}(x)\|^2 \quad (64)$$

Finally, to find the minimum error related to the Cost Function we optimize this relationship in the following way (Gal & Ghahramani, 2016):

$$\widehat{\Delta\theta} \leftarrow -\frac{1}{M\tau} \sum_{i \in S} \frac{\partial}{\partial \theta} \log p(y_i | f^\theta(x)) + \frac{\partial}{\partial \theta} (\lambda_1 \|M_1\|^2 + \lambda_2 \|M_2\|^2 + \lambda_3 \|b\|^2) \quad (65)$$



Where:

$\tau$  : Precision of the model.

#### 4.2.2 Estimators for Uncertainty

By inspecting Eqs. (60) and (65) we can say that they both present a fairly similar optimization procedure. The foregoing constitutes the basis of the comparison proposed in (Gal & Ghahramani, 2016) through which he concludes that a neural network trained with dropout behaves like a Bayesian network. Based on the above, the author proposes to use dropout regularization in a neural network during the analysis of the test set to obtain approximate samples of the posterior predictive distribution. However, it is first necessary to fit these samples by calculating unbiased estimators for the mean and variance  $\text{var}(y)$  of the posterior predictive distribution as described below (D. Dotlic & et al., 2019):

$$\hat{E}(y) = \frac{1}{T} \sum_{t=1}^T f_{\hat{\omega}_t}(x) \quad (66)$$

$$\hat{E}(y^T y) = \tau^{-1} \mathbf{I} + \frac{1}{T} \sum_{t=1}^T f_t(x)^T f_{\hat{\omega}_t}(x) - \hat{E}(y)^T \hat{E}(y) \quad (67)$$

Where:

$f_{\hat{\omega}_t}(x)$  : Output of the Bayesian Neural Network.

$t = 1, \dots, T$  : Samples from the posterior predictive distribution.

By inspecting the above equations, we can say that the mean of the posterior predictive samples can be assumed to be the unbiased estimator of the mean of the approximate distribution  $q_{\theta}(\omega)$ . While the sample variance plus the term  $\tau^{-1} \mathbf{I}$  can also be considered an unbiased estimator, in this case, the variance of  $q_{\theta}(\omega)$ . Considering the above, adjusting the precision of the model  $\tau$  will allow determining the most appropriate weight decay rate  $\lambda$  to reduce the uncertainty of the model (D. Dotlic & et al., 2019):

$$\lambda_i = \frac{(1 - P_i) l_i^2}{2N\tau} \quad (68)$$

Where:

$l_i^2$  : Prior length-scale.

$P_i$  : Dropout probability of the elements in vector  $\hat{\epsilon}_1, 0 \leq P_i \leq 1$  for  $i = 1, 2..$

#### 4.2.3 Experimentation

We present the test scenarios and experimentation published in (R. Llusi & et al., 2020b), where we take into account the data acquired from the AWS network installed in the city of Quito. Two cases are analyzed. The first (Case 1) seeks to evaluate the temperature prediction for the AWS under analysis based on historical temperature values at the same station. In the second scenario (Case 2), we seek to

obtain the temperature prediction based on the historical records of all the stations in the network. We utilize a CPU device with eight cores, 2.10 GHz core clock, and 16 GB of memory to compile the models. We observed a consumption of 45% of the CPU's memory during the simulation tasks. The ADAM optimization algorithm was working with the following parameters: Learning rate = 0.001,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 1e-08$ . For the training, validation and test data splitting, we utilized a distribution of 70%, 15%, 15% respectively (S. El Yacoubi & et al., 2018). The number of epochs used to fit the models is 70. Considering that, we propose the epoch partition to manage the entries of the model, so we adopted a batch size of 16. We implement the Rectified Linear Unit (ReLU) function activation in the models. Finally, we have adopted a down sampling stage to resample the data to 1-hour steps instead of 1-minute steps to implement the Walk Forward validation approach. As we mentioned earlier, the Bayesian neural network relies heavily on inference determination. With this in mind, selecting an appropriate dropout percentage will be crucial in determining the appropriate drop weight for the network to decrease forecast uncertainty. To apply the concept above, we utilize the equation (4.16) to enter the  $\lambda$  values in the dense layer after every dropout operation. For the experiments, we utilized a length scale of 0.01 and percentages of 0.18, 0.2, 0.25 and 0.15, 0.21 for the dropout regularization technique and the precision of the model, respectively. In order to induce priors on the weights (following a Gaussian distribution), we utilize the L2 regularization as established in (Rasp & Lerch, 2018) and the MSE (mean squared error) as the loss function. By experimentation, we determined a limit of 1500 iterations in the Monte Carlo method to estimate the output of the neural network. Finally, we analyzed the temperature forecast for AWS 1 for the different trials and the outcomes of the experiments, considering the two possible scenarios, Case 1 and Case 2.

#### 4.2.4 Results from the experiments

We implemented a number of units per layer of L1: 200 and L2: 100 for the LSTM neural network. We select this number of units per layer aiming to reduce as much as possible the processing complexity. We initially utilized a value of  $\lambda = 8 \times 10^{-5}$  to improve learning, as proposed in (Krogh & Herts, 1991) for a feed-forward neural network. Nevertheless, we modified experimentally this value to  $\times 10^{-7}$ , because the developed models were different from the Feed Forward Network and the outcomes were better. In order to show the reduction of uncertainty in the forecast of the network, we present 10 trials at different times. In Table 5 and Table 6, we present a comparison between the maximum values of the error metrics for all the iterations to establish a base of comparison for the prediction's uncertainty.

Then, to present a more in-depth perspective of the work carried out in this section, Figures 12 and 13 show the MAPE variation analysis for the LSTM network.

Table 5: Parameters and forecast information for LSTM (without Bayesian Modelling)

Scenario	$P_i$	RMSE (°C)	MSE (°C)	MAE (°C)	MAPE (%)
Case 1	0.18	1.3	5.9	1.4	6.8
Case 2	0.25	2.3	12	2.4	10.6

Table 6: Exemplary Table

Scenario	$P_i$	$\tau$	$\lambda (10^{-7})$	RMSE (°C)	MSE (°C)	MAE (°C)	MAPE (%)
Case 1	0.18	0.15	2.23	1.0	3.4	0.9	4.3
Case 2	0.25	0.15	2.04	1.3	5.9	1.5	6.4

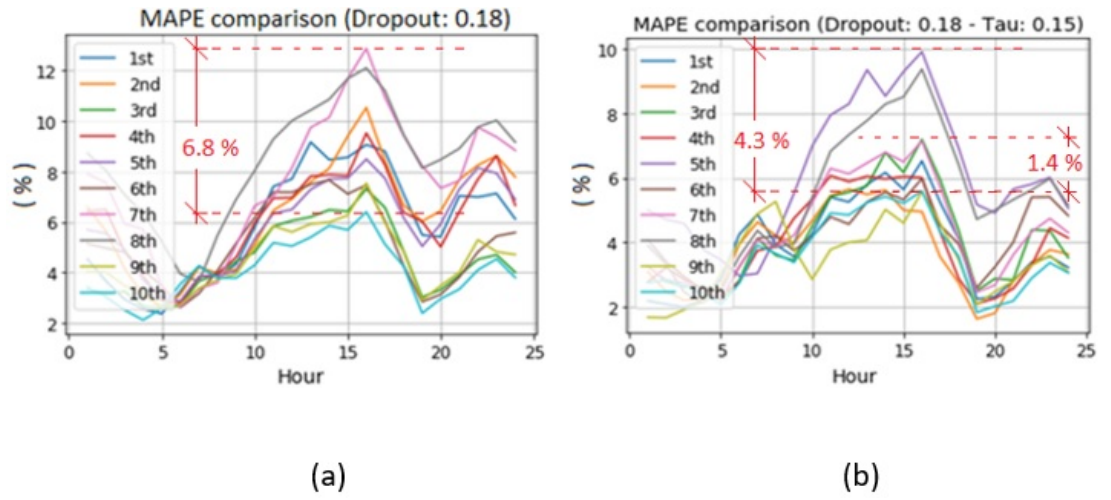


Figure 12: MAPE comparison for the Neural Network with LSTM structure (Scenario 1). (a) Without Bayesian Modelling, (b) With Bayesian Modelling.

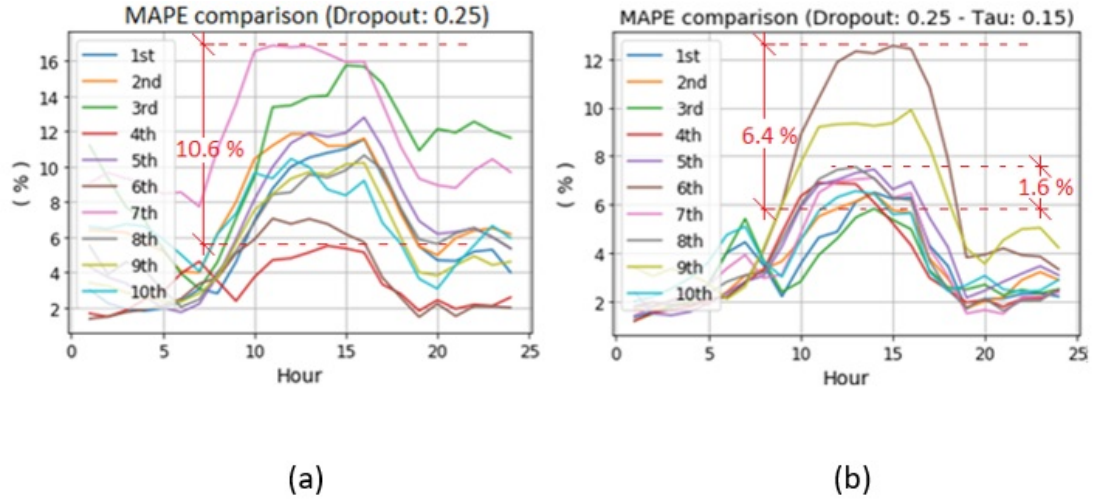


Figure 13: MAPE comparison for the Neural Network with LSTM structure (Scenario 2). (a) Without Bayesian Modelling, (b) With Bayesian Modelling.

#### 4.2.5 Discussion

We reduce the variation of the error (for both scenarios under analysis) through Bayesian modeling applied to the LSTM Neural Network; see Table 5 and 6. Thanks to the above approach, we reduce the error in the forecast for LSTM. In Case 1, for each error metric, it can be said that the error is reduced from 1.3 °C to 1.0 °C for the RMSE, from 5.9 °C to 3.4 °C for the MSE, from 1.4 °C to 0.9 °C for the MAE and from 6.8% to 4.3% for MAPE. We obtain the above when  $P_i=0.18$ ,  $\tau=0.15$ ,  $\lambda = 2.23 \times 10^{-7}$ . In Case 2, the error is reduced from 2.3 °C to 2.04 °C for the RMSE, from 12 °C to 5.9 °C for the MSE, from 2.4 °C to 1.5 °C for the MAE and from 10.6% to 6.4% for the MAPE. We obtain the above when  $P_i=0.25$ ,  $\tau=0.15$ ,  $\lambda = 2.04 \times 10^{-7}$ . We satisfactorily observe a particular reduction in the MSE value. From the MAPE graphic analysis of Figure 12 and Figure 13 we obtain additional information using the walk-forward validation with steps of 24 hours. We selected the MAPE error because the percentage analysis can show clearly the variation of the neural network's output, namely the uncertainty in the temperature forecast of an AWS. In these figures, we can say that 80% of time, the prediction values are closer to the real value expected for the prediction. The boundaries for these curves can be determined in order to appreciate the decrease in the error prediction. Bearing in mind the above, a sharp decrease in percentages of the MAPE can be observed (from 1.4%, and 1.6% respectively).

### 4.3 Adaptive Moment Estimation to reduce uncertainty

In this section, one of the fundamental paradigms of the research carried out is presented: the implementation of an optimizer for the learning task of the neural network. We carry out an explanation of non-convex optimization as an introductory mechanism to introduce the weight decay approach to perform the optimization pro-

cess. We present the weight decay methodology assumed to carry out the investigation and the respective mathematical demonstration of the convergence of the optimizer. Finally, we present the experimentation carried out to verify the operation of the optimizer.

#### 4.3.1 The optimization process

The learning process in a neural network consists of finding the model parameters that minimize its cost function, considering the different types of training data input to the network. The first step in performing an optimization analysis is to determine whether the problem at hand is convex or non-convex. If an optimization problem is convex, the presence of a local minimum implies the existence of a global minimum. If a convex function in a collection of global minima has a minimum, then that minimum will be unique. Therefore, if the optimization problem is convex, all local optima are global, and we only need to locate one. While a non-convex optimization may have multiple locally optimal points and its location process can take a long time to determine, it is recommended to look for stationary points, which are located where  $\nabla_{\theta} f(\theta_{\text{stationary}}) = 0$ . This is because trying to find global and local minima in a non-convex optimization is NP-hard<sup>3</sup> (M. Danilova & et al., 2020). Determining the  $\varepsilon$  - First Order Stationary Points (FOSP) or locating the  $\varepsilon$  - Second Order Stationary Points (SOSP) are thus two possible approaches for locating the stationary points. In the first case, the points to be determined are saddle and plateau points, namely where  $\|\nabla_{\theta} f(\theta_{\text{FOSP}})\|_2 \leq \varepsilon$  and  $\lambda_{\min}(\nabla_{\theta, \theta}^2 f(\theta_{\text{FOSP}})) \geq -\sqrt{\varepsilon}$  (Jain & Kar, 2017). While in the second case, the global, local minima and plateau points are determined considering  $\|\nabla_{\theta} f(\theta_{\text{FOSP}})\|_2 \leq \varepsilon$  and  $\lambda_{\min}(\nabla_{\theta, \theta}^2 f(\theta_{\text{FOSP}})) \geq -\sqrt{\varepsilon}$  (Jain & Kar, 2017). Nowadays, we can use several optimization algorithms to carry out the learning process of the neural network, but in a general way, we can divide them into three groups: Zero-Order Methods (ZO) methods, first-order methods, and second-order methods. The Zero-Order optimization methods ignore the computing of the gradient, bringing it closer through a function value-based gradient estimates (P. C. S. Liu & et al., 2020) (e.g. ZO sign-based SGD (ZO-sign SGD), ZO hessian-based (ZO-Hess), ZO stochastic conditional gradient (ZO-SCG) algorithm and (ZO adaptive momentum method (ZO-ADAMM)). In contrast, the non-convex ZO algorithms use a stationary condition to measure the convergence of the methods (B. K. S. Liu & et al., 2018) (e.g. ZO gradient descent (ZO-GD) and (Stochastic ZO-GD (ZO-SGD)). The first-order method for non-convex optimization is related to the search for an  $\varepsilon$ -approximate first-order stationary point FOSP (given a precision accuracy of  $\varepsilon > 0$ ) such that  $\|\nabla f(\theta)\| \leq \varepsilon$  with  $\theta \in \mathbb{R}^d$  (M. Danilova & et al., 2020) (e.g. Gradient De-

<sup>3</sup>The computational complexity class NP-hard is the set of decision problems containing problems H such that every problem L in NP can be polynomially transformed into H.

scent (GD), stochastic gradient descent (SGD), adaptive moment estimation (ADAM), adaptive gradient algorithm (Adagrad) and root mean square propagation (RMSProp)). For the analysis of the second-order method, it is assumed that  $f(\theta) : \mathbb{R}^d \rightarrow \mathbb{R}$  has a Lipschitz continuous gradient and Hessian, so that the  $\varepsilon$  - second-order stationary point (SOSP), that is,  $\theta \in \mathbb{R}^d$ , exist if  $\|\nabla f(\theta)\|_2 \leq \varepsilon, \lambda_{\min}(\nabla^2 f(\theta)) \geq -\delta$  (Y. Arjevani & et al., 2020), (e.g. Newton Method, Regularized Method and Stochastic Quasi-Newton). In order to present the weight decay approach designed for our research, we introduce the convergence analysis used in (M. Zaheer & et al., 2018) for the ADAM optimizer. To start the optimization analysis, we have to initially presume the following regression loss function:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n (y_i - f_{\theta}(x_i))^2 \quad (69)$$

Where:

$f_{\theta}(x_i)$  : Actual value.

$y_i$  : Predicted value.

It is important to note that  $\xi_t$  is sampled with  $\xi_t = i_t \in \{1, \dots, n\}$ . Using  $\theta$  sampled with  $i_t$ , the cost function can therefore be modified as follows:

$$L(\theta, i_t) = (y_{i_t} - f_{\theta}(x_{i_t}))^2 \quad (70)$$

Now, when calculating the gradient of the cost function, we can write:

$$\nabla_{\theta} L(\theta, i_t) = -2(y_{i_t} - f_{\theta}(x_{i_t})) \nabla f_{\theta}(x_{i_t}) \quad (71)$$

Given that the optimizer's goal is to minimize the objective function  $E_{\xi \sim P}[L(\theta; \xi)]$  over  $\theta$  (M. Zaheer & et al., 2018), in order to obtain  $\min_{\theta} E_{\xi \sim P}[L(\theta; \xi)]$ , we initially assumed that the loss function is L-smooth (Assumption I), namely, that there exists a constant L such that:

$$\|\nabla L(\theta_1; \xi) - \nabla L(\theta_2; \xi)\|_2 \leq L \|\theta_2 - \theta_1\|_2 \forall \theta_1, \theta_2 \text{ and } \xi \quad (72)$$

The assumption I implies that if  $\theta_1, \theta_2$  change slightly for any  $\xi$ , the change in gradient is small because the value of the gradient difference does not change significantly. The function can converge if  $f(\theta_{t+1}) \leq f(\theta_t) - \Delta$ , assuming the function is lower-bounded (i.e., it does not go to  $-\infty$ ). The preceding reflects the change in function value between two algorithm iterations. Now, if the objective function is L-smooth, the cost function can be rewritten as follows:

$$L(\theta_{t+1}) \leq L(\theta_t) + \nabla^T L(\theta_t)(\theta_{t+1} - \theta_t) + \frac{L}{2} \|\theta_{t+1} - \theta_t\|_2^2 \quad (73)$$

### 4.3.2 Weight Decay

The ADAM variation based on weight decay decoupling (ADAMW) described in (Loshchilov & Hutter, 2019) will implement the exponential decay using as its base the following update rule:

$$\theta_{t+1} = (1 - \lambda)\theta_t - \alpha \nabla f_t(\theta_t) \quad (74)$$

The strategy assumed in (Loshchilov & Hutter, 2019) implies that the weight decay adopt the advantages and differences between weight decay and L2 regularization for adaptive gradients by introducing a weight decay factor  $\lambda \theta_t$  as follows:

$$\theta_{t+1} = \theta_t - \alpha \nabla f_t(\theta_t) - \lambda \theta_t \quad (75)$$

To relate the weight decay with the learning process of the neural network, ADAMW uses the learning rate  $\eta_t$  to modify the updating rule through the term  $\eta_t \lambda \theta_t$ . This strategy appears to be interesting because it introduces a Weight Decay factor easily in the update rule of the ADAM optimizer. Nevertheless, we have to note that, as the authors of the ADAMW approach have only tested the algorithm empirically (Loshchilov & Hutter, 2019). Nowadays the authors or even other authors have not provided a mathematical prove of the convergence of this approach.

To overcome this problem in our research work, we have modified the ADAMW approach by combining it with the L-Smooth property assumed for the objective function in ADAM (Zaheer et al., 2018). We carry out this strategy with the purpose of providing a more consistent alternative to weight loss decoupling. So, being aware that the loss function is L-smooth, we propose to implement a weight decay approach by rewriting the equation Eq. (75) in the following manner:

$$\theta_{t+1} = \theta_t - \alpha \nabla f_t(\theta_t) - \lambda \quad (76)$$

To distinguish ourselves from the work of (Loshchilov & Hutter, 2019), we propose this method, which takes into account only the rate of weight decay,  $\lambda$ . To update the learning process, it is necessary to aggregate the rate of weight decay,  $\lambda$ , as suggested by the analogy between ADAM's update rule and weight decay decoupling's rule.

### 4.3.3 ADAML

We refer to our approach as ADAM Logger (ADAML) bearing in mind the way in which the algorithm registers the changes in the location of the minimal error points of the cost function using the rate of weight decay. Below are the specifics of ADAM's original algorithm and the proposed ADAML variation; see Table 7.

Table 7: Specifics of the Optimizer Algorithms ADAM and ADAML.

