NKUA Weather Station Network Project

Short Range Weather Variable Forecasting Using LSTM Artificial Neural Network Architectures

Version 1

Ronaldo Tsela

(August 2023)

Abstract

Weather prediction plays a crucial role in various sectors such as agriculture, transportation, and disaster management. Short-range weather prediction, which involves forecasting atmospheric variables over a limited time span, is of particular significance for making informed decisions in these domains. Weather data, however, are complex with lots of interconnections and correlations between different factors and parameters. Extracting meaningful insights directly from these datasets in order to predict the future atmospheric behavior is really hard and for sure there is not a standard deterministic way of doing it.

This is where Machine Learning (ML) algorithms, particularly Artificial Neural Networks (ANN), come in as powerful tools for modeling such datasets and capturing their complex relationships. In this paper, we introduce the design and setup of a Long Short-Term Memory (LSTM) network customized for predicting short-term weather changes. Our method leverages data gathered from a ground-based weather station situated at the Department of Aerospace and Technology within the Euripus Complex on the island of Evvoia.

The proposed LSTM network is trained on these historical weather data gathered and is tailored to predict meteorological parameters such as temperature, humidity, pressure, and wind speed for a specific short time horizon ranging from few minutes to a few hours. The network architecture, the required data preprocessing techniques, and the training procedures are presented in this document.

Experimental results obtained in this paper underscore the efficacy of the LSTM-based approach in comprehending intricate relationships within weather data, ultimately leading to enhanced forecasting precision. This study, a part of the UOA Weather Station Network project, focuses primarily on designing and deploying a predictive algorithm specifically on the UOA WSN API and UI. Although this algorithm offers weather forecasts using data collected basically from the stations of the mentioned network, the design is versatile enough allowing it to be used for various meteorology-oriented applications, regardless of the specific datasets these applications employ.

Contents

List of Figures

List of Equations

Introduction

In recent years, the field of weather prediction has witnessed significant advancements driven by the integration of cutting-edge technologies and various data-driven methodologies. Among these, Long Short-Term Memory (LSTM) artificial neural networks have emerged as a promising tool for enhancing the accuracy of short-range weather forecasting. Short-range weather prediction, encompassing forecasts over a span of a few hours to a couple of days, holds pivotal importance in sectors ranging from agriculture and energy management to transportation and disaster preparedness. The ability to predict meteorological parameters accurately within this limited time horizon empowers decision-makers to make informed choices and mitigate potential risks benefiting all aspects of everyday life and scientific discoveries.

LSTM, a subtype of Recurrent Neural Network (RNN), exhibits a unique capability to capture temporal dependencies and intricate patterns within sequential data. This makes these specialized networks particularly well-suited for modeling the dynamic and interconnected nature of atmospheric processes. However leveraging the power of LSTMs in short-range weather prediction necessitates a comprehensive understanding of both meteorological principles for atmospheric behavior and machine learning techniques.

This paper explores the intricate relationship between atmospheric variables and the utilization of powerful machine learning algorithms such as the LSTM artificial neural network architecture. We delve into the essential preprocessing steps and data preparation techniques that cater to the specific characteristics of weather data utilized for the model, ensuring optimal performance of the LSTM network and we conclude this study with the possible potentials of this approach.

Meteorological Parameter Interactions and Correlations

Meteorological variables exhibit intricate and interconnected dependencies that play a pivotal role in shaping weather patterns and phenomena. Understanding these dependencies is crucial for accurate weather prediction and forecasting. Various atmospheric parameters influence and interact with each other, creating complex relationships that impact weather conditions even over short time horizons. Here, we explore some key meteorological variable dependencies between the temperature, relative humidity, pressure, precipitation, wind speed and direction.

Temperature and Humidity

Temperature and humidity share a close and intricate relationship that significantly influences weather patterns. As temperatures rise, the air's capacity to hold moisture increases, leading to higher humidity levels. This interaction is particularly evident in warm and tropical climates, where warmer temperatures often result in elevated humidity due to increased evaporation from various water bodies (lakes, oceans, rivers). The combination of high temperature and humidity can fuel the formation of thunderstorms and heavy rainfall, as the warm, moisture-laden air rises, cools, and condenses to form clouds and precipitation. Conversely, during cooler periods, lower temperatures can reduce the air's moisture-holding capacity, leading to lower humidity levels and potentially drier conditions.

Temperature and Wind Speed

Temperature and wind speed are correlated factors contributing to the dynamic movement of air masses. As temperature differences between regions increase, pressure gradients develop, initiating air movement from high-pressure to low-pressure areas. These pressure-driven wind patterns can lead to variations in wind speed. In warmer regions, temperature differentials might be more pronounced, resulting in stronger winds as air rushes to equalize pressure. Additionally, temperature contrasts along the boundary of air masses can spark weather systems, such as cyclones which further influence wind patterns and intensities.

Temperature and Pressure

Temperature and atmospheric pressure share an inverse relationship that shapes local and global weather phenomena. As air warms, it becomes less dense and rises, creating areas of low pressure. Conversely, cooler air is denser and sinks, leading to higher pressure regions. These pressure differences drive wind circulation and the movement of air masses. Understanding the intricate interplay between temperature and pressure is pivotal for predicting shifts in weather systems, the onset of monsoons, and the formation of high and low-pressure zones.

Temperature and Rainfall

The relationship between temperature and rainfall is multifaceted, influenced by factors such as humidity, atmospheric stability, and geographic features. While warmer temperatures can lead to increased evaporation rates and higher humidity levels, promoting the availability of moisture for precipitation, other factors also come into play. Temperature contrasts between air masses can trigger atmospheric instability, which contributes to the formation of convective clouds and intense rainfall. Moreover, the interaction between temperature and topography can lead to *orographic lifting*, where moist air is forced to ascend over mountains, cooling and condensing to produce significant rainfall on the windward side. The interaction between temperature and rainfall is a complex interplay that underscores the need for a comprehensive approach to weather prediction.

Humidity and Wind Speed

The connection between humidity and wind speed lies in their collaborative influence on weather systems. Higher humidity levels provide more moisture content to the atmosphere, which, when combined with favorable wind patterns, can lead to the intensification of storm systems. In tropical regions, warm, humid air converges and rises, creating low-pressure areas that draw in surrounding air. The Coriolis effect, influenced by the Earth's rotation, imparts rotation to these air masses, potentially leading to the development of tropical cyclones or hurricanes. Thus, the interaction between humidity and wind speed contributes to the strength and potential impact of weather events.

Humidity and Pressure

Humidity and atmospheric pressure exhibit a symbiotic relationship that influences air density and circulation. Moist air is less dense than dry air at the same temperature, contributing to lower pressure in humid conditions. As humid air rises due to heating or convergence, it cools and condenses, forming clouds and potentially leading to precipitation. This process can contribute to the development of low-pressure systems, which can influence wind patterns. The combination of humidity and pressure dynamics plays a significant role in shaping the movement of air masses and the formation of weather systems.

Humidity and Rainfall

Humidity and rainfall share a direct and fundamental interaction in the process of cloud formation and precipitation. High humidity levels provide the necessary moisture for water vapor to condense into liquid water or ice crystals within clouds. Consequently, humidity is a critical determinant of the potential for rainfall events.

Wind Speed and Wind Direction

Wind speed and wind direction are strongly linked, driven by the movement of air masses in response to pressure gradients and the Coriolis effect. Wind speed influences wind direction by determining the rate of movement of air. The Coriolis effect imparts a deflection in wind direction due to the Earth's rotation, leading to the development of different wind patterns such as. Wind speed and direction together shape atmospheric circulation, which plays a crucial role in distributing heat, moisture, and weather systems across the globe.

Wind Speed and Pressure
Wind speed and atmospheric pressure share an interdependent relationship in the context of pressure gradients. Pressure differences between regions create force imbalances that drive air movement, resulting in wind patterns. Higher pressure gradients, often associated with stronger pressure differences, can lead to increased wind speeds as air flows from areas of high pressure to low pressure. Additionally, wind speed can influence the development of localized pressure systems, especially in regions affected by strong winds associated with weather events like cyclones.

Wind Speed and Rainfall
Wind speed and rainfall have a significant interaction during weather events such as storms and cyclones. Higher wind speeds contribute to increased moisture transport, allowing for greater amounts of moisture-laden air to converge and rise, leading to enhanced condensation and cloud formation. In regions influenced by tropical cyclones, strong winds drive the circulation of air masses and promote the lifting of warm, moist air, leading to heavy rainfall. Wind speed also affects the horizontal distribution of precipitation, influencing the patterns of rainfall in storm systems.

Pressure and Rainfall
Pressure patterns exert a profound influence on the movement of air masses and the development of weather systems that impact rainfall. Low-pressure systems are often associated with converging air masses, leading to rising warm, moist air and the potential for cloud formation and precipitation. High-pressure systems, on the other hand, are often associated with sinking air, leading to clearer skies and reduced chances of precipitation. The interaction between pressure systems and rainfall is pivotal in understanding the larger-scale atmospheric dynamics that govern weather patterns.