Initial parameters value: $\theta_1 \in \mathbb{R}^d$	For $t = 1$ to $T$ do:
	Draw a sample $\xi_t$ from $P$
Learning rates: $\{\eta_t\}_{t=1}^T$	Compute gradients at moment $t$ :
	$g_t = \nabla L(\theta_t, \xi_t)$
Decay parameters: $0 \leq \beta_1, \beta_2 \leq 1$	Update biased first moment estimate:
	$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
Stability parameter: $\delta > 0$	Update biased second raw moment estimate:
	$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
Update procedure:	Compute bias-corrected first moment estimate:
	$\hat{m}_t = \frac{m_t}{(1 - \beta_1^t)}$
Set $m_0 = 0$ (Initialize 1 <sup>st</sup> moment vector)	Compute bias-corrected second raw moment estimate:
	$\hat{v}_t = \frac{v_t}{(1 - \beta_2^t)}$
and $v_0 = 0$ (Initialize 2 <sup>nd</sup> moment vector)	Update Rule (Update parameters)
	$\theta_{t+1} = \theta_t - \eta_t \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \delta}} - \lambda$
	End for

#### 4.3.4 Proof of ADAML's convergence

To study the convergence of the proposed ADAM variation, we need to remember first that the objective function is  $L$ -smooth. Then, we can inspect two successive iterations of the cost function as follows:

$$L(\theta_{t+1}) \leq L(\theta_t) + \nabla^T L(\theta_t)(\theta_{t+1} - \theta_t) + \frac{L}{2} \|\theta_{t+1} - \theta_t\|_2^2 \quad (77)$$

For the sake of simplicity, we will assume that  $\beta_1 = 0$ , so  $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$  becomes  $m_t = g_t$  (M. Zaheer & et al., 2018).

$$\theta_{t+1} = \theta_t - \eta_t \frac{g_t}{\sqrt{v_t} + \delta} - \lambda \quad (78)$$

Now, considering the successive iterations of the cost function, the update process component-wise becomes:

$$\theta_{i,t+1} - \theta_{i,t} = -\eta_t \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \lambda i \in \{1, \dots, d\} \quad (79)$$

Consequently, we can rewrite the cost function as follows:



$$L(\theta_{t+1}) \leq L(\theta_t) - \sum_{i=1}^d \left( [\nabla L(\theta_t)]_i \left( \frac{\eta_t g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \lambda \right) \right) + \frac{L}{2} \sum_{i=1}^d \left( \frac{\eta_t g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \lambda \right)^2 \quad (80)$$

What develops in:

$$\begin{aligned} L(\theta_{t+1}) \leq L(\theta_t) - \eta_t \sum_{i=1}^d \left( [\nabla L(\theta_t)]_i \left( \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} \right) \right) + \frac{L\eta_t^2}{2} \sum_{i=1}^d \left( \frac{g_{i,t}^2}{(\sqrt{v_{i,t}} + \delta)^2} \right) \\ - L\eta_t \lambda \sum_{i=1}^d \left( \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} \right) + \lambda \sum_{i=1}^d [\nabla L(\theta_t)]_i + \frac{\lambda^2}{2} \end{aligned} \quad (81)$$

Now, the conditional expectation with respect to the sample at iteration  $t$  given a fixed random variable  $\theta_t$ , is obtained as follows:

$$\begin{aligned} \mathbb{E}[L(\theta_{t+1}) \mid \theta_t] \leq L(\theta_t) - \eta_t \sum_{i=1}^d \left( [\nabla L(\theta_t)]_i \times \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} \mid \theta_t \right] \right) \\ + \frac{L\eta_t^2}{2} \sum_{i=1}^d \mathbb{E} \left[ \frac{\eta_t^2 g_{i,t}^2}{(\sqrt{v_{i,t}} + \delta)^2} \mid \theta_t \right] - L\eta_t \lambda \sum_{i=1}^d \left( \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} \mid \theta_t \right] \right) \\ + \lambda \sum_{i=1}^d (\mathbb{E}[[\nabla L(\theta_t)]_i \mid \theta_t]) + \frac{\lambda^2}{2} \end{aligned} \quad (82)$$

To continue with the analysis we need to bound the terms:  $\mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} \mid \theta_t \right]$ ,  $\mathbb{E} \left[ \frac{\eta_t^2 g_{i,t}^2}{(\sqrt{v_{i,t}} + \delta)^2} \mid \theta_t \right]$  and  $\mathbb{E}[[\nabla L(\theta_t)]_i \mid \theta_t]$  in the above relationship. The term  $\mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} \mid \theta_t \right]$  is going to be bound based on the gradient norm. On the other hand, the term  $\mathbb{E} \left[ \frac{\eta_t^2 g_{i,t}^2}{(\sqrt{v_{i,t}} + \delta)^2} \mid \theta_t \right]$  is going to be bound based on batch size. Finally, the term  $\mathbb{E}[[\nabla L(\theta_t)]_i \mid \theta_t]$  is going to be bound, assuming that the loss function has a bound gradient. This constitutes the 2<sup>nd</sup> assumption made in (M. Zaheer & et al., 2018) to prove the convergence of the ADAM optimizer.

In order to bound the terms above we start with the term close to  $-\eta_t \sum_{i=1}^d$ :

$$\dots - \eta_t \sum_{i=1}^d \left( [\nabla L(\theta_t)]_i \times \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} + \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \right) + \dots \quad (83)$$

Considering the property of expectation  $\mathbb{E}[a - b + c] = \mathbb{E}[a - b] + \mathbb{E}[c]$  the cen-

tral term of the above relationship can be written as follows:

$$\dots \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} + \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \dots \quad (84)$$

$$= \dots \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] + \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \dots \quad (85)$$

The terms of the expected value of the loss functions related to  $\mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} \mid \theta_t \right]$ , can be rewritten as follows:

$$\mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] = \frac{\mathbb{E} g_{i,t} \mid \theta_t}{\sqrt{\beta_2 v_{i,t-1}} + \delta} = \frac{|\nabla L(\theta_t)|_i}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \quad (86)$$

So the original relationship can be rewritten as follows:

$$\dots - \eta_t \sum_{i=1}^d \left( |\nabla L(\theta_t)|_i \times \left[ \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] + \frac{|\nabla L(\theta_t)|_i}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \right] \right) + \dots \quad (87)$$

Rearranging the above relationship, the following can be written:

$$\dots - \eta_t \sum_{i=1}^d \left( \frac{[\nabla L(\theta_t)]_i^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} + |\nabla L(\theta_t)|_i \times \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \right) + \dots \quad (88)$$

$$\dots - \eta_t \sum_{i=1}^d \frac{[\nabla L(\theta_t)]_i^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} - \eta_t \sum_{i=1}^d \left( |\nabla L(\theta_t)|_i \times \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \right) + \dots \quad (89)$$

Now, bearing in mind that:

$$-\eta_t \sum_{i=1}^d a_i b_i \leq \left| \eta_t \sum_{i=1}^d a_i b_i \right| \leq \eta_t \sum_{i=1}^d |a_i| |b_i| \quad (90)$$

So the second term can be written in the following way:

$$\begin{aligned} & \eta_t \sum_{i=1}^d \left( |\nabla L(\theta_t)|_i \times \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \right) \\ & \leq \eta_t \sum_{i=1}^d ||\nabla L(\theta_t)|_i| \left| \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \right| \end{aligned} \quad (91)$$

Then the original relationship can be written in the following way:

$$\begin{aligned} \mathbb{E}[L(\theta_{t+1}) | \theta_t] &\leq L(\theta_t) - \eta_t \sum_{i=1}^d \frac{[\nabla L(\theta_t)]_i^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \\ &+ \eta_t \sum_{i=1}^d |[\nabla L(\theta_t)]_i| \left| \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \right| + \dots \end{aligned} \quad (92)$$

Now working on the expected value:

$$\left| \mathbb{E} \left[ \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \right| \quad (93)$$

The internal part of the expected value can be rewritten bearing in mind that  $|\mathbb{E}[x]| \leq \mathbb{E}[|x|]$  as follows:

$$\left| \frac{g_{i,t}}{\sqrt{v_{i,t}} + \delta} - \frac{g_{i,t}}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \right| = |g_{i,t}| \left| \frac{1}{\sqrt{v_{i,t}} + \delta} - \frac{1}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \right| \quad (94)$$

$$= \frac{|g_{i,t}|}{(\sqrt{v_{i,t}} + \delta)(\sqrt{\beta_2 v_{i,t-1}} + \delta)} \left| \sqrt{\beta_2 v_{i,t-1}} - \sqrt{v_{i,t}} \right| \quad (95)$$

Now taking into account the equivalence:

$$|\sqrt{a} - \sqrt{b}| = \frac{|a - b|}{\sqrt{a} + \sqrt{b}} \quad (96)$$

The above can be written as follows:

$$,, = \frac{|g_{i,t}|}{(\sqrt{v_{i,t}} + \delta)(\sqrt{\beta_2 v_{i,t-1}} + \delta)} \frac{|\beta_2 v_{i,t-1} - v_{i,t}|}{\sqrt{v_{i,t}} + \sqrt{\beta_2 v_{i,t-1}}} \quad (97)$$

Then taking into account the update rule  $v_{i,t} = \beta_2 v_{i,t-1} + (1 - \beta_2) g_{i,t}^2$ , the above can be written as follows:

$$,, = \frac{|g_{i,t}|}{(\sqrt{v_{i,t}} + \delta)(\sqrt{\beta_2 v_{i,t-1}} + \delta)} \frac{(1 - \beta_2) g_{i,t}^2}{\sqrt{\beta_2 v_{i,t-1} + (1 - \beta_2) g_{i,t}^2} + \sqrt{\beta_2 v_{i,t-1}}} \quad (98)$$

Now considering the relationship  $\frac{1}{a+b} \leq \frac{1}{a}$  for  $a > 0$  and  $b \geq 0$ , the above can be rewritten as follows:

$$,, = \frac{|g_{i,t}|}{(\sqrt{v_{i,t}} + \delta)(\sqrt{\beta_2 v_{i,t-1}} + \delta)} \frac{(1 - \beta_2) g_{i,t}^2}{\sqrt{\beta_2 v_{i,t-1} + (1 - \beta_2) g_{i,t}^2}} \quad (99)$$

Additionally, considering that  $\sqrt{a+b} \geq \sqrt{b}$  if  $a \geq 0$  and  $b > 0$ , then  $\frac{1}{\sqrt{a+b}} \leq$

$\frac{1}{\sqrt{b}}$ , so the above can be rewritten as follows:

$$,, = \frac{|g_{i,t}|}{(\sqrt{v_{i,t}} + \delta) (\sqrt{\beta_2 v_{i,t-1}} + \delta)} \frac{(1 - \beta_2) g_{i,t}^2}{\sqrt{(1 - \beta_2) g_{i,t}^2}} \quad (100)$$

$$,, = \frac{\sqrt{(1 - \beta_2) g_{i,t}^2}}{(\sqrt{v_{i,t}} + \delta) (\sqrt{\beta_2 v_{i,t-1}} + \delta)} \quad (101)$$

Now using again the relationship  $\frac{1}{a+b} \leq \frac{1}{a}$  for  $a > 0$  and  $b \geq 0$ , the above can be rewritten as follows:

$$,, = \frac{\sqrt{(1 - \beta_2) g_{i,t}^2}}{\delta (\sqrt{\beta_2 v_{i,t-1}} + \delta)} \quad (102)$$

Then the original expected value can be written as:

$$\begin{aligned} E[L(\theta_{t+1}) | \theta_t] &\leq L(\theta_t) - \eta_t \sum_{i=1}^d \frac{[\nabla L(\theta_t)]_i^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \\ &+ \eta_t \sum_{i=1}^d \left| [\nabla L(\theta_t)]_i \frac{\sqrt{(1 - \beta_2)}}{\delta} \right| E \left[ \frac{g_{i,t}^2}{(\sqrt{\beta_2 v_{i,t-1}} + \delta)} | \theta_t \right] + \dots \end{aligned} \quad (103)$$

Now, we are going to use the 2<sup>nd</sup> assumption "Loss function", to handle the absolute value of the gradient of the above equation and the term  $\lambda \sum_{i=1}^d (E[|\nabla L(\theta_t)|_i | \theta_t]) + \frac{\lambda^2}{2}$ , as follows:

Assumption II:  $\|\nabla L(\theta; \xi)\| \leq G, \forall \theta \in \mathbb{R}^d, \forall \xi$  (The function  $L$  has a bound gradient).

It can be said that:

$$\|\nabla L(\theta)\| = \|\mathbb{E}_\xi[\nabla L(\theta; \xi)]\| \quad (104)$$

$$\|\mathbb{E}_\xi[\nabla L(\theta; \xi)]\| \leq \mathbb{E}_\xi[\|\nabla L(\theta; \xi)\|] \quad (105)$$

If  $\|\nabla L(\theta; \xi)\| \leq G$ , then:

$$\mathbb{E}_\xi[\|\nabla L(\theta; \xi)\|] \leq G \quad (106)$$

And finally:

$$|[\nabla L(\theta_t)]_i| \leq G \quad (107)$$

So the overall bound can be written as:

$$E[L(\theta_{t+1}) | \theta_t] \leq L(\theta_t) - \eta_t \sum_{i=1}^d \frac{[\nabla L(\theta_t)]_i^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta}$$

$$\begin{aligned}
 & + \frac{\eta_t G \sqrt{1 - \beta_2}}{\delta} \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \\
 & + \frac{L \eta_t^2}{2} \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{(\sqrt{v_{i,t}} + \delta)^2} \mid \theta_t \right] - L \eta_t \lambda \sum_{i=1}^d \left( \frac{|\nabla L(\theta_t)|_i}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \right) + \lambda G \\
 & + \frac{\lambda^2}{2}
 \end{aligned} \tag{108}$$

In a similar way, we can use the update rule:  $v_{i,t} = \beta_2 v_{i,t-1} + (1 - \beta_2) g_{i,t}^2$  to bound the term  $\frac{L \eta_t^2}{2} \sum_{i=1}^d \mathbb{E} \left[ \frac{\eta_t^2 g_{i,t}^2}{(\sqrt{v_{i,t}} + \delta)^2} \mid \theta_t \right]$  and write the following:

$$\begin{aligned}
 & \frac{L \eta_t^2}{2} \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{(\sqrt{v_{i,t}} + \delta)^2} \mid \theta_t \right] \\
 & = \frac{L \eta_t^2}{2} \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{\left( \sqrt{\beta_2 v_{i,t-1}} + (1 - \beta_2) g_{i,t}^2 + \delta \right)^2} \theta_t \right]
 \end{aligned} \tag{109}$$

Now, considering that  $(1 - \beta_2) g_{i,t}^2$  is non-negative and  $\frac{1}{\sqrt{a+b}} \leq \frac{1}{\sqrt{a}}$ , then:

$$\begin{aligned}
 & \frac{L \eta_t^2}{2} \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{\left( \sqrt{\beta_2 v_{i,t-1}} + (1 - \beta_2) g_{i,t}^2 + \delta \right)^2} \theta_t \right] \\
 & = \frac{L \eta_t^2}{2} \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{(\sqrt{\beta_2 v_{i,t-1}} + \delta)^2} \theta_t \right]
 \end{aligned} \tag{110}$$

Then the following equivalence can be used:

$$\frac{L \eta_t^2}{2} \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{(\sqrt{\beta_2 v_{i,t-1}} + \delta)^2} \mid \theta_t \right] \leq \frac{L \eta_t^2}{2 \delta} \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \tag{111}$$

So the expected value is:

$$\begin{aligned}
 E[L(\theta_{t+1}) \mid \theta_t] & \leq L(\theta_t) - \eta_t \sum_{i=1}^d \frac{[\nabla L(\theta_t)]_i^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \\
 & + \frac{\eta_t G \sqrt{1 - \beta_2}}{\delta} \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \\
 & + \frac{L \eta_t^2}{2 \delta} \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] \dots
 \end{aligned} \tag{112}$$

Now, we can rewrite the above as follows:

$$\begin{aligned} \dots &\leq L(\theta_t) - \eta_t \sum_{i=1}^d \frac{[\nabla L(\theta_t)]_i^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \\ &+ \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right] - \dots \end{aligned} \quad (113)$$

We can reduce the term  $\left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \sum_{i=1}^d \mathbb{E} \left[ \frac{g_{i,t}^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \mid \theta_t \right]$  with the relationship  $\frac{1}{a+b} \leq \frac{1}{a}$ :

$$\begin{aligned} &\left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \sum_{i=1}^d \frac{1}{\delta} \mathbb{E} [g_{i,t}^2 \mid \theta_t] \\ &= \frac{1}{\delta} \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \sum_{i=1}^d \mathbb{E} [g_{i,t}^2 \mid \theta_t] \end{aligned} \quad (114)$$

Bearing in mind that  $g_{i,t}^2$  is equal to  $\|g_t\|^2$  and using the above, we can rewrite the expected value of the loss function, as follows:

$$\begin{aligned} \mathbb{E} [L(\theta_{t+1}) \mid \theta_t] &\leq L(\theta_t) - \eta_t \sum_{i=1}^d \frac{[\nabla L(\theta_t)]_i^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \\ &+ \frac{1}{\delta} \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \mathbb{E} [\|g_t\|^2 \mid \theta_t] - \dots \end{aligned} \quad (115)$$

Now in order to bound the denominator of the 2<sup>nd</sup> term the following assumption can be adopted:

$$v_{i,t} \leq G^2, \forall i, t \quad (116)$$

So it can be written that:

$$\sqrt{\beta_2 v_{i,t-1}} + \delta \leq \sqrt{\beta_2} G + \delta \quad (117)$$

Which consequently gives us

$$-\eta_t \sum_{i=1}^d \frac{[\nabla L(\theta_t)]_i^2}{\sqrt{\beta_2 v_{i,t-1}} + \delta} \leq -\frac{\eta_t}{\sqrt{\beta_2} G + \delta} \sum_{i=1}^d [\nabla L(\theta_t)]_i^2 \quad (118)$$

And Finally, it can be written as:

$$-\frac{\eta_t}{\sqrt{\beta_2} G + \delta} \sum_{i=1}^d [\nabla L(\theta_t)]_i^2 = -\frac{\eta_t}{\sqrt{\beta_2} G + \delta} \|\nabla L(\theta_t)\|_2^2 \quad (119)$$

So the expected value is:

$$\begin{aligned} \mathbb{E}[L\theta_{t+1} \mid \theta_t] &\leq L(\theta_t) - \frac{\eta_t}{\sqrt{\beta_2}G + \delta} \|\nabla L(\theta_t)\|_2^2 \\ &+ \frac{1}{\delta} \left( \frac{\eta_t G \sqrt{1 - \beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \mathbb{E}[\|g_t\|^2 \mid \theta_t] - \dots \end{aligned} \quad (120)$$

In order to proceed it is necessary to keep in mind the fundamental notion of the expected value, we are able to recast the bound in the form  $\mathbb{E}[L\theta_{t+1} \mid \theta_t] \leq L(\theta_t) - \Delta$ . To handle all of the constants involved in the connection, we may make use of assumption III (M. Zaheer & et al., 2018), which will allow us to manage the information shown above.

Assumption III:  $E_\xi [\|\nabla L(\theta; \xi) - \nabla L(\theta)\|_2^2] \leq \sigma^2, \forall \theta \in \mathbb{R}^d, \forall \xi$  (Bound on the variance in stochastic gradients).