The way meteorological variables interact is quite complex as we can see, and weather prediction considers even more factors than the ones we've discussed. This complexity makes short-range prediction a challenging task. Atmospheric behavior without a second thought is pretty predictable but it is also multivariant and highly dynamic meaning that traditional methods for data analysis and pattern extraction are not efficient or require high computational power for even a short time prediction. This is where Automated Computer Intelligent Data Processing, powered by advanced Machine Learning algorithms like LSTM networks that this paper is about, becomes invaluable. These technologies help us create detailed models that can capture the intricate relationships driving short-range weather forecasts without requiring extremely powerful computers and a lot of processing time.

LSTM Networks

LSTM networks are a class of Recurrent Neural Networks (RNN) capable of time-series analysis and prediction related problems described by high historical dependencies and correlation between data. This type of artificial neural network is known to be extremely precise in identifying intricate patterns in data sequences such as the weather data that we are analyzing here for short range prediction.

LSTM networks employ a new kind of special RNN neuron known as LSTM unit introduced originally by [1]. This new neural model addresses the problem of *long-term reliance* present in traditional RNNs according to that paper presents. Simple RNNs are applicable to a wide range of problems from speech recognition to language modeling and image captioning resulting in very precise and efficient operation but the specific feature of LSTM -its memory- makes it even better at certain tasks that require a long-term learning capacity. The long-term memory requirement in specific problems creates a long gap length on the simple RNN networks reducing their performance significantly. Therefore it is not a mistake to say that LSTM's are the extension of simple RNNs.

LSTM Structure

LSTMs are specifically developed to address the challenges that traditional RNNs face in handling long-range dependencies and vanishing/exploding gradient problems. The LSTM structure provides to this computational unit the ability to learn and maintain long-range dependencies, as well as its capacity to mitigate vanishing gradient issues, make it a powerful tool for capturing patterns in sequential data, such as the time-evolving characteristics of dynamic weather data for short-range prediction.

The basic structure of an LSTM unit consists of several components, each serving a distinct purpose in processing sequential data as depicted in Figure 1. These components are described below.

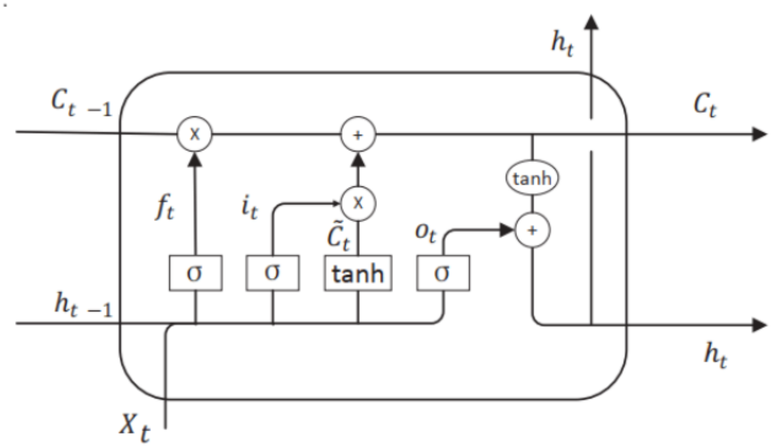


Figure 1 *– The LSTM unit*

Cell State (C_t): The cell state is the key component for the LSTM unit functionality. The cell state acts as a memory within the LSTM with the ability to add or remove information regulated by structures known as gates. The cell state traverses the entire unit from its input to its output as can be seen on top of the unit depicted in Figure. Gates in general are composed out of sigmoid neural net layer and a pointwise multiplication operation. The values generated from the sigmoid activation function ranges from 0 to 1 and refer to the percentage of data passed through that gate.

Hidden State (h_t): The hidden state is the output of the LSTM cell for a given time step. It contains distilled information that the LSTM has deemed relevant to the current prediction task based on the input and previous time step's hidden state.

Candidate Cell State (C_t^{-1}): This is the new candidate value that can be added to the cell state. It is calculated based on the current input and previous hidden state and is a candidate for addition to the cell state after considering the input gate.

Input Gate (i_t): The input gate determines how much new information should be added to the cell state. It considers the current input and previous hidden state and produces a value between 0 and 1 that represents the update to the cell state.

Forget Gate (f_t): The forget gate decides what information should be discarded from the cell state. It considers the current input and previous hidden state and determines the proportion of information to forget from the cell state, again producing values between 0 and 1.

Output Gate (o_t): The output gate determines the amount of information from the cell state that should be used to compute the hidden state thus the output. It considers the current input and previous hidden state and produces a value between 0 and 1.