To manage the above, we must first consider the use of mini batches ( $b_t$ ):

$$g_t(\cdot) = \frac{1}{b_t} \sum_{\xi \in B_t} L(\cdot; \xi) \quad (121)$$

With the above, we prove that:

$$\mathbb{E}[\|g_t\|_2^2 \mid \theta_t] \leq \frac{1}{b_t} \left( \sigma^2 + \|\nabla L(\theta_t)\|_2^2 \right) \quad (122)$$

So the expected value is:

$$\begin{aligned} \mathbb{E}[L\theta_{t+1} \mid \theta_t] &\leq L(\theta_t) - \|\nabla L(\theta_t)\|_2^2 \left( \frac{\eta_t}{\sqrt{\beta_2}G + \delta} - \frac{1}{\delta b_t} \left( \frac{\eta_t G \sqrt{1 - \beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \right) \\ &+ \frac{\sigma^2}{\delta b_t} \left( \frac{\eta_t G \sqrt{1 - \beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) - \frac{LG\eta_t\lambda}{\delta} + \lambda G + \frac{\lambda^2}{2} \end{aligned} \quad (123)$$

Now, using the relationship  $\frac{1}{a+b} \leq \frac{1}{a}$  the 4<sup>th</sup> term above can be modified and the following can be done:

$$\dots - \frac{L\eta_t\lambda}{\delta} \sum_{i=1}^d |\nabla L(\theta_t)|_i + \dots \quad (124)$$

Then, using the Assumption II, the following can be written:

$$\mathbb{E}[L\theta_{t+1} \mid \theta_t] \leq L(\theta_t) - \|\nabla L(\theta_t)\|_2^2 \left( \frac{\eta_t}{\sqrt{\beta_2}G + \delta} - \frac{1}{\delta b_t} \left( \frac{\eta_t G \sqrt{1 - \beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \right)$$

$$+ \frac{\sigma^2}{\delta b_t} \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) - \frac{LG\eta_t\lambda}{\delta} + \lambda G + \frac{\lambda^2}{2} \quad (125)$$

We can define the expected value for the bound as  $E[L\theta_{t+1} \mid \theta_t] \leq L(\theta_t) - \Delta + c$ . However, to drive the constant component to zero and reach convergence, we should obligatorily tune the different parameters, such as batch sizes, learning rates, and extra-free parameters. To accomplish the above, we can propose the following:

$$E[L(\theta_{t+1}) \mid \theta_t] \leq L(\theta_t) - \|\nabla L(\theta_t)\|_2^2 \left( \frac{\eta_t}{\sqrt{\beta_2}G + \delta} - \frac{1}{\delta b_t} \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \right) \\ + \frac{\sigma^2}{\delta b_t} \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) + \lambda G \left( 1 - \frac{\eta_t L}{\delta} \right) + \frac{\lambda^2}{2} \quad (126)$$

To achieve the aforementioned, we can choose:

- $b_t$  (batch size)  $\geq 1$
- $\eta_t$  (learning rate)  $= \eta$ , such that  $\frac{L\eta}{2\delta} \leq \frac{G\sqrt{1-\beta_2}}{\delta}$
- $\beta_2$  such that:  $\frac{2G\sqrt{1-\beta_2}}{\delta^2} \leq \frac{1}{2} \left( \frac{1}{\sqrt{\beta_2}G + \delta} \right)$
- $\eta_t = \eta$ , such that  $\frac{L\eta}{2\delta} \leq \frac{G\sqrt{1-\beta_2}}{\delta}$

Choosing  $b_t$  (batch size)  $\geq 1$ , then the 2<sup>nd</sup> term of the above can be rewritten as follows:

$$\frac{1}{\delta b_t} \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \leq \frac{1}{\delta} \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \quad (127)$$

So,

$$- \frac{1}{\delta b_t} \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \geq - \frac{1}{\delta} \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \quad (128)$$

And finally,

$$\frac{\eta_t}{\sqrt{\beta_2}G + \delta} - \frac{1}{\delta} \left( \frac{\eta_t G \sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \right) \frac{1}{b_t} \\ \geq \eta_t \left[ \frac{1}{\sqrt{\beta_2}G + \delta} - \frac{1}{\delta} \left( \frac{G\sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t}{2\delta} \right) \right] \quad (129)$$

On the other hand, choosing  $\eta_t$  (learning rate)  $= \eta$ , such that:  $\frac{L\eta}{2\delta} \leq \frac{G\sqrt{1-\beta_2}}{\delta}$ , i.e.,  $\eta \leq \frac{2G\sqrt{1-\beta_2}}{L}$ , allow us to bound the term above in the following way:



$$\frac{G\sqrt{1-\beta_2}}{\delta} + \frac{L\eta}{2\delta} \leq \frac{2G\sqrt{1-\beta_2}}{\delta} \quad (130)$$

$$-\left(\frac{G\sqrt{1-\beta_2}}{\delta} + \frac{L\eta}{2\delta}\right) \geq -\frac{2G\sqrt{1-\beta_2}}{\delta} \quad (131)$$

$$\frac{1}{\sqrt{\beta_2}G + \delta} - \frac{1}{\delta} \left(\frac{G\sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t}{2\delta}\right) \geq \frac{1}{\sqrt{\beta_2}G + \delta} - \frac{2G\sqrt{1-\beta_2}}{\delta^2} \quad (132)$$

Then, let choose  $\beta_2$ , such that:  $\frac{2G\sqrt{1-\beta_2}}{\delta^2} = \frac{1}{2} \frac{1}{G+\delta}$ , we obtain:

$$\beta_2 = 1 - \frac{\delta^4}{16G^2(G+\delta)} \quad (133)$$

Bearing in mind that  $\frac{\delta^4}{16G^2(G+\delta)}$  is close to zero, then  $\beta_2$  is close to 1. Now remembering that:

$$\frac{1}{\sqrt{\beta_2}G + \delta} - \frac{1}{\delta} \left(\frac{G\sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t}{2\delta}\right) \geq \frac{1}{2} \left(\frac{1}{\sqrt{\beta_2}G + \delta}\right) \quad (134)$$

Then it can be said:

$$\frac{\eta_t}{\sqrt{\beta_2}G + \delta} - \frac{1}{\delta} \left(\frac{\eta_t G\sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta}\right) \frac{1}{b_t} \geq \eta_t \left[ \frac{1}{\sqrt{\beta_2}G + \delta} - \frac{1}{\delta} \left(\frac{G\sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t}{2\delta}\right) \right] \quad (135)$$

What give us:

$$,, \geq \frac{\eta}{2} \left(\frac{1}{\sqrt{\beta_2}G + \delta}\right) \quad (136)$$

And consequently,

$$-\left(\frac{\eta_t}{\sqrt{\beta_2}G + \delta} - \frac{1}{\delta} \left(\frac{\eta_t G\sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta}\right) \frac{1}{b_t}\right) \leq -\frac{\eta}{2} \left(\frac{1}{\sqrt{\beta_2}G + \delta}\right) \quad (137)$$

Note, that  $\eta_t = \eta$ , such that  $\frac{L\eta}{2\delta} \leq \frac{G\sqrt{1-\beta_2}}{\delta}$ , then it can be said:

$$\frac{\eta_t G\sqrt{1-\beta_2}}{\delta} + \frac{L\eta_t^2}{2\delta} \leq \frac{2\eta G\sqrt{1-\beta_2}}{\delta} \quad (138)$$

After selecting the parameters described above, the following can be written:

$$\mathbb{E}[\mathbf{L}\theta_{t+1} \mid \theta_t] \leq \mathbf{L}(\theta_t) - \|\nabla \mathbf{L}(\theta_t)\|_2^2 \left( \frac{\eta}{2} \left(\frac{1}{\sqrt{\beta_2}G + \delta}\right) + \frac{2\eta\sigma^2 G\sqrt{1-\beta_2}}{\delta^2 b_t} \right)$$

$$+\lambda G \left(1 - \frac{\eta L}{\delta}\right) + \frac{\lambda^2}{2} \quad (139)$$

Then, to reach a stationary point with a minimal error, we need to assume that  $\|\nabla_{\theta} f(\theta)\| \leq \varepsilon$ , and finalize the bounding as follows:

$$\begin{aligned} & \left(\|\nabla L(\theta_t)\|_2^2\right) \frac{\eta}{2} \left(\frac{1}{\sqrt{\beta_2}G + \delta}\right) \\ & \leq L(\theta_t) - E[L\theta_{t+1} | \theta_t] + \frac{2\eta\sigma^2 G \sqrt{1-\beta_2}}{\delta^2 b_t} + \lambda G \left(1 - \frac{\eta L}{\delta}\right) + \frac{\lambda^2}{2} \end{aligned} \quad (140)$$

To account for randomness, we may compute the total expectation as follows:

$$\begin{aligned} & \frac{1}{2\sqrt{\beta_2}G + \delta} E_{\text{total}} \left[\|\nabla L(\theta_t)\|_2^2\right] \leq \frac{E_{\text{total}} [L(\theta_t)] - E_{\text{total}} [L(\theta_{t+1})]}{\eta} \\ & + \frac{2\sigma^2 G \sqrt{1-\beta_2}}{\delta^2 b_t} + \lambda G \left(1 - \frac{\eta L}{\delta}\right) + \frac{\lambda^2}{2} \end{aligned} \quad (141)$$

Keeping the preceding in mind, it should be noted that the term  $\frac{E_{\text{total}} [L(\theta_t)] - E_{\text{total}} [L(\theta_{t+1})]}{\eta}$  comes the rule of total expectation. At this point, we need to remember that the algorithm runs from 1 to T as follows:

$$\begin{aligned} & \frac{1}{2\sqrt{\beta_2}G + \delta} \sum_{t=1}^T E_{\text{total}} \left[\|\nabla L(\theta_t)\|_2^2\right] \\ & \leq \frac{E_{\text{total}} [L(\theta_t)] - E_{\text{total}} [L(\theta_{T+1})]}{\eta} + \frac{2\sigma^2 G \sqrt{1-\beta_2}}{\delta^2} \sum_{t=1}^T \frac{1}{b_t} \\ & + \lambda G \left(1 - \frac{\eta L}{\delta}\right) + \frac{\lambda^2}{2} \end{aligned} \quad (142)$$

Then, the above relationship can be rewritten as:

$$\frac{C_1}{T} \sum_{t=1}^T E_{\text{total}} \left[\|\nabla L(\theta_t)\|_2^2\right] \leq \frac{L(\theta_t) - L(\theta_{\text{global min}})}{T\eta} + \frac{C_2}{T} \sum_{t=1}^T \frac{1}{b_t} + C_3 \quad (143)$$

Where:

$$C_1 = \frac{1}{2\sqrt{\beta_2}G + \delta} \quad (144)$$

$$C_2 = \frac{2\sigma^2 G \sqrt{1-\beta_2}}{\delta^2} \quad (145)$$

$$C_3 = \lambda G \left(1 - \frac{\eta L}{\delta}\right) + \frac{\lambda^2}{2} \quad (146)$$

In conclusion, in order for the equation to make sense, we require that:

$$\frac{L(\theta_t) - L(\theta_{\text{global min}})}{T\eta} + \frac{C_2}{T} \sum_{t=1}^T \frac{1}{b_t} + C_3 \leq \epsilon C_1 \quad (147)$$

Assuming  $b_t = b$  and a constant batch size,  $b = \left\lceil \frac{2C_2}{C_1\epsilon - 2C_3} \right\rceil$ , the following is obtained:

$$C_2 \frac{1}{T} \sum_{t=1}^T \frac{1}{b_t} = \frac{C_2}{b} \quad (148)$$

In the same manner for:

$$\frac{C_1}{T} (\ln(T) + \gamma) \leq \frac{C_1\epsilon}{2} \quad (149)$$

We can choose  $T = \frac{2(L(\theta) - L(\theta_{\text{global min}}))}{\eta C_1 \epsilon}$ , thus:

$$\frac{L(\theta) - L(\theta_{\text{global min}})}{T\eta} \leq \frac{C_1\epsilon}{2} \quad (150)$$

The process described above leads to:  $\frac{1}{T} \sum_{t=1}^T E_{\text{total}} [\|\nabla L(\theta_t)\|_2^2] \leq \epsilon$  and considering that  $T$  grows, consequently  $\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T E_{\text{total}} [\|\nabla L(\theta_t)\|_2^2] = 0$ . As a result, we have reached a stationary point, so we have proof that the ADAML algorithm converges.

#### 4.3.5 Experimentation

Using the convolutional LSTM encoder-decoder structure and Bayesian uncertainty reduction, we evaluated the performance of the ADAML optimizer. We chose the MAE-based correction because it has the lowest error rate of the three possibilities proposed by the ADAM methodology. In order to establish a comparison scenario, the ADAML forecast was compared to the predictions of four models: LSTM, LSTM stacked (Zaytar & Amrani, 2016), ARIMA (M. Murat & et al., 2018), and ADAMW. To carry out the experimentation at this stage of the research, we have again utilized a time series of 75 days of temperature acquired from the AWS network installed in the Andean city of Quito (Ecuador) (R. Llugsí & et al., 2020a). A distribution of 70%, 15%, 15%, for the training, validation, and test sets, respectively (S. El Yacoubi & et al., 2018) has been adopted to feed the neural networks. We have utilized a "down-sampling" strategy to resample the data in steps of 1 hour in order to use walk-forward validation. ADAML and ADAMW operate with  $\alpha=0.001$ ,  $\beta_1=0.9$ ,  $\beta_2=0.999$  and  $\delta=1e-08$ . To overcome any problem related to randomness, we have utilized a posterior scale ( $l_i^2$ ) of 0.01, a dropout probability ( $P_i$ ) of 0.05 and a precision of the model ( $\tau$ ) equal to 0.1. The LSTM model works with 200 neurons, while the LSTM stacked model has a structure of 150 neurons per layer. Finally, the ARIMA model is structured

utilizing a first-order auto-regressive model with a non-seasonal differencing equal to 1 and a second-order moving average model. We carried out the test considering the temperature forecast of the AWS 1 based on the temperature time series of the same station.

#### 4.3.6 Discussion of Results

Table 8 displays the error metrics (in °C) chosen to compare the accuracy of the models to the actual values of the day predicted. Also included are the correlation coefficient between actual and predicted data, as well as the mean and standard deviation of the predicted time series. Figure 14 demonstrates the efficacy of the models by

Table 8: Networks with best prediction accuracy (per error metric).

Model	MSE	RMSE	MAE	Error <sub>max</sub>	r	$\mu$	$\sigma^2$
Conv – LSTM ADAMW	0.21	0.46	0.37	1.03	0.97	18.55	0.83
Conv - LSTM ADAML	0.13	0.36	0.32	0.62	0.97	18.66	1.13
ARIMA	0.58	0.76	0.66	1.69	0.88	19.14	1.5
LSTM	0.46	0.68	0.57	1.36	0.85	18.76	1.05
Stacked LSTM	0.96	0.98	0.70	2.42	0.83	19.32	0.67

comparing a 24-hour forecast to the actual temperature.

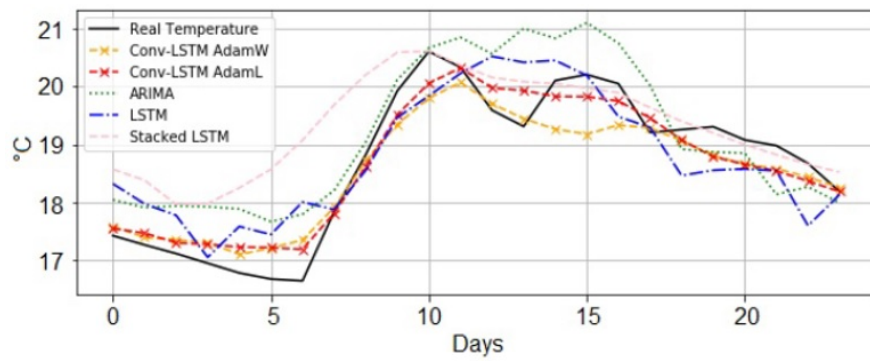


Figure 14: Comparison of error between models.

## 5 Error detection and adjustment approach

In this chapter, we describe the method for detecting errors in measurements acquired from an AWS network. For the error detection stage, we present the correlation analysis that will be used to establish the extant relationship between the AWS time series. From these relationships, we propose the formation of an AWS neighborhood that will help us determine whether a station is outside the measurement range. Lastly, we present the algorithm that will carry out the process of modifying the parameters of a failure station's sensors based on the measurements of neighboring stations. This section corresponds to the research that we conducted in the journal article: A novel approach for detecting error measurements in a network of automatic weather stations, DOI: 10.1080/17445760.2021.2022672, published in the International Journal of Parallel, Emergent, and Distributed Systems, Taylor & Francis, and incorporates some additional details.

### 5.1 Error Detection

In this section, we present the method for determining whether an AWS has measurement errors based on information from "neighboring" stations. To be more specific, the proposed method aims to determine whether a station named "Under Test" has measurement errors by analyzing the time series of two adjacent AWS. A station's AWS neighbors can be ascertained in two different methods. In the beginning, we can select the AWSs in the network whose series have the highest correlation with the AWS under analysis. Second, while considering the error, we can choose the neighboring stations whose predicted series exhibit the lowest error metrics in relation to the analyzed AWS. When discussing the correlation method, the selection is based on choosing the most highly correlated stations, taking into consideration the total quantity of data collected (75 days) from the network of stations. Consequently, we conduct weekly and daily analyses to investigate and confirm this relationship in depth.

In the case of error analysis, a 24-hour air temperature forecast for each station in the network is obtained from four neural network models. At this stage, we analyze the data from 11 days of continuous forecasts. Next, three error metrics were calculated between the actual (11-day) series of stations and their predictions. The AWS's neighbors are determined based on the error metrics with the lowest values.

#### 5.1.1 Correlation analysis of the stations of the network

A series of 75-day data obtained from each AWS in the network of stations (from 03-18-2020 to 05-31-2020) was used to analyze the first relationship approach between the stations in the network. Table 9 presents the results of the calculation of the correlation coefficient ( $r$ ) considering all time series and the distance between

stations ( $d$ ) to illustrate the spatial relationship between the network's stations.

Table 9: Correlation Coefficient ( $r$ ) and distance ( $d$ ) in kilometres between the stations in the Network.

	Station				
Station	$1_{r,d}$	$3_{r,d}$	$4_{r,d}$	$5_{r,d}$	$6_{r,d}$
$1_{r,d}$	-	0.76, 21.04	0.68, 38.46	0.47, 7.45	0.53, 12.7
$3_{r,d}$	0.76, 21.04	-	0.57, 17.48	0.47, 14.29	0.48, 8.4
$4_{r,d}$	0.68, 38.46	0.57, 17.48	-	0.32, 31.77	0.64, 25.82
$5_{r,d}$	0.47, 7.45	0.47, 14.29	0.32, 31.77	-	0.16, 6.41
$6_{r,d}$	0.53, 12.7	0.48, 8.4	0.64, 25.82	0.16, 6.41	-

Based on the analysis, we can conclude that the AWS-1 temperature data is related to the data from the AWS-3 and AWS-4 stations. Similarly, the AWS 3 time series is associated with the AWS 1 and AWS 4 time series. In contrast, for AWS 4, AWS 1 and AWS 6 produce the greatest results. Compared to AWS 1 and AWS 3, AWS 5 has the fewest relationships within the network.

The AWS 6 time series is correlated with the AWS 1 and AWS 4 time series. Considering the preceding, we can conclude that the optimal relationship between time series leads to the formation of the neighborhood between AWS 1, AWS 2, and AWS 3. See Tables 10 and 11 for a weekly and daily correlation computed for a more in-depth examination of this neighborhood.

The correlation analysis is a reasonable approximation for measuring the relationship between the data of the stations and, by extension, the weather at the locations where the stations were installed. To verify this hypothesis, we compared the forecasts generated by four distinct models for each station. To determine the accuracy of the prediction, we compared the predicted data with the actual data from the same station. Next, possible station neighborhoods are determined by comparing the actual data from the station (under analysis) with the best-correlated predictions from the other stations in the network.

We can provide a graphical representation of the advantages of correlation and the similarity between the environmental data between the stations that we are analyzing. See Figure 15 for a comparison of the data series of AWS 1 and AWS 3 (only through May 31, 2020 for presentation purposes) to contrast their similarity. As evidenced by the day-by-day and week-by-week analyses, there are nuances that cannot be completely appreciated using the correlation coefficient independently. This is why we initially

Table 10: Correlation Coefficient per week.

Weeks	$r_{AWS_1, AWS_3}$	$r_{AWS_1, AWS_4}$
13 – 05 – 2020 to 19-04-2020	0.80	0.75
20 – 05 – 2020 to 26-04-2020	0.80	0.78
27 – 04 – 2020 to 03-05-2020	0.78	0.71
04 – 05 – 2020 to 10-05-2020	0.81	0.70
11 – 05 – 2020 to 17-05-2020	0.67	0.72
18 – 05 – 2020 to 24-05-2020	0.61	0.84
25 – 05 – 2020 to 31-05-2020	0.77	0.85

Table 11: Correlation Coefficient per week.

Days	Correlations	
	$r_{AWS_1, AWS_3}$	$r_{AWS_1, AWS_4}$
25 – 05 – 2020	0.95	0.97
26 – 05 – 2020	0.83	0.90
27 – 05 – 2020	0.79	0.81
28 – 05 – 2020	0.88	0.91
29 – 05 – 2020	0.74	0.99
30 – 05 – 2020	0.83	0.99
31 – 05 – 2020	0.91	0.96



Figure 15: Graphic analysis between AWS 1 and AWS 3 (Temperature).

employed the DTW method. This method enables us to visualize the relationship between the time series of the stations; see Fig. 16. In Fig. 16, the similarity between the

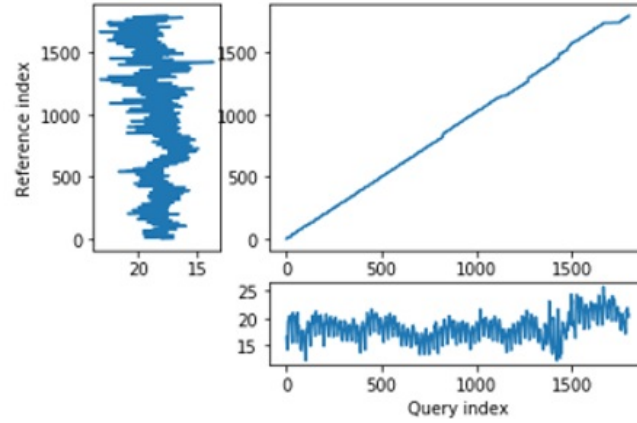


Figure 16: DTW comparison between AWS 1 & AWS 3.