LSTM Operation Procedure

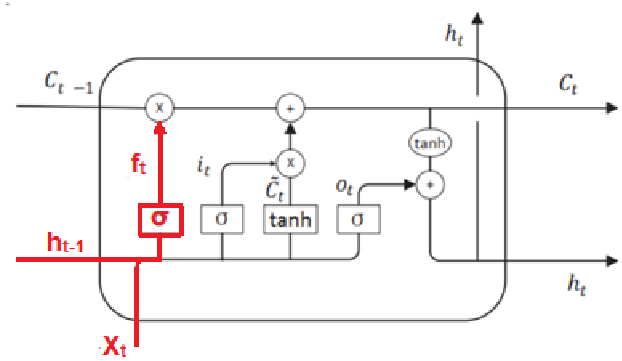


Figure 2 *– Forget state process*

The first step consists of deciding what information should be discarded from the cell state, considering the current input x_t and the previous hidden state h_{t-1}. This multipliable value represents the fraction of the information to be kept (1 complete, 0 get rid).

$$f_t = \sigma\left(w_f \cdot [h_{t-1}, x_t] + b_f\right) \qquad \text{Equation 1}$$

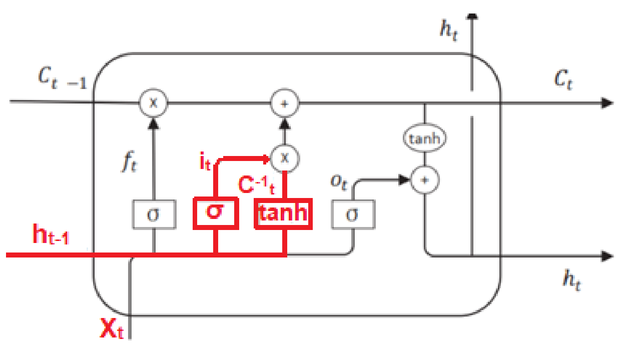


Figure 3 *– Candidate state process*

The next step decides what new information should be stored in the cell state. The input portion i_t for that specific timestep is defined, and the candidate value C_t^{-1} is set.

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \qquad \text{Equation 2}$$

$$C_t^{-1} = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \qquad \text{Equation 3}$$

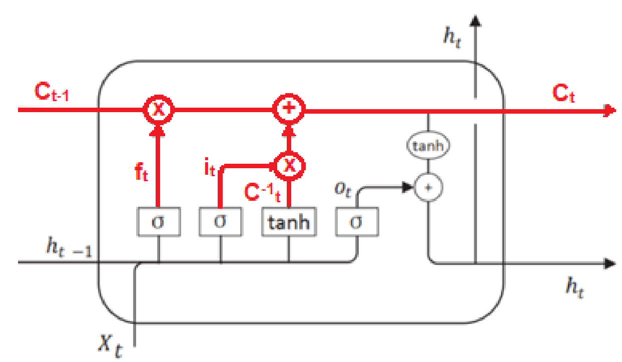


Figure 4 *– Update the current state*

Next the cell state is updated by dropping the portion f_t of the old information C_{t-1} and adding the portion i_t of the new candidate information C_t^{-1} one.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t^{-1} \qquad \text{Equation 4}$$

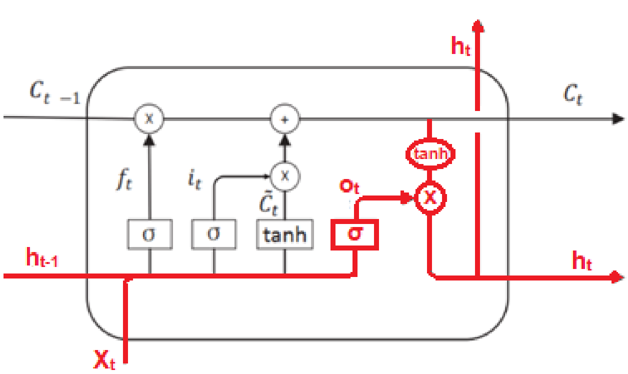


Figure 5 *– Output state*

Finally, the output is produced. The output is based on a filtered version of the cell state. A portion o_t of this value is used to distill the updated cell state C_t

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \qquad \text{Equation 5}$$

$$h_t = o_t \cdot \tanh(C_t) \qquad \text{Equation 6}$$

The set of equations provided above governs the behavior of the LSTM neuron. By employing this set of equations along with a training dataset, the weights w_j and biases b_j for $j = f, i, c, o$ are determined through the training process to solve the given problem. This involves the computation of loss, perplexity, and accuracy using forward propagation of data, followed by the adjustment of weights and biases through backward propagation of errors, employing the chain rule to calculate derivatives. For a more detailed guide on implementing a complete LSTM neural architecture from scratch, it is worth referring to the provided *link*.

The subsequent sections present the design, training process, and outcomes of an LSTM-based neural network architecture for temperature, humidity, pressure, and wind speed prediction. This endeavor utilizes the well-established Keras library, provided by TensorFlow for Python programming.