AWS 1 and AWS 3 time series is depicted graphically. The technique reveals that, with the exception of the most recent data collected, the weather at each station's location is identical.

### 5.1.2 24-Hour Forecast

In this part, we provide the approach for 24-hour prediction, taking into consideration four ARIMA models, LSTM, LSTM Stacked (two layers), and a convolutional LSTM model (Llugsi & et al., 2021b) and (Llugsi & et al., 2021c), respectively.

*Neural Network configuration* For the operation of the models, we have allocated 70%, 15%, and 15% of the data set to the training, validation, and test sets, respectively. The optimal season and sample size were determined through experimentation to be 70 and 16, respectively. We determined the preceding based on the possibility that this methodology could be applied to stations with limited memory in the future. The LSTM model employs 200 neurons, whereas the stacked LSTM model employs 150 neurons per layer. ADAM optimizer is utilized by LSTM-operating models. We considered a first-order auto-regressive model with non-seasonal differentiation equal to one and a second-order moving average model when constructing the ARIMA model. In a similar fashion, the LSTM Convolutional Model employs 64 filters in the convolutional layer and 300 neurons in the LSTM layer. Finally, we utilized the ADAM Logger (ADAML) optimizer to optimize the LSTM convolutional network. We employ this method because the incorporation of a weight decay decoupling into the optimization procedure enhances the neural network forecasts. This is due to the fact that the addition of the weight decay rate to the update rule enhances the search for the lowest error in the cost function during the learning process (Llugsi & et al., 2021b), (Llugsi & et al., 2021c).  $\alpha = 0.001$ ,  $\beta-1 = 0.9$ ,  $\beta-2 = 0.999$  y  $\varepsilon = 1 \text{ e-}8$  and the dropout regularization  $P_i$  and precision of the model  $\tau$  for AWS 1 and AWS 2 were



0.05 and 0.1, respectively.

*Comparison between real data and Forecast* We present the results of error metrics and the correlation coefficient obtained from the comparison between the actual and predicted series over a period of 11 days (extra days to the initial data period of 75 days) in Table 12. This was done to verify the accuracy of models developed as suggested in (P. Sharma & Sharma, 2021). To get a more precise idea of the forecast,

Table 12: Error metrics and the associated Correlation for every station in the network.

Station	Model	MSE	RMSE	MAE	r
AWS 1	ARIMA	3.36	1.83	1.22	0.42
AWS 1	LSTM	1.88	1.37	1.05	0.5
AWS 1	LSTM Stacked	1.84	1.36	1.00	0.52
AWS 1	Conv. LSTM	3.50	1.87	1.55	0.25
AWS 3	ARIMA	2.52	1.59	1.11	0.7
AWS 3	LSTM	10.57	3.25	2.58	0.3
AWS 3	LSTM Stacked	11.43	3.38	2.64	0.2
AWS 3	Conv. LSTM	3.88	1.97	1.50	0.4
AWS 4	ARIMA	1.84	1.36	0.97	0.81
AWS 4	LSTM	2.89	1.70	1.34	0.76
AWS 4	LSTM Stacked	2.26	1.50	1.22	0.79
AWS 4	Conv. LSTM	2.46	1.57	1.16	0.80
AWS 5	ARIMA	0.99	0.99	0.75	0.78
AWS 5	LSTM	3.44	1.86	1.46	0.49
AWS 5	LSTM Stacked	3.71	1.93	1.45	0.55
AWS 5	Conv. LSTM	4.57	2.14	1.67	0.34
AWS 6	ARIMA	3.95	1.99	1.46	0.68
AWS 6	LSTM	5.61	2.37	1.92	0.46
AWS 6	LSTM Stacked	4.52	2.13	1.73	0.56
AWS 6	Conv. LSTM	3.92	1.98	1.59	0.62

in Fig. 17 we present a graphical comparison between the 24-hour forecast obtained over the past 11 days. From the trials, we can say that, in almost all cases, the ARIMA

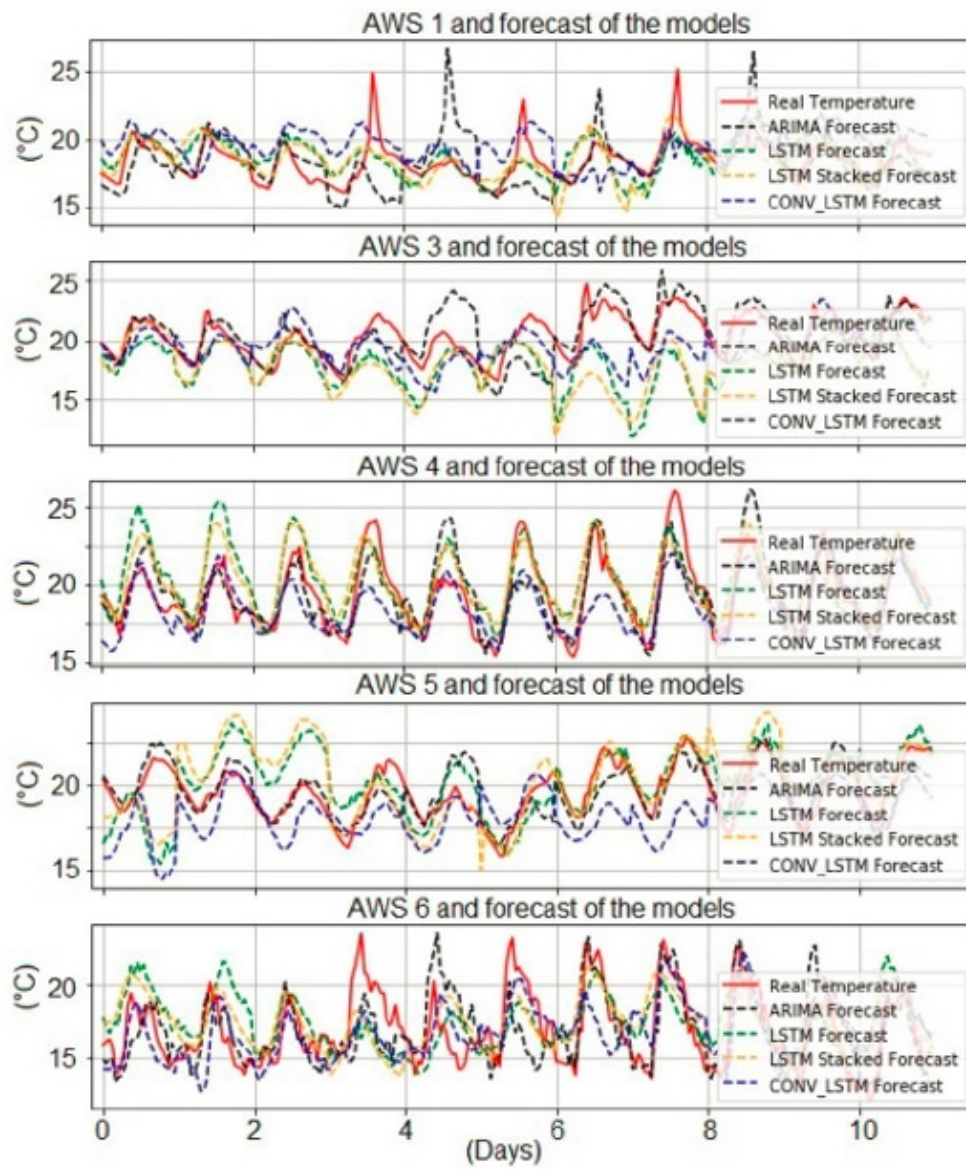


Figure 17: Comparison between real temperature and forecast.

model is the neural network that works best. Stations AWS 5, AWS 4 and AWS 3 with MSE equal to  $0.99^{\circ}\text{C}$ ,  $1.84^{\circ}\text{C}$  and  $2.52^{\circ}\text{C}$  respectively, are the stations where ARIMA predictions are more reliable.

The model with lower error prediction for AWS 1 was LSTM stacked with MSE equal to  $1.84^{\circ}\text{C}$  and for AWS 6, there are two models performing well: the convolutional LSTM model with MSE equal to  $3.92^{\circ}\text{C}$  and ARIMA with  $3.95^{\circ}\text{C}$ . On the other hand, the strongest linear relationship between the actual and predicted series (according to the correlation coefficient) is present in all the stations with the ARIMA model except AWS 1, where the LSTM stacked model presents the best correlation.

*Comparison between forecast data from neighbors* To determine the optimal relationship and identify the stations' neighbors, we compare the forecast temperatures of all AWS with each station's actual data. Table 13 illustrates the error metrics used to determine the stations' neighbors and the correlation coefficients associated with this selection. This procedure is a variation on the method described in (H. Astsatryan & et al., 2021), and with it we can observe that the correlation experimentally demonstrates the degree of linear dependence between the time series.

The forecast and the real value of the AWS under analysis are shown in Fig. 18, to

Table 13: Exemplary Table

Station	Neighbour	Model	MSE	RMSE	MAE	r
AWS 1	AWS 4	Conv. LSTM	1.59	1.26	0.98	0.69
AWS 1	AWS 5	Conv. LSTM	3.03	1.74	1.34	0.22
AWS 3	AWS 4	Conv. LSTM	2.69	1.64	1.36	0.62
AWS 3	AWS 5	Conv. LSTM	3.11	1.76	1.40	0.54
AWS 4	AWS 3	Conv. LSTM	3.39	1.84	1.50	0.57
AWS 4	AWS 1	Conv. LSTM	4.00	1.84	1.50	0.44
AWS 5	AWS 1	Conv. LSTM	2.98	1.73	1.41	0.21
AWS 5	AWS 3	Conv. LSTM	3.11	1.77	1.43	0.35
AWS 6	AWS 4	Conv. LSTM	7.28	2.70	2.28	0.54
AWS 6	AWS 3	LSTM	7.34	2.71	2.19	0.36

clarify the results of the selection. There are differences between the preceding procedure and the first method discussed in this section. We observe a close relationship between the actual temperature at AWS 1 and the forecast at AWS 4 and AWS 5. The optimal relationship for AWS 3 is between AWS 4 and 5. The AWS 4 series has a

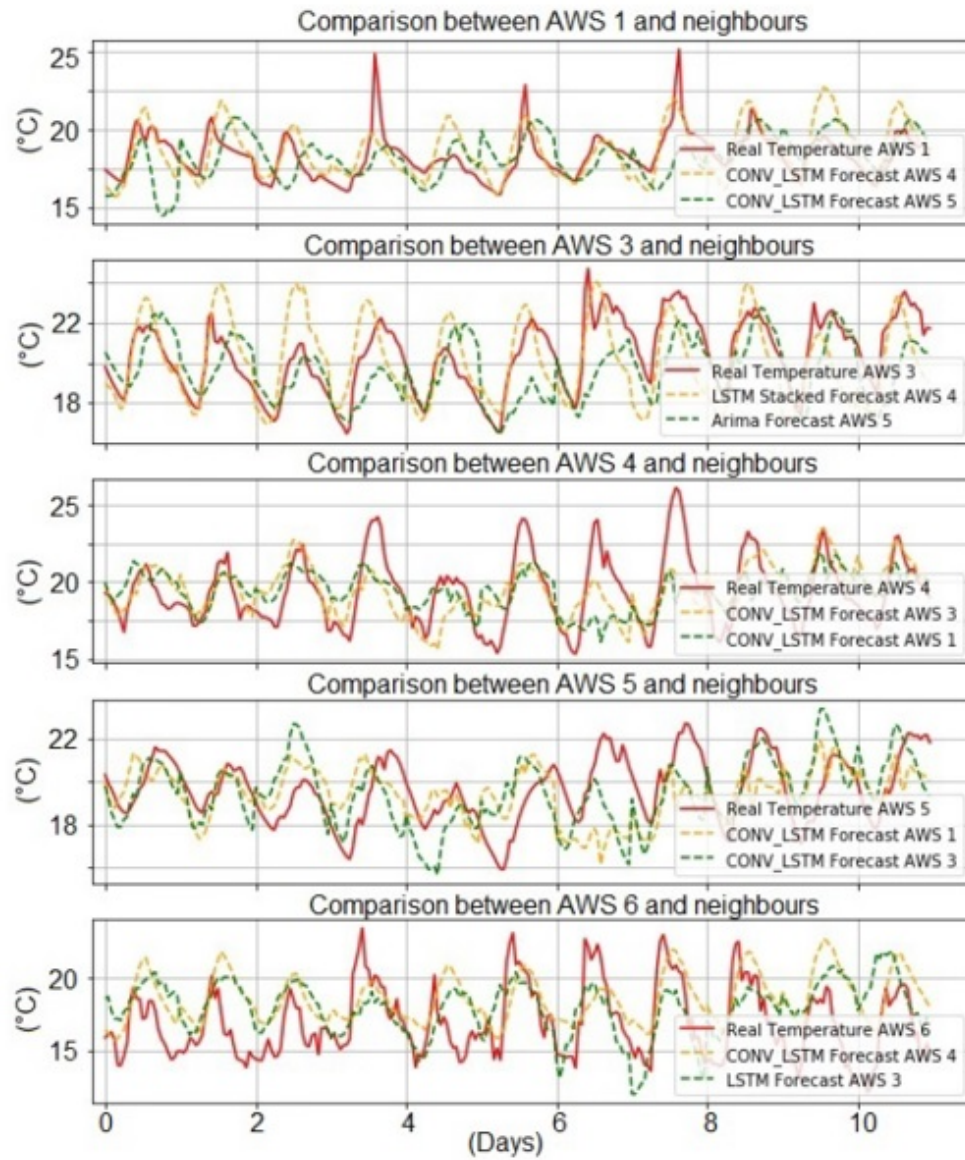


Figure 18: Comparison of forecasted temperature for the AWS Network.

strong relationship with AWS 1 and 3, while the AWS 5 series is related to AWS 1 and 3. However, it is necessary to note that the method cannot be used to determine a neighborhood for AWS 6.

### 5.1.3 Discussion of results

Through the research, we can say that the determination of the neighbours of the AWSs based on the correlation coefficients and error metrics is effective. That is, through these methodologies we can effectively determine if a station experiences a deviation in temperature measurements. We can inspect the relationship between the actual data of the station under analysis and its neighbours by visualizing the distribution of the observations in the data set thanks to the Kernel Density Estimation (KDE) diagram, see Fig. 19. The stations AWS 1, AWS 3, and AWS 4 have a strong linear

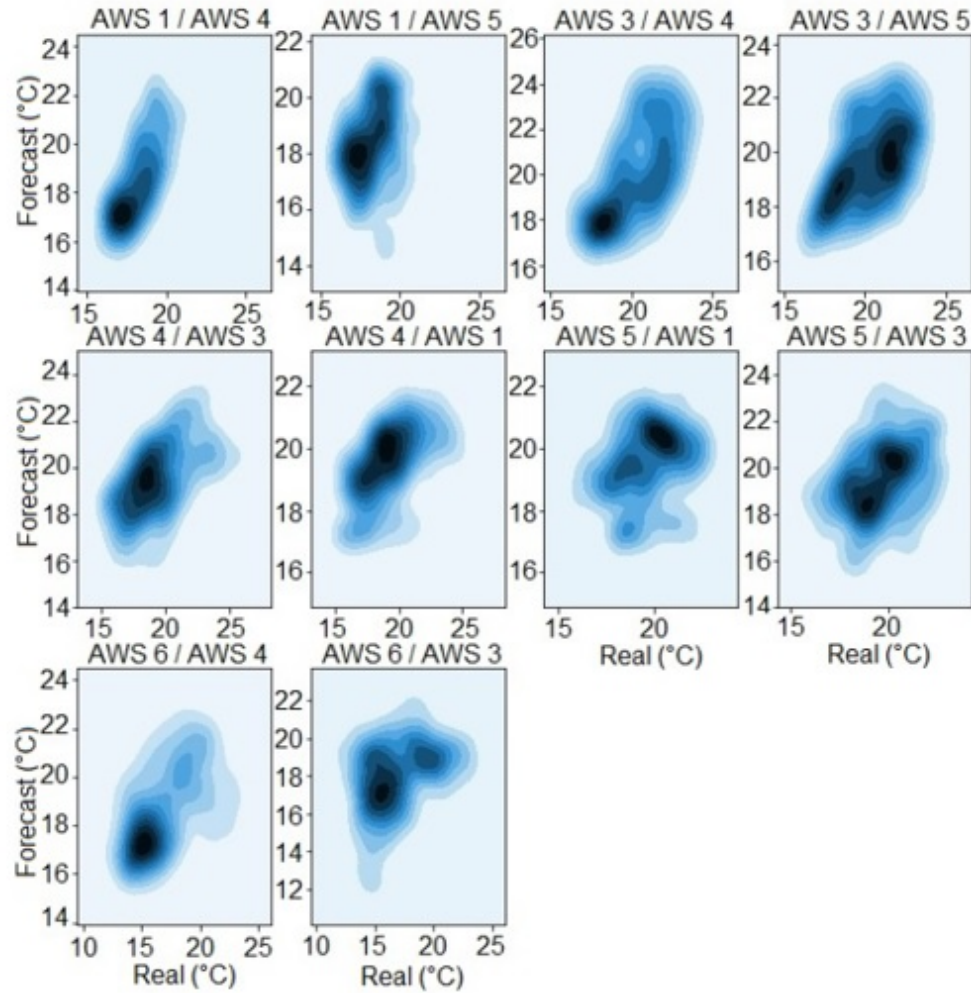


Figure 19: Visualizing the distribution of observations in the dataset.

relationship with their respective communities; see Fig. 19. In light of the correlation coefficient and error metrics, this allows us to assert that there is at least one neighboring station that is significantly associated with the previously mentioned stations. In the case of AWS 1, the neighboring stations with the strongest relationships are AWS

3 and AWS 4, which have minimum daily correlation coefficients of 0.74 and 0.81, respectively, as shown in Table 11. In contrast, based on the lowest error metrics, the selected neighbors were AWS 4 and AWS 5, with respective MAE values of 0.98°C and 1.34°C. From the two analyses, we can conclude that AWS 4 is the station most closely related to the station under analysis. Similarly, we can deduce that the optimal relationship for AWS 3 and AWS 4 is with AWS 4 and AWS 1 (and AWS 3); see Table 14.

When a time series exhibits a particular trend, it is easier to anticipate its future

Table 14: Determination of station neighbourhoods and thresholds based on correlation coefficients and MAE.

	AWS Neighbours	
Weeks	(r)	(MAE)
AWS 1	3(0.74) and 4(0.81)	4(0.98°C) and 5(1.34°C)
AWS 3	1(0.74) and 4(0.72)	4(1.36°C) and 5(1.40°C)
AWS 4	1(0.81) and 3(0.72)	1(1.50°C) and 3(1.50°C)
AWS 5	3(0.44) and 4(0.02)	1(1.41°C) and 3(1.43°C)
AWS 6	1(0.46) and 3(0.29)	3(2.28°C) and 4(2.19°C)

behavior, as there is a high likelihood that it will continue to behave in the same manner (Walasek & Gajda, 2021). In the instance of the Andean city of Quito, however, weather forecasting is difficult. To demonstrate this variability and its effect on the temperature forecast, we investigated the mean and standard deviation of the group of 11 days selected for the analysis of the forecast. Below are the minimum and maximum values derived per day for the mean ( $\sigma^2$ ) and variance (*sigma2*); see Table 15. Analyzing the mean values included in Table 15, we can conclude that AWS 1 is the station with the most seasonal time series, followed by AWS 4, AWS 5, and AWS 6. Nonetheless, when examining Fig. 17, it is important to note that the series derived from AWS 1 and AWS 3 demonstrate that the climate of Quito can change dramatically in a matter of days. AWS 4 has the most consistent weather, whereas AWS 3 and AWS 6 are located in areas where the weather is highly variable and therefore its forecast is difficult to obtain. The previous variations are attributable to the dramatic climate change between the south and north of Quito. Inspecting AWS 1 and AWS 3, we can conclude that the most abrupt alterations occur within days. However, we cannot compare the microclimates of the aforementioned regions because the variations occurred on separate days. Similarly, we can generally state that the climate in the



Table 15: Exemplary Table

Station	$\mu_{\min}$	$\mu_{\max}$	$\sigma_{\min}^2$	$\sigma_{\max}^2$
AWS 1	17.52	19.46	0.18	4.65
AWS 3	18.96	21.87	1.39	3.61
AWS 4	18.60	20.53	1.13	10.21
AWS 5	18.27	20.57	0.34	3.26
AWS 6	15.91	18.38	1.97	9.72

various locations where AWSs were installed is related. AWS 4 is located in the area of Quito with the most reliable weather conditions. Tests confirm that the climate of Quito is highly variable, but relatively stable, and therefore predictable.