Methodology

Dataset Selection

To train and test the proposed neural network architecture for predicting future values of various meteorological parameters, we employ a dataset spanning one month. This dataset encompasses weather observations gathered from February 21, 2023, to March 21, 2023. These observations originate from a ground-based weather station situated within the Department of Aerospace Science and Technology at the Euripus Complex on Evvoia Island.

When developing prediction algorithms to model phenomena, it is crucial to acknowledge that incorporating multiple features which describe the specific phenomenon under investigation contributes to enhanced accuracy. However, for short to medium-range predictions, the aforementioned features prove sufficient to yield acceptable results.

Graphs depicting these recorded data are illustrated in Figure 6, Figure 7, Figure 8, and Figure 9. Notably, the data points are collected at 5-minute intervals.

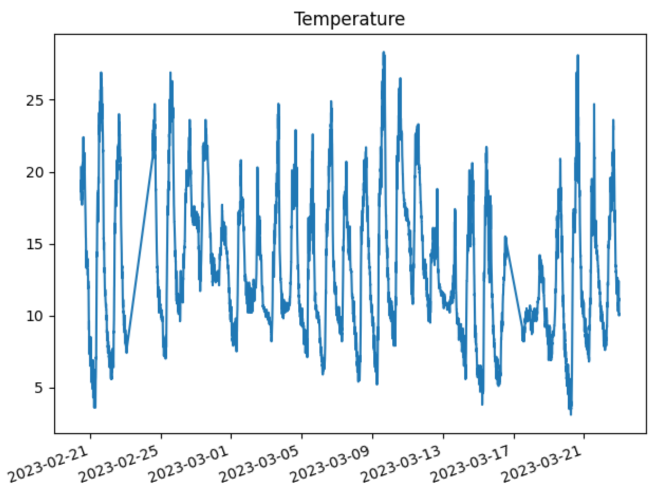


Figure 6 – Temperature recordings (Celsius)

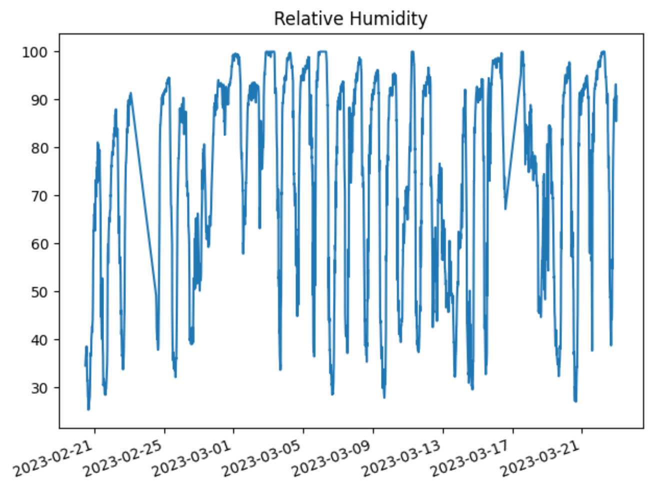


Figure 7 – Relative humidity (%)

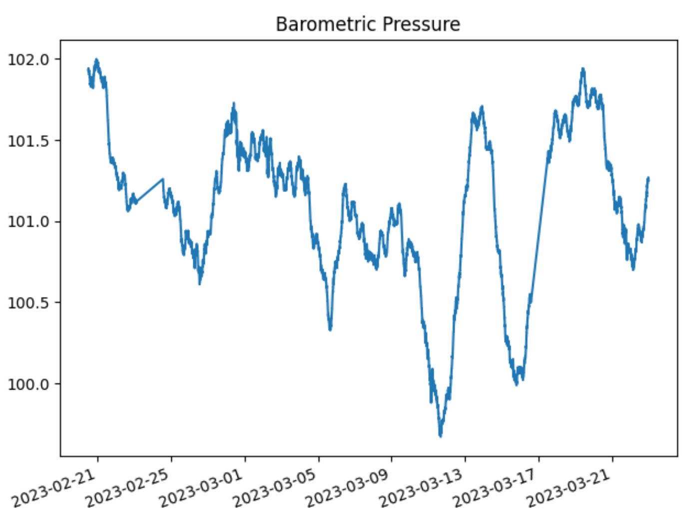


Figure 8 *– Barometric pressure recordings (kPa)*

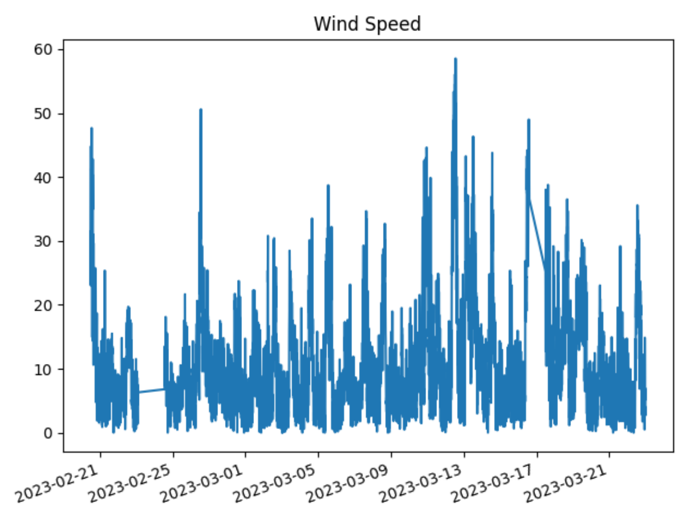


Figure 9 *– Wind speed recordings (m/s)*

Preprocessing

Normalization

Computers process data by interpreting it into simple numbers, which are stored as bits in memory. From a low-level perspective, certain operations such as division and multiplication are significantly impacted by the magnitude of numbers, influencing mainly execution speed.

Additionally, when working with specific models like the LSTM neural network proposed in this paper, which produces output values within the range of 0 to 1, data normalization becomes essential. The process of normalization limits data values within the 0 to 1 range. The most straightforward technique for normalizing data involves subtracting the mean value from each data sample and subsequently dividing the result by its standard deviation.

Let's consider a dataset with n samples denoted as $D = [d_1, d_2, \dots, d_n]$, where $d_i,\ i = 1,2, \dots n$ are the data samples. Then the normalized dataset D' can be constructed by each of the $d_i{}'$ that are produced by the following Equation 7.

$$d_i' = \frac{\left(d_i - \frac{\sum_j d_j}{n}\right)}{\frac{1}{n-1}\sqrt{\sum_k \left(d_k - \frac{\sum_j d_j}{n}\right)^2}} \quad \text{Equation 7}$$

Training and Testing Dataset

It is important to note that the dataset should serve a dual purpose: training the model and validating its performance. Consequently, the dataset is divided into two sets - the training dataset and the testing dataset - in a ratio of 80% for training and 20% for testing respectively.

Let's denote the cardinality of the normalized dataset as |D'|, and the training and validation datasets as D_{train} and D_{val}, respectively. Given a training size denoted by s, as defined by the Equation 8, the formation of datasets follows as described below.

$$s = 0.8 \cdot |D'| \quad \text{Equation 8}$$

$$D_{train} = D'(1, 2, \dots s) \quad \text{Equation 9}$$

$$D_{val} = D'(s+1, s+2, \dots n) \quad \text{Equation 10}$$

The training dataset comprises all features presented sequentially, along with their corresponding expected values. In particular, for a sequence of P historical data timesteps, the value following an additional F future timesteps encompasses the anticipated value-essentially, the value the model aims to approximate. In essence, the expected data is structured as follows:

$$X_k = D_k \quad \text{Equation 11}$$

$$Y_k = D_k(P + F + 1, P + F + 2, \dots n) \quad \text{Equation 12}$$

Where $k = train, val$ for the training and validation datasets respectively.

Data Reduction

Typically, datasets comprise a substantial volume of data recordings, with frequent sampling intervals. Specifically in this case, the dataset consists of recordings captured at 5-minute intervals. However, notable variations are not expected within one-hour intervals, as evident from the above figures. Consequently, it is possible to reduce the data load by employing a down-sampling approach. This not only improves memory management but also accelerates training while maintaining the desired performance.

For down-sampling, a timestep of T is employed. As a result, the construction of datasets for both training and validation takes the following form:

$$X_k = D_k(1, T+1, 1+2\cdot T, \dots) \qquad \text{Equation 13}$$
$$Y_k = D_k(P+F+1, P+F+2+T, \dots) \qquad \text{Equation 14}$$

Where k corresponds to either $train$ or val. Additionally, to further optimize training computations, batches of size 64 are generated, each with a sequence length of $P\,/\,T$.

LSTM Network Architecture

The architecture of the proposed predictor is very simple and consists of an input layer, a hidden layer of LSTM units and a dense layer before the single output generation. The input layer possesses a shape of $P/T \cdot m$, where m represents the count of features utilized for predictions. In the present context, these features encompass temperature, humidity, pressure, wind speed, and rainfall—a total of five attributes.

The LSTM layer comprises 64 LSTM units, and the resulting outputs are directed to a single Dense unit. Figure 10 provides a representation of the computational graph inherent in the proposed artificial neural network architecture, which is based on the LSTM unit.