## 5.2 Auto-Adjutment

To address the loss of precision of a meteorological parameter, we propose employing the methodology proposed in the preceding section regarding the weather forecast of stations in the vicinity of an AWS network. Using time series, we seek information on the short-term (24-hour) weather forecast of two neighboring stations to the station under analysis (SUT). Because it is possible for a neighboring station to have measurement errors that could result in a superfluous adjustment to the SUT, we select two stations.

### 5.2.1 Auto-Adjustment Process

We have developed a test scenario to evaluate the error detection phase and subsequent error correction strategy. To initiate the process, we have generated a time series of errors with a low correlation coefficient relative to the weekly correlation of each station of the network (obtained in the previous section). Thus, we initially regard the weekly correlation derived in the preceding section as the process's thresholds. However, as will be seen later, we modified them experimentally.

To detect the error, we obtain the 24-hour temperature forecast for each of the SUT's neighbors and then compare them with the actual station's series using the Correlation Coefficient Calculated (Ccal). If the predictions exceed the correlation threshold, Correlation Coefficient Typical (CTyp), the adjustment procedure is initiated. The adjustment procedure entails calculating the forecasts of the two nearest neighbors, averaging them, and using the least squares technique to determine the equation coefficient that modifies the sensor data.

To verify if we can adjust the data acquired with the proposed process, we experimentally modify the data based on the coefficient previously calculated. Then, the procedure is repeated, i.e., comparing the neighboring series with the temperature series of the station under test. If the process is successful, the algorithm stops; otherwise, it is calculated again (see Fig. 20).

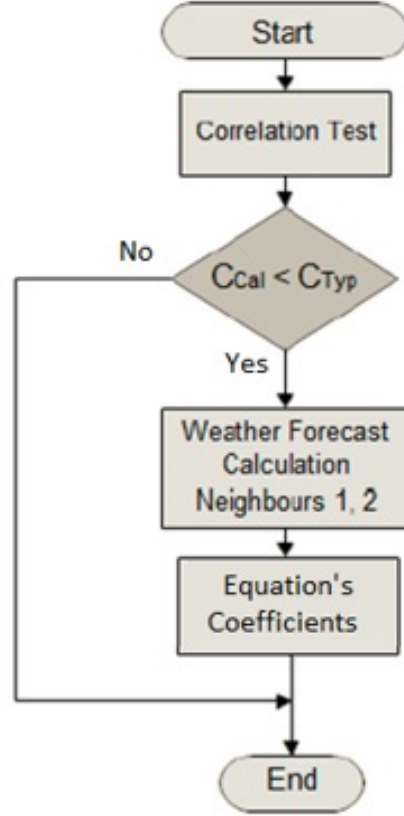


Figure 20: Flow diagram of the auto-adjustment process.

### 5.2.2 Relationship between series

We are able to determine the trigger threshold thanks to the correlation analysis that we conducted in the previous section. Nevertheless, during the experimentation stage, we defined more precise threshold values for the process to be successful. Based on our experiments, the following thresholds have been determined for each station (see Table. 16).

In this section, we can propose the following parameters for conducting experiments: The data distribution for training, validation and testing is 70%, 15%, 15% respectively. Epoch and batch sizes were 70 and 16, respectively. ADAM optimization algorithm with learning rate = 0.001,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 1e-8$ . Posterior scale  $l_i^2$  of 0.01. ADAML operates with  $\alpha = 0.001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  y  $\epsilon = 1e-8$ . Finally, dropout regularization  $P_i$  and precision of the model  $\tau$  for AWS 1 and AWS 2 were 0.05 and 0.1 respectively (for the relationship with AWS 2), and 0.05 and 0.01 respec-



Table 16: Relationship between the stations

Station	Neighbour 1	Threshold 1	Neighbour 2	Threshold 2
AWS 1	AWS 3	0.7	AWS 4	0.8
AWS 3	AWS 1	0.6	AWS 4	0.7
AWS 4	AWS 1	0.8	AWS 3	0.8
AWS 5	AWS 3	0.8	AWS 4	0.8
AWS 6	AWS 1	0.8	AWS 3	0.9

tively (for the relationship with AWS 3). Dropout regularization  $P_i$  and precision of the model  $\tau$  for AWS 2 were 0.05 and 0.3 respectively, for the relationship with AWS 1, and 0.05 and 0.02 respectively, for the relationship with AWS 5.

### 5.2.3 Discussion of results

Based on the results of the experiments, two stations, AWS 1 and AWS 3, present the best results for carrying out the tasks of self-adjustment based on the information of the neighboring stations, see Fig. 21 and Fig. 22.

We can see that the stations in the south (AWS 1) and the north (AWS 3) of the city

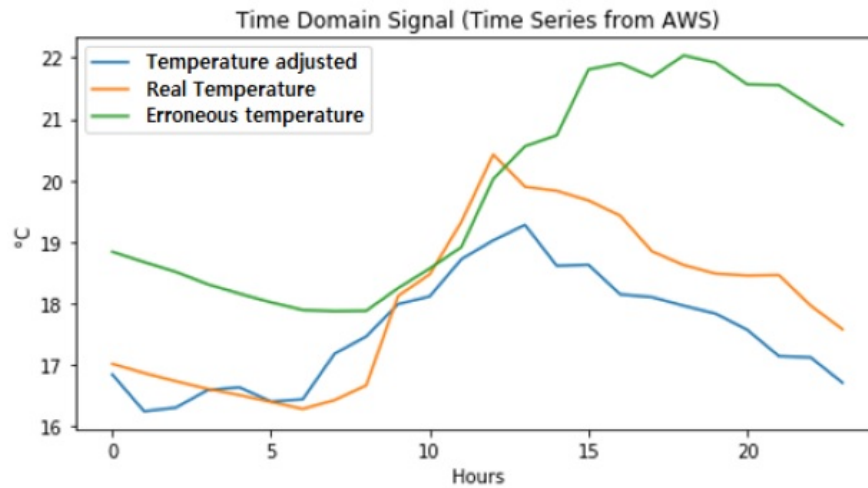


Figure 21: Result of the adjustment process for AWS 1.

provide an adequate response to the methodology proposed in this work. Despite meeting the correlation threshold, the adjusted data for stations AWS 4, AWS 5, and AWS 6 are distinct from the pattern distribution of the actual data. This may be the result of a lack of test stations to define in greater detail the environmental relationships that may exist between the various city sectors. Considering the preceding, we can confirm that the meteorology in the rugged terrain geography of Quito varies significantly over

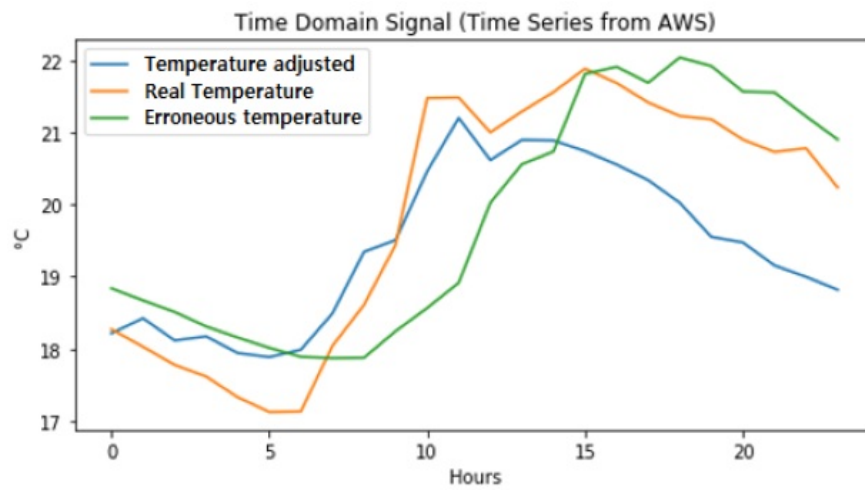


Figure 22: Result of the adjustment process for AWS 3.

a relatively small area.

## 6 General Conclusions and Future work

### 6.1 Conclusions

This study focuses on the development of micro-automatic weather stations with auto-adjustment of the parameter measurement deviation (error of the measured parameter) using a Bayesian approach and adaptive moment estimation to minimize uncertainty. Using a classical neural network model (i.e. LSTM or GRU), we acquire the lowest values of RMSE, MSE, and MAE when the network predicts the temperature at a single station using the temperatures from all stations in the network. When we applied the new method, however, we obtained the lowest error metrics when we used only the temperature series of each AWS to generate the forecast at that station. We proposed using LSTM structures, stacked LSTM structures (with two layers), ARIMA models, and the convolutional encoder-decoder structure to establish the various test scenarios.

In this study, we have devised a Bayesian-based uncertainty approach that enables us to improve the behavior of the models. Using the dropout regularization technique, we reduced the uncertainty in the forecast of a neural network (applied to short-term weather forecasting) by adjusting the posterior predictive distribution based on estimators. By injecting stochastic noise into the models, we can use stochastic regularization techniques to enhance the learning of the neurons in the network. With weight-decays of  $2.04 \times 10^{-7}$  and  $2.23 \times 10^{-7}$ , we obtained a maximum error forecast of 12% and demonstrated that for LSTM, the variation of the error is reduced by nearly half. In addition, the Walk Forward validation is an essential instrument for conducting the research and facilitating the analysis of the forecast error over the next 24 hours, which is an acceptable time range for a short-term weather forecast for Quito.

To complement the strategy of uncertainty reduction, we propose decoupling the weight decay from the gradient-based update in order to enhance the models' learning process. To put forward this strategy, we developed a novel approach called ADAML. To prove the convergence of this approach, we utilized three assumptions: i) the loss function is L-smooth; ii) the loss function has a bound gradient; and iii) the variance of the loss is bound. From the trials conducted, we can see that the modification to the update rule of ADAM,  $\theta_{t-1} = \theta_t - \eta_t \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \delta}} - \lambda$ , converges optimally, and using the parameters  $\alpha = 0.001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  and  $\delta = 1 \times 10^{-8}$ , we reduce the forecast's error effectively. The configuration of the network meets a very good performance with a maximum error of  $0.62^\circ\text{C}$  and we visually confirmed this bearing in mind that the forecast curve presents a trend similar to the actual data of the predicted day. The results show that the combination of the concepts of ADAML, walk-forward validation, and Bayesian theory performs better than the approaches pre-

sented in (Zaytar & Amrani, 2016) and (M. Murat & et al., 2018). Which involves the generation of a powerful combination to get a proper weather forecast for the Andean region, especially at latitude  $0^{\circ}$ .

On the other hand, in Chapter 5, we proposed two approaches to determining measurement errors at weather stations that were based on the use of error metrics or correlation analysis to build station neighborhoods. In the first approach, we build the station neighborhoods by correlating 75-day data from each station in the network. While in the second approach, we obtain the lowest error metrics between the series predicted for 11 extra days of the stations and the actual values of the station under analysis to build the neighborhood of stations. The outcome of the research reveals that the northern central part of the city experiences the most variable weather conditions of all the areas selected for monitoring.

Through the analysis, a correlation between network station information and station communities is determined. Using this methodology, we would ascertain whether a station has measurement errors based on the variation of data relative to the data of neighboring stations, thereby ensuring the integrity of the data. This is possible because we constructed station neighborhoods by identifying the stations that are most closely related to the station under analysis. Three stations, AWS 1, AWS 3, and AWS 4, achieved MAE values between  $0.98^{\circ}\text{C}$  and  $1.50^{\circ}\text{C}$  and correlation coefficients between 0.72 and 0.81 based on the results of the analysis. Intriguingly, despite a distance of 38.46 kilometers and a decline of 400 meters, the closest relationship appears to exist between AWS 1 and AWS 4. We can justify this based on the fact that the city's central-northern region has the greatest temperature variation. In addition, we can confirm that the AWS 1, AWS 4, and AWS 6 stations had a maximal average temperature difference of approximately  $20^{\circ}\text{C}$ . In contrast, the AWS 3 and AWS 5 stations attained an average temperature of  $22^{\circ}\text{C}$ , which is quite intriguing given that these stations are installed on the slopes of the Pichincha volcano, one in the north of the city and the other in the south. We discovered that the meteorological forecast generated by a neural network from the time series of two stations highly related to the Station Under Test (SUT) can be used to adjust a particular environmental parameter at a station with a failed measurement (at a correlation coefficient threshold of 0.8). We verify that the average forecast derived from the time series of the stations adjacent to the SUTs can be used to modify (via an equation) the measurement of a failed sensor.

Interestingly, the proposed methodology provides an unanticipated solution to Quito's lack of ground stations: the compilation of meteorological data at specific locations without the need for physical stations. To illustrate the aforementioned, we can imagine installing roving stations at various locations in the south and north of the city for a few months. Then, with the aid of "main" stations, we are able to generate environmental data specific to these locations.

Finally, based on the analysis conducted during this study, we can divide Quito's climate into three zones. The south and far north regions of Quito, which maintain a relatively constant temperature, and the central and near-north regions, which experience cooler weather than the rest of the city.

## **6.2 Limitations and future work**

It is crucial to highlight that, despite the benefits of the technique that was suggested in this study effort, there are certain limitations that emerged when doing the experiments, including the following:

In the case of adjustments based on data from neighboring stations, it is evident that obtaining a 24-hour forecast reduces the precision of the sensor calibration equation settings. This is due to the fact that the standard procedure for obtaining the calibration equation of a sensor in the laboratory necessitates a broad range of variation, such as temperature. In Ecuador, the average temperature variation ranges from 9°C to 24°C, limiting our ability to derive an equation that permits precise sensor adjustments at a weather station. Our research team believes that by obtaining an extended time prediction, such as 72 hours, we could enhance the technique's accuracy. Nonetheless, operating with a relatively narrow temperature range would always be a limitation. This is a future-applicable proposal so long as we can continue to operate with a sufficient network of stations.

An additional limitation of the approach is that we originally proposed to work with the information processing in each AWS of the network of stations. We are unable to implement the above proposal, and it is preferable to perform the processing on a computer located outside of the network. The decision was made because, while it is true that the processing capacity of the Raspberry Pis, a component of the network stations, enables the implementation of models on Amazon Web Services (AWS), the process is sluggish and requires a large amount of computational resources. This could result in an unsuccessful acquisition of meteorological parameters by the Raspberry. In the future, we could propose implementing the automated modification methodology on other types of platforms in order to determine whether the processing can be performed on each AWS in the network.

It is important to note that at the outset of the investigation, we proposed installing 16 stations to encompass multiple areas of Quito. The construction costs for each AWS, which included electronic equipment and a protection enclosure, amounted to approximately 300 dollars per unit. Consequently, we resolved to establish five stations throughout Quito. Our research team believes that this issue could be addressed and resolved in the future by securing funding and enhancing the methodology presented in this study.

It is important to observe that, based on the data collected by each station in Quito, it was determined that certain areas could become new data collection sites of interest. However, the COVID-19 pandemic had a subsequent impact on the location and care of each station, as in many cases the individuals responsible for the care of the stations had to flee the city and were unable to monitor the condition of the equipment.

Regarding the Calibration of the sensors, we proposed to carry out a calibration with a secondary pattern, namely, in the field. We implemented this philosophy along the research to adjust and maintain the measurement of the information of the AWSs of the network. We made this decision considering the high cost of calibration for each AWS, which was around USD 400. Additionally, it must be said that the calibration can only be carried out at INAMHI, which is the only institution in the country that has an adequate laboratory for the calibration of meteorological stations and that to date still does not officially have the permits to issue calibration certificates but can only generate calibration reports. We consider that this problem could be treated and solved in the future by seeking funding and improving the technique presented in this research.

For the city's weather division, we have to indicate that despite the fact that the behavior (at the three zones) remains nearly stable for all the research data collected, we still need to implement additional stations in the city's far north. This will help us comprehend the weather on the city's outskirts in greater detail.

## References

- A. Alzahrani, P. S., & et al. (2017). Solar irradiance forecasting using deep neural networks. *Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, 114*, 304-313. doi: 10.1016/j.procs.2017.09.045
- Aggarwal, C. (2018). Neural networks and deep learning. *Springer Nature*. doi: 10.1007/978-3-319-94463-0
- Bochenek, B., & Ustrnul, Z. (2022). Machine learning in weather prediction and climate analyses—applications and perspectives. *MDPI. Atmosphere, 13*(180). doi: 10.3390/atmos13020180
- Comercio. (2020). Quito se convirtió en la ciudad más poblada del ecuador con más de 2,7 millones de habitantes en el 2018. *El Comercio*.
- D. Dotlic, B. I., & et al. (2019). Uncertainty in profit scoring (bayesian deep learning). *Seminar Information Systems (WS18/19). Humboldt-Universität zu Berlin*.
- D. Kreuzer, M. M., & et al. (2020). Short-term temperature forecasts using a convolutional neural network – an application to different weather stations in germany. *Machine Learning with Applications, 2*(15). doi: 10.1016/j.mlwa.2020.100007
- E. Abrahamsen, O. B., & et al. (2018). Machine learning in python for weather forecast based on freely available weather data. *Proceedings of the 59th Conference on Simulation and Modelling (SIMS 59)*, 169-176.
- E. Pope, D. S., & Jackson, D. (2020). An adaptive markov chain approach for probabilistic forecasting of categorical events. *Monthly Weather Review, 148*(9), 3681-3691. doi: 10.1175/MWR-D-19-0239.1
- Gal, Y., & Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. *30th Conference on Neural Information Processing Systems (NIPS 2016)*.
- Giorgino, T. (2009). Computing and visualizing dynamic time warping alignments in r: The dtw package. *Journal of Statistical Software*.
- G. Zhang, B. P., & et al. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting, 14*, 35–62. doi: 10.1016/S0169-2070(97)00044-7
- Géron, A. (2017). Hands-on machine learning with scikit-learn and tensorflow: Concepts, tools, and techniques to build intelligent systems. *O'Reilly Media*.
- H. Astsatryan, H. G., & et al. (2021). Air temperature forecasting using artificial neural network for ararat valley. *Earth Science Informatics, 14*, 711–722. doi: 10.1007/s12145-021-00583-9
- H. Bluestein, F. C., & et al. (2022). Atmospheric observations of weather and climate. *Atmosphere-Ocean, 60*, 149-187. doi: 10.1080/07055900.2022.2082369

- Huang, C., & Kuo, P. (2018). A deep cnn-lstm model for particulate matter (pm2.5) forecasting in smart cities. *Sensor Magazine*, 18(7). doi: 10.3390/s18072220
- I. Kamal, H. B., & et al. (2020). Dern: Deep ensemble learning model for shortand long-term prediction of baltic dry index. *MDPI. Applied Sciences*, 10. doi: 10.3390/app10041504
- INAMHI. (2016). Analysis of the impact of the main elements of the climate in the ecuadorian agricultural sector. *Meteorological Studies and Research, INAMHI*.
- INEC. (2019). Methodological document of the continuous agricultural production and area survey (espac). *Directorate of Agricultural and Environmental Statistics, INEC*.
- Jain, P., & Kar, P. (2017). Non-convex optimization for machine learning (foundations and trends in machine learning). *Now publishers Inc*.
- J. Cullman, M. D., & et al. (2019). State of climate services. agriculture and food services. *World Meteorological Organization (WMO)*, 142.
- J. Frnda, M. D., & et al. (2019). A weather forecast model accuracy analysis and ecmwf enhancement proposal by neural network. *MDPI. Sensors*, 19(23). doi: 10.3390/s19235144
- Jurafsky, D., & Martin, J. (2021). Speech and language processing. *University of Stanford*.
- J. Yao, W. P., & et al. (2019). Quality of uncertainty quantification for bayesian neural network inference. *ICML Workshop on Uncertainty and Robustness in Deep Learning*.
- Khiatani, K., & Ghose, U. (2017). Weather forecasting using hidden markov model. *International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, 220-225. doi: 10.1109/IC3TSN.2017.8284480
- Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.
- Krogh, A., & Herts, J. (1991). A simple weight decay can improve generalization. *Proceedings of the 4th International Conference on Neural Information Processing Systems*, 950–957. doi: 10.5555/2986916.2987033
- L. Cardelli, M. K., & et al. (2019). Statistical guarantees for the robustness of bayesian neural networks. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.
- Llugsí, R., & et al. (2021a). Comparison between adam, adamax and adamw optimizers to implement a weather forecast based on neural networks for the andean city of quito. *ETCM, IEEE*. doi: 10.1109/ETCM53643.2021.9590681
- Llugsí, R., & et al. (2021b). A novel adam approach related to decoupling weight decay (adaml). *Latin American Conference on Computational Intelligence*. doi: 10.1109/LA-CCI48322.2021.9769816



- Llugsi, R., & et al. (2021c). A novel encoder-decoder structure for time series analysis based on bayesian uncertainty reduction. *Latin American Conference on Computational Intelligence*. doi: 10.1109/LA-CCI48322.2021.9769850
- Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*.
- M. Alsharif, M. Y., & et al. (2019). Time series arima model for prediction of daily and monthly average global solar radiation: the case study of seoul, south korea. *MDPI. Symmetry*, 11(2). doi: 10.3390/sym11020240
- Matich, D. J. (2001). Redes neuronales: Conceptos básicos y aplicaciones. *Universidad Tecnológica Nacional—Facultad Regional Rosario, Departamento de Ingeniería Química*.
- Maule. (2019). Effects of frost on agriculture: Know how to deal with low temperatures. *Catholic University of Maule*.
- M. Danilova, P. D., & et al. (2020). Recent theoretical advances in non-convex optimization. *ArXiv*.
- M. Jordan, Z. G., & et al. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2), 183 – 233. doi: 10.1023/A:1007665907178
- M. J. Uddin, M., Y. Li, & et al. (2022). Effects of learning rates and optimization algorithms on forecasting accuracy of hourly typhoon rainfall: Experiments with convolutional neural network. *Earth and Space Science*, 9. doi: 10.1029/2021EA002168
- M. Murat, I. M., & et al. (2018). Forecasting daily meteorological time series using arima and regression models. *International Agrophysics*, 32(2), 253–264.
- M. P. Byrne, A. D. R., A. G. Pendergrass, & Wodzicki, K. R. (2018). Response of the intertropical convergence zone to climate change: Location, width, and strength. *Current Climate Change Reports*, 4, 355–370. doi: 10.1007/s40641-018-0110-5
- M. Schultz, B. G., C. Betancourt, & et al. (2021). Can deep learning beat numerical weather prediction? *Philosophical Transaction of the Royal Society*. doi: 10.1098/rsta.2020.0097
- Muck, P., & Homam, M. (2018). Iot based weather station using raspberry pi 3. *International Journal of Engineering & Technology*, 7, 145 – 148.
- M. Vladimirova, J. V., & et al. (2019). Understanding priors in bayesian neural networks at the unit level. *ICML 2019 - 36th International Conference on Machine Learning*.
- M. Zaheer, S. R., & et al. (2018). Adaptive methods for nonconvex optimization. *Advances in Neural Information Processing Systems*.
- Narayanaa, F., & Turhan, L. (2018). Preserving order of data when validating defect prediction models. *California Polytechnic State University*.