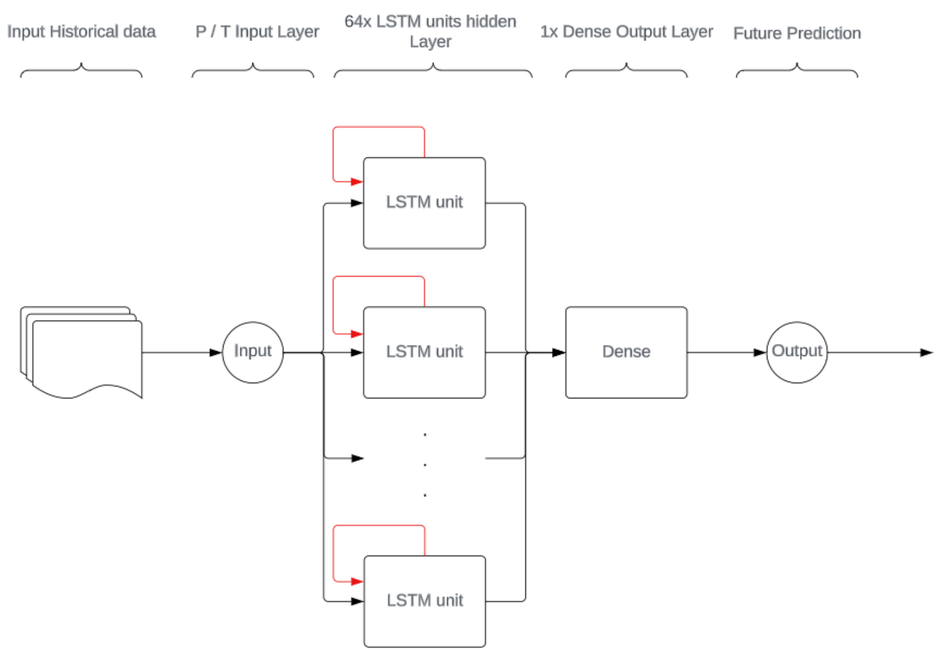


Figure 10 *– The computational graph of the proposed network for short range weather prediction based on ANN*

The architecture comprises a total of 18241 trainable parameters. To train these parameters, 15 epochs are executed using the training dataset, D_{train}, while the validation results are assessed using the D_{val} dataset. The training process employs the Adam optimization algorithm with a learning rate of 0.001. The loss history is tracked using the mean squared error (MSE) loss function.

Executing on a moderately powered personal computer, the training process on the mentioned dataset and for the mentioned training epochs concludes in under a minute.

Prediction

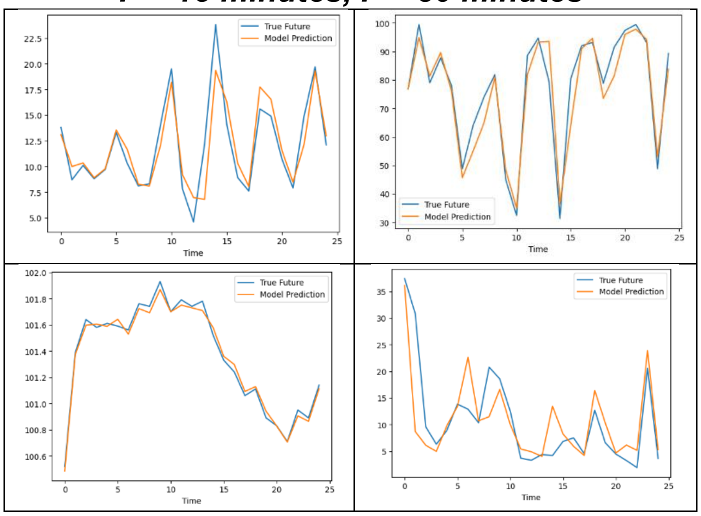
The proposed model involves utilizing a subset of a sequence containing P historical recordings. These data samples undergo a reduction at intervals of T through down sampling. Subsequently, they are normalized using the Equation 7 before being fed into the processing chain of an artificial neural network to generate predictions. The prediction for the future time step F is formulated according to the LSTM rule outlined in the preceding section. Additional weights are employed from both the input and output layers to form the prediction. To revert to the actual predicted value, the inverse operation of the normalization is executed, as described by the provided Equation 15.