- Naser, M., & Alavi, A. (2021). Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences. *Architecture, Structures and Construction*. doi: 10.1007/s44150-021-00015-8
- Neal, R. (1996). Bayesian learning for neural networks. lecture notes in statistics. *Springer*.
- Oke, T. (2006). Initial guidance to obtain representative meteorological observations at urban sites. instruments and observing methods report. *World Meteorological Organization (WMO)*, 81.
- O'Carroll, C., & Leslie, J. (2009). Goes-o, geostationary operational environmental satellites. *NASA's Goddard Space Flight Center, NOAA National Environmental Satellite, Data and Information Service*.
- Perin, G., & Picek, S. (2021). On the influence of optimizers in deep learning-based side-channel analysis. selected areas in cryptography. *Lecture Notes in Computer Science*. doi: 10.1007/978-3-030-81652-0\_24
- P. Ladyzynski, Z. Z., & et al. (2013). Stock trading with random forests, trend detection tests and force index volume indicators. *International Conference on Artificial Intelligence and Soft Computing ICAISC*, 7895, 441-452. doi: 10.1007/978-3-642-38610-7\_41
- Postalcioglu, S. (2020). Performance analysis of different optimizers for deep learning-based image recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(2). doi: 10.1142/S0218001420510039
- P. Schober, C. B., & et al. (2018). Correlation coefficients: Appropriate use and interpretation. *Anesthesia & Analgesia*. doi: 10.1213/ANE.0000000000002864
- P. Sharma, S. S., & Sharma, S. (2021). Artificial neural network approach for hydrologic river flow time series forecasting. *Agricultural Research*, 11, 465-476. doi: 10.1007/s40003-021-00585-5
- P. Tormene, T. G., & et al. (2008). Matching incomplete time series with dynamic time warping: An algorithm and an application to post-stroke rehabilitation. *Artificial Intelligence in Medicine*, 45. doi: 10.1016/j.artmed.2008.11.007
- Raschka, S. (2018). Model evaluation, model selection, and algorithmselection in machine learning. *University of Wisconsin-Madison*.
- Rasp, S., & Lerch, S. (2018). Neural networks for postprocessing ensemble weather forecasts. *Waves Weather*, 146, 3885-3900. doi: 10.1175/MWR-D-18-0187.1
- R. Llugsi, A. F., & et al. (2020a). Deep learning to implement a statistical weather forecast for the andean city of quito. *ANDESCON, IEEE*. doi: 10.1109/ANDESCON50619.2020.9272106

- R. Llugsi, A. F., & et al. (2020b). Uncertainty reduction in the neural network's weather forecast for the andean city of quito through the adjustment of the posterior predictive distribution based on estimators. *Conference on Information and Communication Technologies of Ecuador, 1307*, 535-548. doi: 10.1007/978-3-030-62833-8\_39
- R. Sieber, L. A., V. Slonosky, & Pudmenzky, C. (2022). Formalizing trust in historical weather data. *Weather, Climate, and Society, 14*(3), 993-1007. doi: 10.1175/WCAS-D-21-0077.1
- Sagheer, A., & Kotb, M. (2018). Unsupervised pretraining of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems. *Scientific Reports Nature, 9*, 1-16. doi: 10.1038/s41598-019-55320-6
- Scher, S., & Messori, G. (2019). Weather and climate forecasting with neural networks: using general circulation models (gcms) with different complexity as a study ground. *European Geosciences Union, Copernicus Publications, 2797 – 2809*.
- Sedgwick, P. (2012). Pearson's correlation coefficient. *British Medical Journal*.
- S. El Yacoubi, M. F., & et al. (2018). A multilayer perceptron model for the correlation between satellite data and soil vulnerability in the ferlo. *International Journal of Parallel, Emergent and Distributed Systems, 34*(1), 3-12. doi: 10.1080/17445760.2018.1434175
- S. Liu, B. K., & et al. (2018). Zeroth-order stochastic variance reduction for non-convex optimization. *Advances in Neural Information Processing Systems, 31*, 3731–3741.
- S. Liu, P. C., & et al. (2020). A primer on zeroth-order optimization in signal processing and machine learning. *IEEE Signal Processing Magazine, 37*(5). doi: 10.1109/MSP.2020.3003837
- S. Serrano, J. R., & et al. (2017). Heavy rainfall and temperature proyections in a climate change scenario over quito, ecuador. *La Granja Revista de Ciencias de la Vida, 25*, 16-32. doi: 10.17163/lgr.n25.2017.02
- S. Sun, G. Z., & et al. (2019). Functional variational bayesian neural networks. *International Conference on Learning Representations*.
- T. Dimri, S. A., & Sharif, M. (2020). Time series analysis of climate variables using seasonal arima approach. *Journal of Earth System Science, 129*(1), 1–16. doi: 10.1007/s12040-020-01408-x
- Unesco. (2020). City of quito. *Unesco*.
- Universitario. (2020). Inamhi cumple 51 años de contribuir al progreso del país. *Ecuador Universitario*.

- V. Athira, P. G., & et al. (2018). Deepairnet: Applying recurrent networks for air quality prediction. *International Conference on Computational Intelligence and Data Science*, 132, 1394-1403. doi: 10.1016/j.procs.2018.05.068
- Villacis, E., & Marrero, N. (2017). Precipitaciones extremas en la ciudad de quito, provincia de pichincha- ecuador. *ING. Hidráulica y ambiental*, 38(2), 102–113.
- Walasek, R., & Gajda, J. (2021). Fractional differentiation and its use in machine learning. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, 13, 270–277. doi: 10.1007/s12572-021-00299-5
- Willink, R., & White, R. (2012). Disentangling classical and bayesian approaches to uncertainty analysis. *Technical Report No. CCT/12-08. BIPM: Sevres*.
- Witten, I., & Frank, E. (2005). Data mining: Practical machine learning tools and techniques. *Morgan Kaufmann*.
- WMO. (2008). Guide to meteorological instruments and methods of observation. *WMO*, 8.
- X. ZongBen, Z. H., & et al. (2010). L 1/2 regularization. *Science China Information Sciences*, 53, 1159–1169. doi: 10.1007/s11432-010-0090-0
- Y. Arjevani, Y. C., & et al. (2020). Second-order information in non-convex stochastic optimization: Power and limitations. *Proceedings of Thirty Third Conference on Learning Theory*.
- Y. Tao, L. M., & et al. (2018). Hierarchical attention-based recurrent highway networks for time series prediction. *Cornell University*.
- Zaytar, M., & Amrani, C. E. (2016). Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. *International Journal in Computing Applications*, 143(11), 7–11.

## **Appendix**

### **A Published Articles**

### **B Calibration reports**

## **Appendix I**

### Published Articles

# Deep Learning to implement a Statistical Weather Forecast for the Andean City of Quito

Publisher: IEEE

[Cite This](#) PDFRicardo Llugsí Cañar ; Allyx Fontaine ; Pablo Lupera Morillo ; Samira El Yacoubi [All Authors](#)

100

Full

Text Views



## Abstract

### Document Sections

I. Introduction

II. State of the Art

III. Network  
Implementation

IV. Methodology

V. Experimentation

[Show Full Outline ▼](#)[Authors](#)[Figures](#)[References](#)[Keywords](#)[Metrics](#)

## Abstract:

A weather forecast is classically based on the use of a Physical Model computing information from Conventional Weather Stations (CWS), Automatic Weather Stations (AWS) and Geostationary Operational Environmental Satellite (GOES) imagery. For the Andean city of Quito, located at the latitude 0° this methodology turns to be inefficient and inaccurate. Several reasons can justify this problem but two can be highlighted: the sensibility of the Intertropical Convergence Zone and the geographic pattern of the Andean Region (microclimates). Both problems can be solved through the acquisition of new stations, nevertheless, this solution is expensive. In this paper, an investigative approach built on low cost small single-board computers (used as AWS) and Deep Learning is presented for solving the above. Eight neural networks for Temperature Forecast have been validated in base of the Walk Forward Technique. And temperature or Temperature/Humidity Time Series have been entered into the networks to find what could be the best inputs to generate the predictions. Finally, errors between 0.74 °C and 2.24°C have been obtained from the predictions and it was determined that the lowest values of error are obtained when one prediction hinges on the temperature series of all the stations in the network.

Published in: 2020 IEEE ANDESCON

Date of Conference: 13-16 October 2020

INSPEC Accession Number: 20179047

Date Added to IEEE Xplore: 01 December 2020

DOI:  
[10.1109/ANDESCON50619.2020.9272106](https://doi.org/10.1109/ANDESCON50619.2020.9272106)

► ISBN Information:

Publisher: IEEE

Conference Location: Quito, Ecuador

## I. Introduction

Nowadays the environmental data acquisition in the Andean city of Quito is carried out with the use of CWS and AWS. The deployment of this stations is done in large areas, which implies that the weather behaviour in the zone is not represented accurately [1]. The data acquired is processed then with Mathematical models running in Regional offices of the World Meteorological Organization (WMO) [2]. But it is crucial for the models to work with environmental data in real time to implement a proper Weather Forecast. New approaches such as low cost small single-board computers configured as AWS are becoming more popular because of specific advantages such as Portability, Real Time Data acquisition/transmission and Internet Connectivity [3]. Additionally, new Weather Forecast Techniques currently use Deep Learning for Time Series [4], [5] and satellite imagery [6], [7]. Thus, the fast development of the Deep Learning paradigm combined with the new Low-Cost Hardware enhancement allows the creation of powerful processing platforms (in an accurate and inexpensive way) Weather Forecast that can be unthinkable 12 years ago [8]. In this paper a Forecast of Temperature for the Andean city of Quito using eight different neural networks is

[Sign in to Continue Reading](#)



Conference on Information and Communication Technologies of Ecuador

↳ TICEC 2020: **Information and Communication Technologies** pp 535–548 | [Cite as](#)

[Home](#) > [Information and Communication Technologies](#) > Conference paper

# Uncertainty Reduction in the Neural Network's Weather Forecast for the Andean City of Quito Through the Adjustment of the Posterior Predictive Distribution Based on Estimators

[Ricardo Llusi](#) , [Allyx Fontaine](#), [Pablo Lupera](#), [Jessica Bechet](#) & [Samira El Yacoubi](#)

Conference paper | [First Online: 11 November 2020](#)

**590** Accesses | **5** Citations

Part of the [Communications in Computer and Information Science](#) book series (CCIS, volume 1307)

## Abstract

The weather forecast in cities as Quito is highly complicated due to its proximity to Latitude 0° and because it is located in the Andes mountains range. A statistical post-processing is compulsory in order to improve the output from the physical model and to improve the weather forecast in the city. A neural network can be applied in order to carry out this task but it is necessary first to reduce its uncertainty. The Bayesian Neural Networks (BNN) have been studied deeply thanks to its probability analysis, the uncertainty can be approximated. In this paper an analysis founded on the adjustment of the posterior predictive distribution based on estimators is carried out in order to reduce the prediction error variation (implicitly the uncertainty) in a Short-Term Weather Forecast for the Andean city of Quito. From the analysis it is obtained a maximum error forecast of 12% and it is proven that for Long Short Term Memory (LSTM) structures, the variation of the error reduces almost to the half with weight-decays of  $2.04 \times 10^{-7}$  and  $2.23 \times 10^{-7}$ .

## Keywords

Neural network

Uncertainty

Bayesian

Weight decay

Walk forward validation



# Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito

Publisher: IEEE

[Cite This](#)[PDF](#)Ricardo Llugsí ; Samira El Yacoubi ; Allyx Fontaine ; Pablo Lupera [All Authors](#)

4

Paper

Citations

993

Full

Text Views

**Abstract**

## Document Sections

- I. Introduction
- II. Neuronal Networks and Weather Forecast
- III. Experimentation
- IV. Results and Discussion
- V. Conclusions

[Authors](#)[Figures](#)[References](#)[Citations](#)[Keywords](#)[Metrics](#)**Abstract:**

The main function of an optimizer is to determine in what measure to change the weights and the learning rate of the neural network to reduce losses. One of the best known optimizers is Adam, which main advantage is the invariance of the magnitudes of the parameter updates with respect to the change of scale of the gradient. However, other optimizers are often chosen because they generalize in a better manner. AdamW is a variant of Adam where the weight decay is performed only after controlling the parameter-wise step size. In order to present a comparative scenario for optimizers in the present work, a Temperature Forecast for the Andean city of Quito using a neural network structure with uncertainty reduction was implemented and three optimizers (Adam, AdaMax and AdamW) were analyzed. In order to do the comparison three error metrics were obtained per hour in order to determine the effectiveness of the prediction. From the analysis it can be seen that Adam and AdaMax behave similarly reaching a maximum MSE per hour of 2.5°C nevertheless AdamW allows to reduce this error around 1.3°C.

**Published in:** 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)**Date of Conference:** 12-15 October 2021**INSPEC Accession Number:** 21418813**Date Added to IEEE Xplore:** 10 November 2021**DOI:** 10.1109/ETCM53643.2021.9590681**Publisher:** IEEE**► ISBN Information:****Conference Location:** Cuenca, Ecuador**I. Introduction**

The fundamental function of the optimization algorithms is to find the optimization values of the appropriate neural network weights to minimize the objective function (loss / cost function). Currently the most used optimization techniques to train neural networks are based on the use of gradients, such as backpropagation. However, the use of algorithms based on gradient determination are very limited in their ability to find generalized solutions. Decoupled Decay Regularization techniques have been identified as a possible solution to this problem. In [1] and [2] the currently best-known fields of application of neural networks, object recognition and speech recognition, respectively, are described. Other fields in which these networks have been used recently are the economy [3] and the environment [4], where the fundamental task is the achievement of a parameter prediction based on the analysis of time series. One of the newest fields today for the application of neural networks is the environment. Since, as seen previously, the achievement of a Meteorological Forecast is very useful when implementing a statistical post-processing task to correct the forecast of a system's physical model [5]. There is evidence from several authors of the use of neural networks for prognosis has been demonstrated [4], [6]. At present the use of basic recurring networks for the treatment of Time Series has been almost completely put aside to give way to the use of

[Sign in to Continue Reading](#)

# A novel Encoder-Decoder structure for Time Series analysis based on Bayesian Uncertainty reduction

**Publisher:** IEEE

[Cite This](#)

[PDF](#)

Ricardo Llugsí ; Samira El Yacoubi ; Allyx Fontaine ; Pablo Lupera [All Authors](#)

34

Full

Text Views



## Abstract

### Document Sections

I. Introduction

II. Neural networks  
and Weather  
Forecast

II. Experimentation

III. Results and  
Discussion

III. Conclusions

[Authors](#)

[Figures](#)

[References](#)

[Keywords](#)

[Metrics](#)

## Abstract:

In the present work, a novel Convolutional LSTM Encoder-Decoder structure for the implementation of Weather Forecast for the Andean city of Quito is presented. Aside from the above, the Encoder-Decoder structure uses a Walk-Forward validation, an adjustment of the Bayesian posterior predictive distribution and the ADAMW optimizer to carry out the forecast. The aforementioned stages are combined to obtain 4 error metrics per hour. The prediction is done in base of acquired data from a network of Automatic Weather Stations. The results show that the Convolutional Encoder-Decoder structure with a dropout probability of 0.05 and a model precision equal to 0.1 performs better than a LSTM model, LSTM Stacked model or ARIMA models reaching a maximum error of 1.03 °C. Finally, the methodology could be applied as an effective option to implement the post-processing stage for the physical model of a Weather Forecast System.

**Published in:** [2021 IEEE Latin American Conference on Computational Intelligence \(LA-CCI\)](#)

**Date of Conference:** 02-04 November 2021

**INSPEC Accession Number:** 21757404

**Date Added to IEEE Xplore:** 11 May 2022

**DOI:** [10.1109/LA-CCI48322.2021.9769850](#)

**► ISBN Information:**

**Publisher:** IEEE

**Conference Location:** Temuco, Chile

## I. Introduction

Currently, neural networks are applied to the analysis of images and audio to carry out tasks such as object [1] and speech recognition [2]. However, neural networks have also begun to be used to implement Time Series analysis and carry out tasks such as forecasting. This has attracted attention in areas related to economy [3] and environment [4]. In [4] a 12-hours meteorological forecast is obtained through the use of a Stacked LSTM model. The information entered to the model was obtained from Meteorological Stations installed in 9 Moroccan airports. In this case, the error per hour is in the range of approx. 0.01 °C to 3 °C. On the other hand, in [5] the air temperature and precipitation Time Series recorded between 01-01-1980 and 31-12-2010 in the cities of Jokioinen, Dikopshof, Lleida and Lublin were modelled and successfully forecasted using a seasonal ARIMA model. In this case, it can be seen that the error (based on the simple difference between the forecast and the real measured value) per hour falls in the range of between approximately 3.02 °C and 4.5 °C. It is interesting to note that the structures Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) and Convolutional Long Short-Term Memory (ConvLSTM) have been adopted as the most common models for time series analysis. In this paper, a novel Encoder-Decoder structure for the implementation of Weather Forecast for the Andean city of Quito is presented. Aside from the above, the Encoder-Decoder structure uses a Walk-Forward validation, an adjustment of the Bayesian posterior predictive distribution and the ADAMW optimizer to carry out the forecast. The aforementioned stages are combined to obtain 4 error metrics per hour. The prediction is done in base of acquired data from a network of Automatic Weather Stations. The results show that the Convolutional Encoder-Decoder structure with a dropout probability of 0.05 and a model precision equal to 0.1 performs better than a LSTM model, LSTM Stacked model or ARIMA models reaching a maximum error of 1.03 °C. Finally, the methodology could be applied as an effective option to implement the post-processing stage for the physical model of a Weather Forecast System.

[Sign in to Continue Reading](#)

# A novel Adam approach related to Decoupled Weight Decay (AdamL)

**Publisher:** IEEE[Cite This](#) PDFRicardo Llugsí ; Samira El Yacoubi ; Allyx Fontaine ; Pablo Lupera [All Authors](#)

48

Full

Text Views



## Abstract

### Document Sections

- I. Introduction
- II. Nonconvex Optimization & Weight Decay
- III. Features of the Neural Network
- IV. Experimentation
- V. Conclusions

### Authors

### Figures

### References

### Keywords

### Metrics

## Abstract:

The use of optimizers makes it possible to reduce losses during the learning process of a neural network. Currently there are some types of optimizers whose effectiveness has already been proven, an example of this is Adam. Adam is an extension to Stochastic Gradient Decent that makes use of Momentum and Adaptive Learning to converge faster. An interesting alternative to complement the Adam's work is the addition of weight decay. This is done to decouple the weight decay from the gradient-based update. Some attempts have been developed previously, however its correct operation has not been keenly proven. In this work, a weight decay decoupling alternative is presented and acutely analyzed. The algorithm's convergence is mathematically verified and its operation too through the use of a Convolutional Encoder-Decoder network and the application of strategies for error reduction. The AdamL operation is verified by the achievement of a proper Temperature Forecast with a percentage error lower than 4.5%. It can be seen too that the forecast error deepens around noon but it does not exceed 1.47°C.

**Published in:** 2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)**Date of Conference:** 02-04 November 2021 **INSPEC Accession Number:** 21757832**Date Added to IEEE Xplore:** 11 May 2022 **DOI:** 10.1109/LA-CCI48322.2021.9769816**► ISBN Information:****Publisher:** IEEE**Conference Location:** Temuco, Chile

## I. Introduction

Bearing in mind that trying to find global and local minima in non-convex optimization can be classified as NP-hard [1], it is recommended to look for stationary points, which are located where  $\nabla_{\theta} f(\theta_{\text{stationary}}) = 0$ . In order to do that 2 scenarios can be proposed, the determination of the  $\epsilon$  - First Order Stationary Point (FOSP) or the localization of the  $\epsilon$  - Second Order Stationary Points (SOSP) where  $\|\nabla_{\theta} f(\theta_{\text{FOSP}})\|_2 \leq \epsilon$  and [2]. In the first case the points to be determined are global and local minima, saddle and plateau points while in the second case the global and local minima and saddle points are determined.

[Sign in to Continue Reading](#)

Authors



Figures



References





## A novel approach for detecting error measurements in a network of automatic weather stations

R. Llusi, S. El Yacoubi, A. Fontaine & P. Lupera


To cite this article: R. Llusi, S. El Yacoubi, A. Fontaine & P. Lupera (2022): A novel approach for detecting error measurements in a network of automatic weather stations, *International Journal of Parallel, Emergent and Distributed Systems*, DOI: [10.1080/17445760.2021.2022672](https://doi.org/10.1080/17445760.2021.2022672)

To link to this article: <https://doi.org/10.1080/17445760.2021.2022672>




Published online: 06 Jan 2022.




Submit your article to this journal 



View related articles 



View Crossmark data 

## Appendix II

### Calibration reports

## REGISTRO DE CALIBRACIÓN DE TEMPERATURA

*Temperature Calibration Register*

### DATOS GENERALES

*General Information*

Usuario: Ricardo Llugsí

*User*

Dirección Fiscal: Legarda

*Legal Address*

Nº de Solicitud: S-014

*Application Number*

### DATOS DEL INSTRUMENTO

*Instrument Information*

Objeto: Sensor de Temperatura

*Object*

Fabricante / Marca: ASAIR

*Manufacturer / Brand*

Modelo / Tipo: AM2320B

*Model / Type*

Serial: 1709004EE5

*Serial number*

Rango: -40 a 80 °C

*Range*

División/Resolución: 0.1 °C

*Division/Resolution*

Código: OT-2506

*Code*

Fecha de Recepción: 2021-06-09

*Reception date*

Fecha de Calibración: 2021-06-15

*Calibration date*

Este registro de calibración proporciona evidencia documental para la trazabilidad a los patrones nacionales, llevados a cabo por las unidades de medición de acuerdo con el Sistema Internacional de Unidades. (SI).

*This calibration register provides documentary evidence*

*for the traceability to national standards, carried out by*

*the units of measurement according to the international*

*System of Units (SI).*

### INAMHI

Es responsabilidad del usuario establecer la frecuencia de calibración de este instrumento. Esta declaración es aspecto auditable en el sistema de gestión en su empresa.

*The user shall be responsible for establishing the calibration frequency of this instrument. This statement is an auditable aspect in the management system of his company.*



Firmado electrónicamente por:

**CESAR DAVID  
TONATO  
PERALTA**

MSc. David Tonato

COORDINADOR DEL LABORATORIO

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL.

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

Dirección: Calle Núñez de Vela N36-15 y Corea, Quito - Ecuador.

Teléfonos: (593 - 2) 397 11 00 Ext. 87058. website: [www.inamhi.gob.ec](http://www.inamhi.gob.ec). correo: [labmet@inamhi.gob.ec](mailto:labmet@inamhi.gob.ec).

## PROCEDIMIENTO

Calibración efectuada usando el método de comparación directa con nuestros patrones, según lo establecido en el procedimiento "LMET-PC-01 CALIBRACION DE SENSORES DE TEMPERATURA"

## CONDICIONES AMBIENTALES

TEMPERATURA AMBIENTE:	(	20.7	±	0.6	)	°C
HUMEDAD RELATIVA:	(	51.8	±	5.0	)	%HR
PRESIÓN ATMOSFÉRICA:	(	732.0	±	2.5	)	hPa

## PATRONES

### TERMOMETRO DE ESTANDARES CHUB-E4

MARCA	FLUKE
MODELO	1529-R
SERIAL/CÓDIGO	B05221
Nº CERTIFICADO	SC

Rango °C	Incertidumbre	Correccion °C	Resolucion (°C) :	Deriva °C
-38.834	± 25 mK	0.01	0.0001	
29.765	± 20 mK	0.01	0.0001	0.001
156.599	± 30 mK	0.01	0.0001	

Temperatura Ambiente de Calibracion: 23 °C

### Termómetro con Resistencia de Platino 100 Ω:

MARCA	FLUKE Hart Scientific
MODELO	5626
SERIAL/CÓDIGO	2373
Nº CERTIFICADO	LNMT-20171300007D

Rango °C	Incertidumbre	Correccion °C	Resolucion (°C) :	Deriva °C
-38.834	± 25 mK	0.01	0.0001	
29.765	± 20 mK	0.01	0.0001	0.001
156.599	± 30 mK	0.01	0.0001	

Temperatura Ambiente de Calibracion: 23 °C

### Termómetro con Resistencia de Platino 100 Ω:

MARCA	RTD Company
MODELO	5616-12
SERIAL/CÓDIGO	54328
Nº CERTIFICADO	14095

Rango °C	Incertidumbre	Correccion °C	Resolucion (°C) :	Deriva °C
-38.009	± 0.011 °C	0.012	0.001	
0.006	± 0.009 °C	0.01	0.001	0.001
156.015	± 0.011 °C	0.01	0.001	

Temperatura Ambiente de Calibracion: 23 °C

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

**PATRONES**

**HORNO DE POZO SECO**

MARCA FLUKE  
MODELO 9171  
SERIAL/CÓDIGO B25585  
Nº CERTIFICADO LNM-T-201715300028D  
INCERTIDUMBRE: Inmersión B al 100%

Temperatura °C	U	Estabilidad	Axial	Radial
-10	± ( 0.0816	0.038	0.079	0.046 )
20	± ( 0.0734	0.027	0.053	0.050 )
60	± ( 0.0923	0.044	0.104	0.050 )

Inmersión C al 70%

Temperatura °C	U	Estabilidad	Axial	Radial
-10	± ( 0.0809	0.036	0.079	0.046 )
20	± ( 0.0731	0.027	0.053	0.049 )
60	± ( 0.091	0.032	0.104	0.054 )

Temperatura Ambiente de Calibración: 23 °C

**LUGAR DONDE SE REALIZA LA CALIBRACIÓN:**

La calibración fué realizada en el laboratorio de Temperatura y Humedad, localizado en Calle Nuñez de Vela N36-15 y Corea, Quito - Ecuador.

Se establecen las diferencias entre la lectura "LI" del Instrumento a calibrar y la lectura "LP" del patrón a fin de obtener el error absoluto.

$$EA = LI - LP$$

**El resultado corregido del instrumento viene dado por:**

$$RC = LI + C, \text{ siendo } C = \text{Corrección} = -EA$$

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

Dirección: Calle Núñez de Vela N36-15 y Corea, Quito - Ecuador.  
Teléfonos: (593 - 2) 397 11 00 Ext. 87058. website: www.inamhi.gob.ec. correo:labmet@inamhi.gob.ec.



**Magnitud: Temperatura**
**Especificaciones Técnicas:**
**RANGOS:**
**EXACTITUD:**
**Resolución del**
**Instrumento**

( - 40.0      a    80.0    ) °C      ± (    0.500    ) °C      0.1 °C

**PARAMETRO SIN AJUSTE**

Nominal	LI	LP	EA	± EMP	± U <sub>exp</sub> (K=2)	UNIDAD	GRÁFICO
-10°C	-9.4	-9.785	0.385	0.50	0.14	°C	G-1
1°C	1.2	1.085	0.125	0.50	0.13		
10°C	9.9	10.047	- 0.127	0.50	0.12		
20°C	19.7	19.974	- 0.234	0.50	0.15		
30°C	29.3	29.887	- 0.557	0.50	0.20		
40°C	39.4	39.798	- 0.398	0.50	0.18		

**PARAMETRO CON AJUSTE**

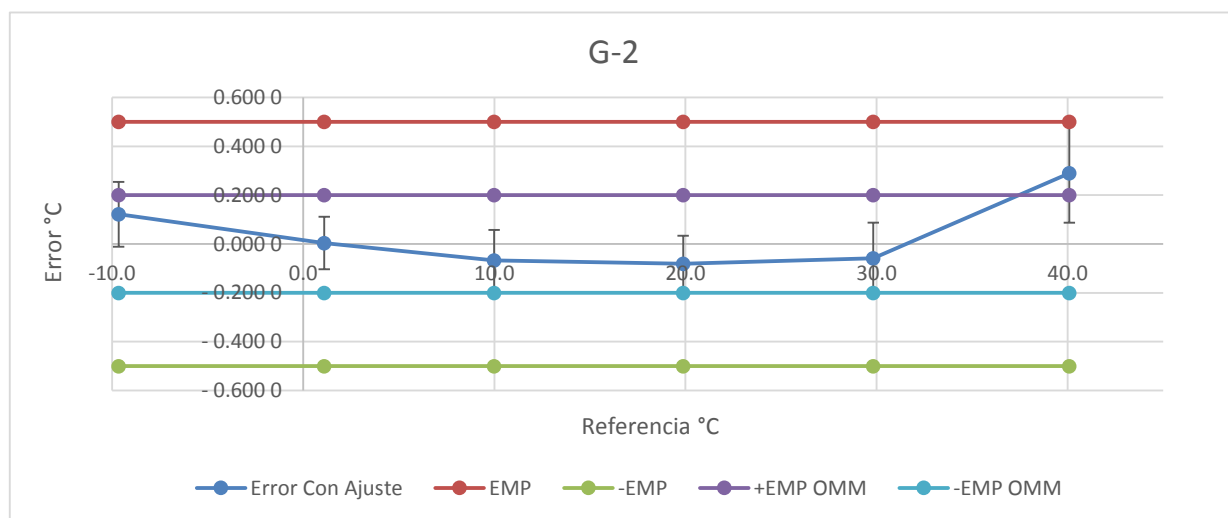
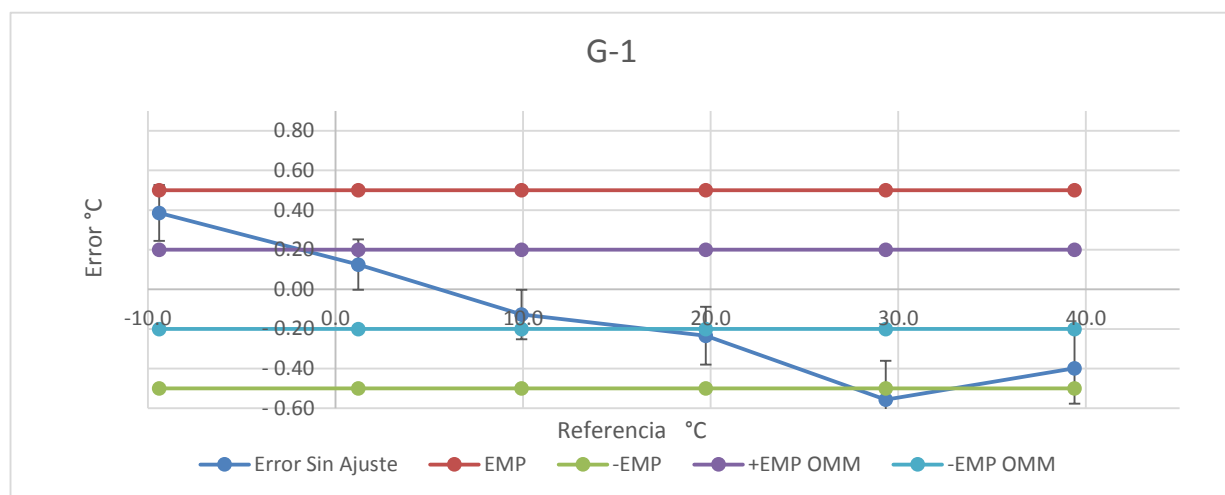
Nominal	LI	LP	EA	± EMP	± U <sub>exp</sub> (K=2)	UNIDAD	GRÁFICO
-10°C	-9.7	-9.781	0.121	0.50	0.13	°C	G-2
1°C	1.1	1.096	0.004	0.50	0.107		
10°C	10.0	10.057	- 0.067	0.50	0.125		
20°C	19.9	19.961	- 0.081	0.50	0.115		
30°C	29.8	29.889	- 0.059	0.50	0.15		
40°C	40.1	39.801	0.289	0.50	0.20		

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

Dirección: Calle Núñez de Vela N36-15 y Corea, Quito - Ecuador.

Teléfonos: (593 - 2) 397 11 00 Ext. 87058. website: [www.inamhi.gob.ec](http://www.inamhi.gob.ec). correo: [labmet@inamhi.gob.ec](mailto:labmet@inamhi.gob.ec).



ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

Dirección: Calle Núñez de Vela N36-15 y Corea, Quito - Ecuador.  
Teléfonos: (593 - 2) 397 11 00 Ext. 87058. website: [www.inamhi.gob.ec](http://www.inamhi.gob.ec). correo: [labmet@inamhi.gob.ec](mailto:labmet@inamhi.gob.ec).

**INCERTIDUMBRE:**

“La incertidumbre expandida reportada de la medición se establece como la incertidumbre de medición estándar multiplicada por el factor de cobertura  $k$  calculado, de tal manera que la probabilidad de cobertura corresponde a aproximadamente 95%”.

**OBSERVACIONES:**

Después del proceso de verificación el sensor necesitó coeficientes de corrección y aún así no cumple con la normativa OMM a la temperatura de 40°C, pero sí con las especificaciones técnicas de fábrica por lo que se recomienda no operarlo en condiciones cercanas a esa temperatura para aplicaciones meteorológicas y realizar las verificaciones pertinentes para determinar la deriva en el tiempo del sensor.

Reporte Sin Ajuste			
<i>Parametros antes del ajuste</i>			
Curva de ajuste original		Coeficientes antes del ajuste	
Curva Reportada	Por defecto	a0	0.000000
Modificación Realizada	No	a1	1.000000
Tipo de curva	Polinomio	a2	
Grado de Polinomio	1	a3	

Reporte de Ajuste			
<i>Parametros despues del ajuste</i>			
Curva de ajuste		Coeficientes despues del ajuste	
Curva Reportada	Por defecto	a0	-0.13471825
Modificación Realizada	No	a1	1.01790345
Tipo de curva	Polinomio	a2	
Grado de Polinomio	1	a3	

Resultados de Ajuste y Recalibración	
Tolerancia maxima en °C Especificaciones Técnicas	0.50
Tolerancia maxima en °C por la OMM	0.20
Como se encontró	Fuera de Tolerancia
Como se dejó	En Tolerancia
Ajuste realizado	Sí
Fecha sugerida por la OMM para la próxima Calibración	2022-06-16



Firmado electrónicamente por:  
**JIMMY SEBASTIAN  
 NARVAEZ ORDONEZ**

Fís. Jimmy Narváez  
 RESPONSABLE TÉCNICO



Firmado electrónicamente por:  
**SANTIAGO FERNANDO  
 RAMON ZAMBRANO**

Ing. Santiago Ramón  
 RESPONSABLE DE CALIDAD

LOS RESULTADOS SE REFIEREN ÚNICAMENTE AL INSTRUMENTO ANTERIORMENTE DESCRITO

Fin del registro de calibración.

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

## REGISTRO DE CALIBRACIÓN DE HUMEDAD RELATIVA

*Relative Humidity Calibration Register*

### DATOS GENERALES

*General Information*

Usuario: Ricardo Llugsí

*User*

Dirección Fiscal: Legarda

*Legal Address*

Nº de Solicitud: S-014

*Application Number*

### DATOS DEL INSTRUMENTO

*Instrument Information*

Objeto: Sensor de Humedad Relativa

*Object*

Fabricante / Marca: ASAIR

*Manufacturer / Brand*

Modelo / Tipo: AM2320B

*Model / Type*

Serial: 170904EE5

*Serial number*

Rango: 0 A 100%

*Range*

División/Resolución: 0.1%

*Division/Resolution*

Código: OT-2505

*Code*

Fecha de Recepción: 2021-06-09

*Reception date*

Fecha de Calibración: 2021-06-11

*Calibration date*

Este registro de calibración proporciona evidencia documental para la trazabilidad a los patrones nacionales, llevados a cabo por las unidades de medición de acuerdo con el Sistema Internacional de Unidades. (SI).

*This calibration register provides documentary evidence for the traceability to national standards, carried out by the units of measurement according to the international System of Units (SI).*

### INAMHI

Es responsabilidad del usuario establecer la frecuencia de calibración de este instrumento. Esta declaración es aspecto auditable en el sistema de gestión en su empresa.

*The user shall be responsible for establishing the calibration frequency of this instrument. This statement is an auditable aspect in the management system of his company.*



Firmado electrónicamente por:

**CESAR DAVID  
TONATO  
PERALTA**

MSc. David Tonato

COORDINADOR DEL LABORATORIO

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL.

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THERE OF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

Dirección: Calle Núñez de Vela N36-15 y Corea, Quito - Ecuador.

Teléfonos: (593 - 2) 397 11 00 Ext. 87058. website: [www.inamhi.gob.ec](http://www.inamhi.gob.ec). correo: [labmet@inamhi.gob.ec](mailto:labmet@inamhi.gob.ec).

**PROCEDIMIENTO**

Calibración efectuada usando el método de comparación directa con nuestros patrones, según lo establecido en el procedimiento "LMET-PC-02 CALIBRACION DE SENSORES DE HUMEDAD RELATIVA AMBIENTE"

**CONDICIONES AMBIENTALES**

<b>TEMPERATURA AMBIENTE:</b>	(	21.6	±	0.5	)	°C
<b>HUMEDAD RELATIVA:</b>	(	48.6	±	3.4	)	%HR
<b>PRESIÓN ATMOSFÉRICA:</b>	(	730.6	±	1.4	)	hPa

**PATRONES**

THUNDER SCIENTIFIC 2500ST-LT HUMIDITY GENERADOR:	MARCA	THUNDER SCIENTIFIC				
	MODELO	2500				
	SERIAL/CÓDIGO	1310985				
	Nº CERTIFICADO	11374				
		Rango %hr	Incertidumbre		Correccion %hr	Resolucion (%HR) :
	10.00	±	0.07 %HR	-0.04	0.01	0.01
	20.00	±	0.12 %HR	-0.06	0.01	
	49.99	±	0.28 %HR	-0.08	0.01	
	80.06	±	0.42 %HR	-0.08	0.01	
	Temperatura Ambiente de Calibracion:					
						24 °C

**CARACTERIZACION DE LA CAMARA:**

MARCA	THUNDER SCIENTIFIC		
MODELO	2500		
SERIAL/CÓDIGO	1310985		
Nº CERTIFICADO	XXXXXX		
INCERTIDUMBRE:	XXXXXX		
Temperatura °C	Estabilidad	Uniformidad	
-10	± ( 0.0816	0.038 )	
20	± ( 0.0734	0.027 )	
60	± ( 0.0923	0.044 )	

**HIGRÓMETRO:**

MARCA	RH SYSTEMS	
MODELO	473	
SERIAL/CÓDIGO	13-0414	
Nº CERTIFICADO	11374	
FECHA DE CALIBRACIÓN	02/03/2021	
Lectura higrómetro	Corrección	Incertidumbre
10.72 %HR	-0.66 %HR	0.19 %HR
36.92 %HR	0.24 %HR	0.63 %HR
49.88 %HR	0.09 %HR	0.73 %HR
68.72 %HR	0.29 %HR	1.00 %HR
79.94 %HR	0.53 %HR	1.20 %HR
92.71 %HR	2.74 %HR	1.40 %HR

**LUGAR DONDE SE REALIZA LA CALIBRACIÓN:**

La calibración fué realizada en el laboratorio de Temperatura y Humedad, localizado en Calle Nuñez de Vela N36-15 y Corea, Quito - Ecuador.

Se establecen las diferencias entre la lectura "LI" del Instrumento a calibrar y la lectura "LP" del patrón a fin de obtener el error absoluto.

$$EA = LI - LP$$

**El resultado corregido del instrumento viene dado por:**

$$RC = LI + C, \text{ siendo } C = \text{Corrección} = -EA$$

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THERE OF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

Magnitud: Humedad Relativa
Especificaciones Técnicas:
RANGOS:
EXACTITUD:
Resolución del Instrumento

( 0.0 a 100.0 ) % ± ( 3.000 ) %HR 0.1 %HR

**PARAMETRO SIN AJUSTE**

Nominal	LI	LP	EA	± EMP	± U <sub>exp</sub> (K=2)	UNIDAD	GRÁFICO
20%	26.1	20.08	6.06	3.00	0.65	%HR	G-1
35%	38.5	35.35	3.13	3.00	0.65		
50%	51.9	50.17	1.72	3.00	1.01		
70%	69.9	70.34	-0.43	3.00	1.21		
85%	83.0	86.43	-3.41	3.00	1.41		
92%	90.2	95.02	-4.83	3.00	1.41		

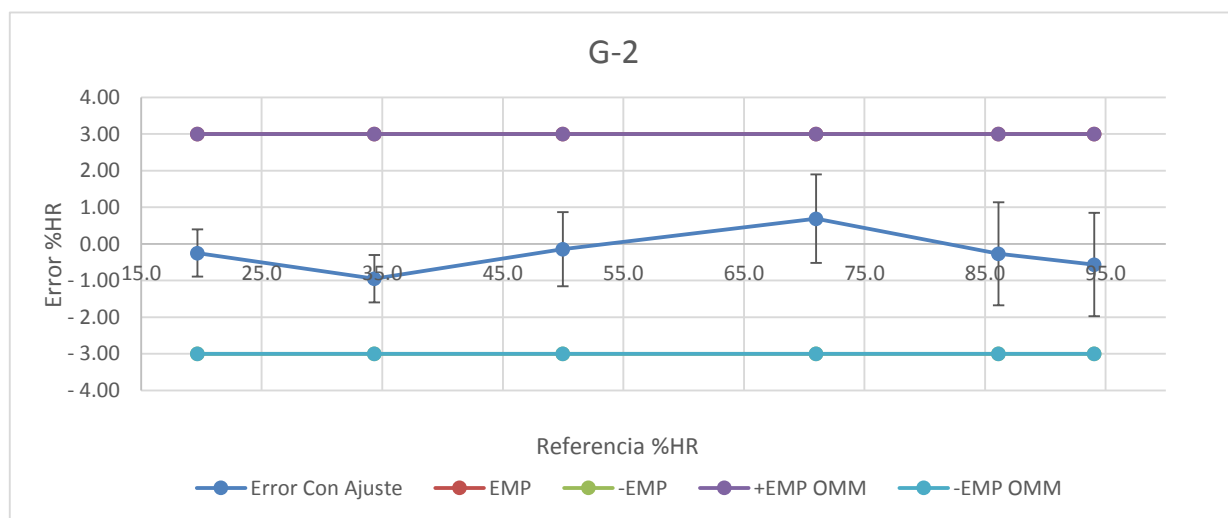
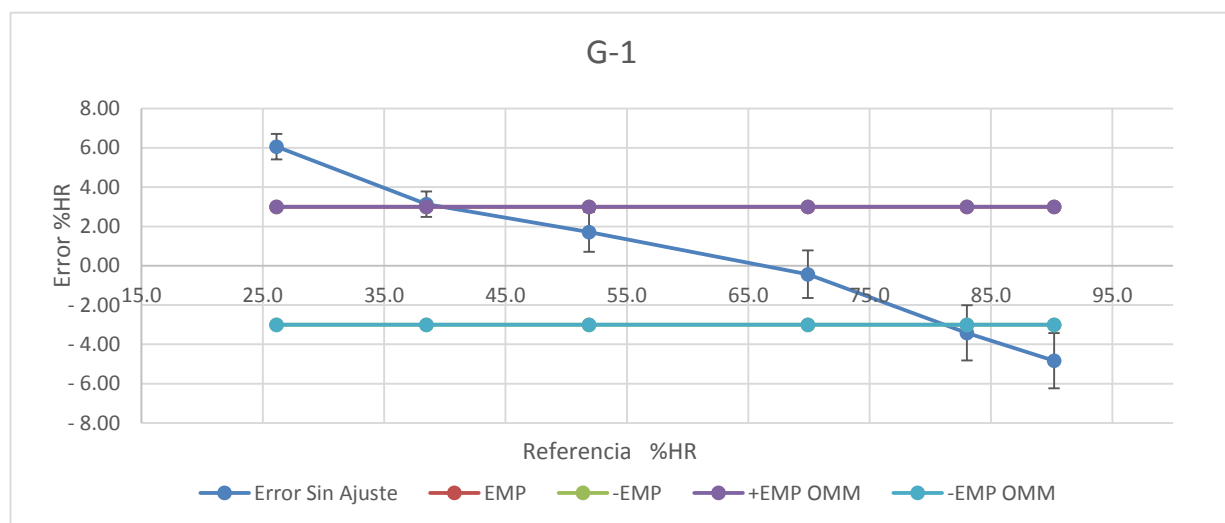
**PARAMETRO CON AJUSTE**

Nominal	LI	LP	EA	± EMP	± U <sub>exp</sub> (K=2)	UNIDAD	GRÁFICO
20%	19.7	19.92	-0.25	3.00	0.64	%HR	G-2
35%	34.3	35.29	-0.95	3.00	0.65		
50%	50.0	50.12	-0.14	3.00	1.01		
70%	71.0	70.30	0.69	3.00	1.21		
85%	86.1	86.40	-0.27	3.00	1.41		
92%	94.1	94.62	-0.56	3.00	1.41		

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL.

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THERE OF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

Dirección: Calle Núñez de Vela N36-15 y Corea, Quito - Ecuador.  
Teléfonos: (593 - 2) 397 11 00 Ext. 87058. website: [www.inamhi.gob.ec](http://www.inamhi.gob.ec). correo: [labmet@inamhi.gob.ec](mailto:labmet@inamhi.gob.ec).



ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THERE OF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

Dirección: Calle Núñez de Vela N36-15 y Corea, Quito - Ecuador.  
Teléfonos: (593 - 2) 397 11 00 Ext. 87058. website: [www.inamhi.gob.ec](http://www.inamhi.gob.ec). correo: [labmet@inamhi.gob.ec](mailto:labmet@inamhi.gob.ec).

**INCERTIDUMBRE:**

“La incertidumbre expandida reportada de la medición se establece como la incertidumbre de medición estándar multiplicada por el factor de cobertura  $k$  calculado, de tal manera que la probabilidad de cobertura corresponde a aproximadamente 95%”.

**OBSERVACIONES:**

El instrumento cumple con las especificaciones técnicas definidas por el fabricante y de la OMM con los coeficientes de ajuste realizado.

Reporte Sin Ajuste			
<i>Parametros antes del ajuste</i>			
Curva de ajuste original		Coeficientes antes del ajuste	
Curva Reportada	Por defecto	a0	0.00000
Modificación Realizada	No	a1	1.00000
Tipo de curva	Polinomio	a2	
Grado de Polinomio	1	a3	

Reporte de Ajuste			
<i>Parametros después del ajuste</i>			
Curva de ajuste original		Coeficientes después del ajuste	
Curva Reportada	Por defecto	a0	-9.97681534
Modificación Realizada	No	a1	1.16021050
Tipo de curva	Polinomio	a2	
Grado de Polinomio	1	a3	

Resultados de Ajuste y Recalibración	
Tolerancia maxima en %HR Especificaciones técnicas	3.00
Tolerancia maxima en %HR por la OMM	3.00
Como se encontró	Fuera de Tolerancia
Como se dejó	En Tolerancia
Ajuste realizado	Sí
Fecha sugerida por la OMM para la próxima Calibración	2022-06-12



Firmado electrónicamente por:  
**JIMMY SEBASTIAN  
 NARVAEZ ORDONEZ**

Fís. Jimmy Narváez  
 RESPONSABLE TÉCNICO



Firmado electrónicamente por:  
**SANTIAGO FERNANDO  
 RAMON ZAMBRANO**

Ing. Santiago Ramón  
 RESPONSABLE DE CALIDAD

LOS RESULTADOS SE REFIEREN ÚNICAMENTE AL INSTRUMENTO ANTERIORMENTE DESCRITO

Fin del registro de calibración.

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THERE OF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.





Número de Registro:

C - 2504 - PA

## REGISTRO DE CALIBRACIÓN DE PRESIÓN ATMOSFÉRICA

*Calibration Register*

### DATOS GENERALES

*General Information*

Usuario: Ricardo Llugsí  
*User*  
Dirección Fiscal: Legarda  
*Legal Address*  
Nº de Solicitud: S-014  
*Application Number*

### DATOS DEL INSTRUMENTO

*Instrument Information*

Objeto: Sensor de presión atmosférica  
*Object*  
Fabricante / Marca: BOSCH  
*Manufacturer / Brand*  
Modelo / Tipo: BMP-180 GY-68  
*Model / Type*  
Serial: 1244U252025  
*Serial number*  
Rango: (300 a 1100) hPa  
*Range*  
División/Resolución: 0.1 hPa  
*Division/Resolution*  
Código: OT-2504  
*Code*  
Fecha de Recepción: 2021-06-09  
*Reception date*  
Fecha de Calibración: 2021-06-18  
*Calibration date*

Este registro de calibración proporciona evidencia documental para la trazabilidad a los patrones nacionales, llevados a cabo por las unidades de medición de acuerdo con el Sistema Internacional de Unidades. (SI).

*This calibration register provides documentary evidence for the traceability to national standards, carried out by the units of measurement according to the international System of Units (SI).*

### INAMHI

Es responsabilidad del usuario establecer la frecuencia de calibración de este instrumento. Esta declaración es aspecto auditable en el sistema de gestión en su empresa.

*The user shall be responsible for establishing the calibration frequency of this instrument. This statement is an auditable aspect in the management system of his company.*



Firmado electrónicamente por:

**CESAR DAVID  
TONATO  
PERALTA**

MSc., David Tonato  
COORDINADOR DEL LABORATORIO

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL.

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

Dirección: Calle Núñez de Vela N36-15 y Corea, Quito - Ecuador.

Teléfonos: (593 - 2) 397 11 00 Ext. 87058. website: [www.inamhi.gob.ec](http://www.inamhi.gob.ec). correo: [labmet@inamhi.gob.ec](mailto:labmet@inamhi.gob.ec).

**PROCEDIMIENTO**

Calibración efectuada usando el método de comparación directa con nuestros patrones, según lo establecido en el procedimiento "LMET-PC-03 CALIBRACION DE SENSORES DE PRESION ABSOLUTA AMBIENTE "

**CONDICIONES AMBIENTALES**

<b>TEMPERATURA AMBIENTE:</b>	(	19.03	±	0.65	)	°C
<b>HUMEDAD RELATIVA:</b>	(	51.63	±	6.35	)	%HR
<b>PRESIÓN ATMOSFÉRICA:</b>	(	730.28	±	1.68	)	hPa

**PATRONES**

<b>CAMARA BAROMETRICA:</b>	MARCA	THEODOR FRIEDRICHS			
	MODELO	s/m			
	SERIAL/CÓDIGO	s/s			
	Nº CERTIFICADO	s/c			
	hPa		U	Estabilidad	Uniformidad
	500	± (	0.0816	0.038	0.079
	750	± (	0.0734	0.027	0.053
	1100	± (	0.0923	0.044	0.104

Temperatura Ambiente de Calibración: 24 °C

**MONITOR DE PRESION DE REFERENCIA:**

MARCA	FLUKE
MODELO	RPM4 A200K
SERIAL/CÓDIGO	2234
Nº CERTIFICADO	LFP-037-2021
FECHA CALIBRACIÓN:	2021-02-19

Rango hPa	Incertidumbre	Correccion hPa	Resolucion hPa:	Deriva hPa
600.00	± 8.6E-02 hPa	0.006	0.001	
650.00	± 8.6E-02 hPa	0.078	0.001	
700.00	± 8.6E-02 hPa	0.080	0.001	
750.00	± 8.6E-02 hPa	0.083	0.001	
800.00	± 8.6E-02 hPa	0.065	0.001	0.001
850.00	± 8.6E-02 hPa	0.056	0.001	
900.00	± 8.6E-02 hPa	0.053	0.001	
950.00	± 8.6E-02 hPa	0.054	0.001	
1000.00	± 8.6E-02 hPa	0.045	0.01	

Temperatura Ambiente de Calibración: 24 °C

**LUGAR DONDE SE REALIZA LA CALIBRACIÓN:**

La calibración fué realizada en el laboratorio de Temperature y Humedad, localizado en Calle Nuñez de Vela N36-15 y Corea, Quito - Ecuador.

Se establecen las diferencias entre la lectura "LI" del Instrumento a calibrar y la lectura "LP" del patrón a fin de obtener el error absoluto.

$$EA = LI - LP$$

**El resultado corregido del instrumento viene dado por:**

$$RC = LI + C, \text{ siendo } C = \text{Corrección} = -EA$$

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

**Especificaciones Técnicas:**
**RANGOS:**

( 300.0 a 1 100.0 ) hPa

**Magnitud: Presión Absoluta**
**EXACTITUD:**
 $\pm$  1.0 hPa

**Resolución del Instrumento**

0.1 hPa

**PARAMETRO SIN AJUSTE**

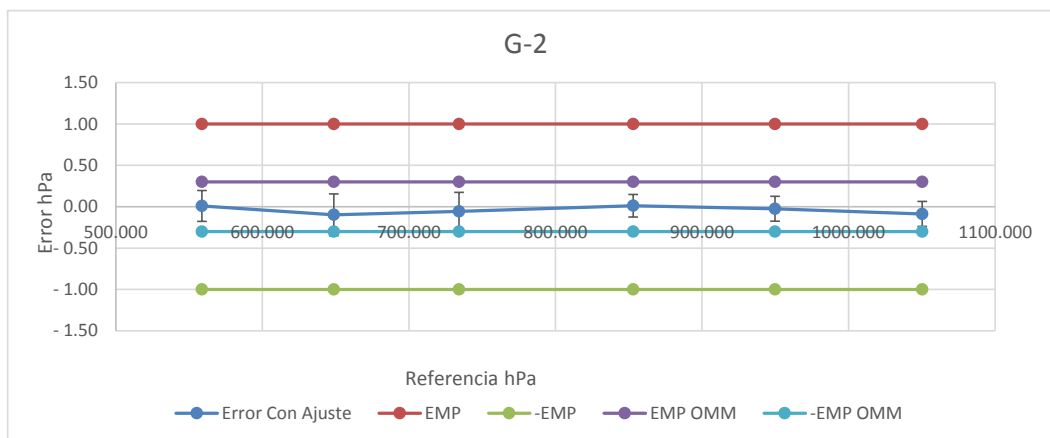
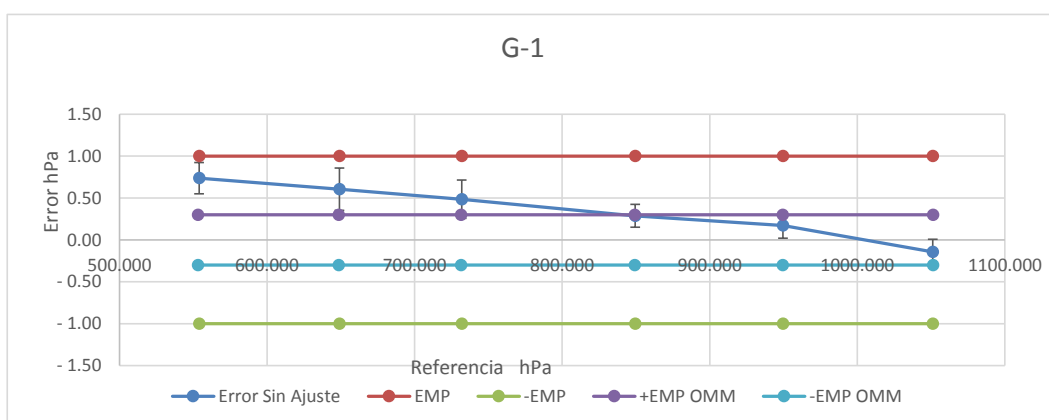
Nominal	LI	LP	EA	$\pm$ EMP	$\pm U_{exp} (K=2)$	UNIDAD	GRÁFICO
1050	1051.2	1051.2929	- 0.143	1.00	0.151	hPa	G-1
950	949.6	949.4691	0.171	1.00	0.151		
850	849.5	849.2018	0.288	1.00	0.136		
730	731.9	731.3939	0.486	1.00	0.229		
650	649.0	648.4245	0.606	1.00	0.253		
550	553.9	553.1629	0.737	1.00	0.187		

**PARAMETRO CON AJUSTE**

Nominal	LI	LP	EA	$\pm$ EMP	$\pm U_{exp} (K=2)$	UNIDAD	GRÁFICO
1050	1050.3	1050.3869	- 0.087	1.00	0.151	hPa	G-2
950	949.9	949.8745	- 0.025	1.00	0.151		
850	853.0	853.0081	0.012	1.00	0.136		
730	734.2	734.2650	- 0.055	1.00	0.229		
650	648.7	648.8080	- 0.098	1.00	0.253		
550	558.8	558.7806	0.009	1.00	0.187		

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.



ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL.

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.

Dirección: Calle Núñez de Vela N36-15 y Corea, Quito - Ecuador.  
Teléfonos: (593 - 2) 397 11 00 Ext. 87058. website: [www.inamhi.gob.ec](http://www.inamhi.gob.ec). correo: [labmet@inamhi.gob.ec](mailto:labmet@inamhi.gob.ec).

**INCERTIDUMBRE:**

“La incertidumbre expandida reportada de la medición se establece como la incertidumbre de medición estándar multiplicada por el factor de cobertura  $k$  calculado, de tal manera que la probabilidad de cobertura corresponde a aproximadamente 95%”.

**OBSERVACIONES:**

El instrumento cumple con las especificaciones definidas por el fabricante. Sin embargo, se le realizó el ajuste de tal manera que el error sea mínimo y cumpla con la normativa de la OMM, aunque la precisión del equipo no es la adecuada para uso meteorológico.

Reporte Sin Ajuste			
<i>Parametros antes del ajuste</i>			
Curva de ajuste original		Coeficientes antes del ajuste	
Curva Reportada	Por defecto	a0	0.000000
Modificación Realizada	No	a1	1.000000
Tipo de curva	Polinomio	a2	
Grado de Polinomio	1	a3	

<i>Parametros después del ajuste</i>			
Curva de ajuste original		Coeficientes después del ajuste	
Curva Reportada	Por defecto	a0	-1.7026344
Modificación Realizada	No	a1	1.0016867
Tipo de curva	Polinomio	a2	
Grado de Polinomio	1	a3	

Resultados de Ajuste y Recalibración	
Tolerancia maxima en hPa	1.00
Como se encontró	Cumple especificaciones técnicas
Como se dejó	En Tolerancia
Ajuste realizado	Sí
Fecha sugerida por la OMM para la próxima Calibración	2022-06-19



Firmado electrónicamente por:  
**JIMMY SEBASTIAN  
 NARVAEZ ORDONEZ**

Fís. Jimmy Narvárez  
 RESPONSABLE TÉCNICO



Firmado electrónicamente por:  
**SANTIAGO FERNANDO  
 RAMON ZAMBRANO**

Ing. Santiago Ramón  
 RESPONSABLE DE CALIDAD

LOS RESULTADOS SE REFIEREN ÚNICAMENTE AL INSTRUMENTO ANTERIORMENTE DESCRITO

Fin del registro de calibración.

ADVERTENCIA: EL PRESENTE REGISTRO NO CONSTITUYE AUTORIZACIÓN LEGAL DE SU USO PARA LA CERTIFICACIÓN METROLÓGICA A TERCEROS. LA REPRODUCCIÓN DEL DOCUMENTO DEBE SER TOTAL Y DEBE ESTAR AMPARADO POR AUTORIZACIÓN ESCRITA DE LABMET, VALIDO SÓLO EN ORIGINAL

WARNING: THIS REGISTER DOES NOT CONSTITUTE A LEGAL AUTHORIZATION FOR THE USE THEREOF FOR THE METROLOGICAL CERTIFICATION OF THIRD PARTIES. THIS CALIBRATION CERTIFICATE MAY NOT BE REPRODUCED OTHER THAN IN FULL, EXCEPT WITH THE WRITTEN APPROVAL OF LABMET, VALID IN ORIGINAL ONLY.