$$y_{real} = y_{pred} \cdot \sigma + \mu \qquad \text{Equation 15}$$

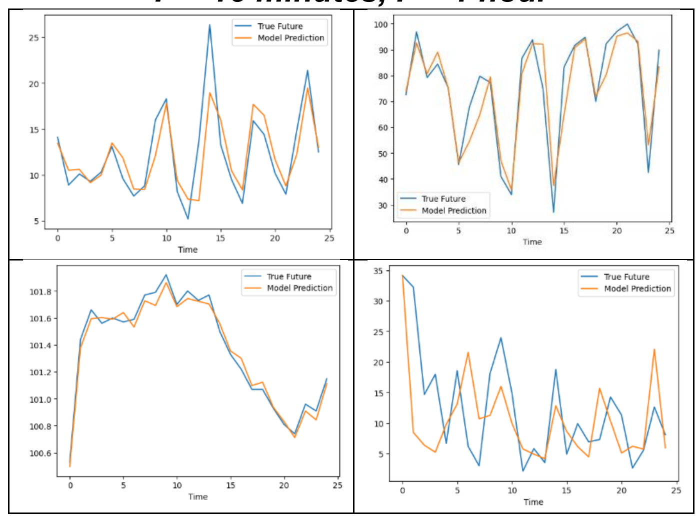
Results

Various combinations of the high-level functional parameters P and F are evaluated across the meteorological variables considered in the study, including temperature, humidity, pressure, and wind speed. This evaluation is conducted using the designated verification dataset. More specifically, predictions are made based on historical data spanning intervals of 15 minutes, 30 minutes, 1 hour, and 2 hours, with corresponding forecast horizons of 30 minutes, 1 hour, 2 hours, 4 hours, and 2 days for all the variables. The outcomes of this experimentation are visually presented in the following graphs. In the upper left corner is depicted the temperature, in the upper right corner the humidity, in the lower left corner the pressure and in the lower right corner the wind speed.

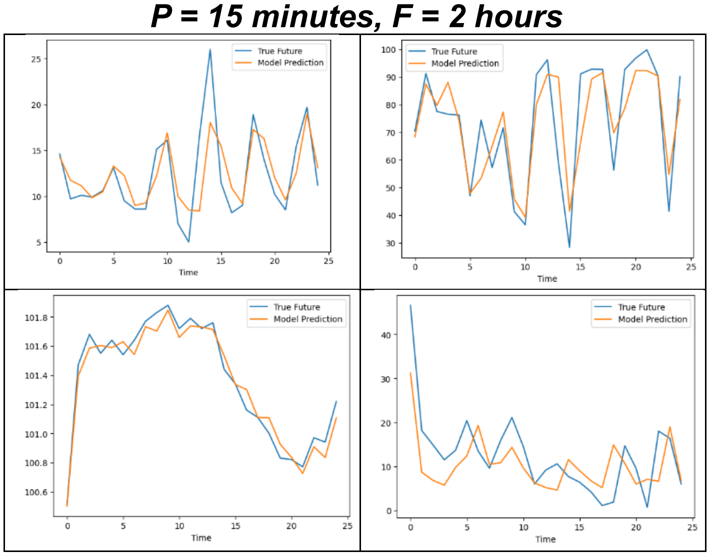
P = 15 minutes, F = 30 minutes



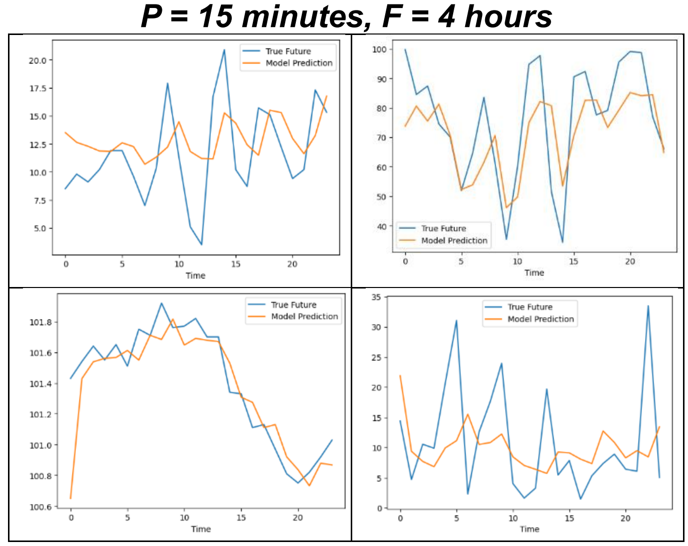
P = 15 minutes, F = 1 hour



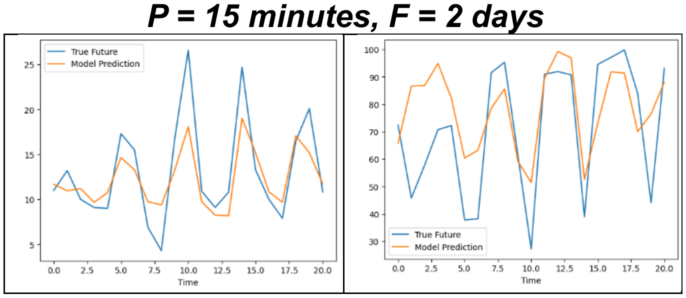
P = 15 minutes, F = 2 hours



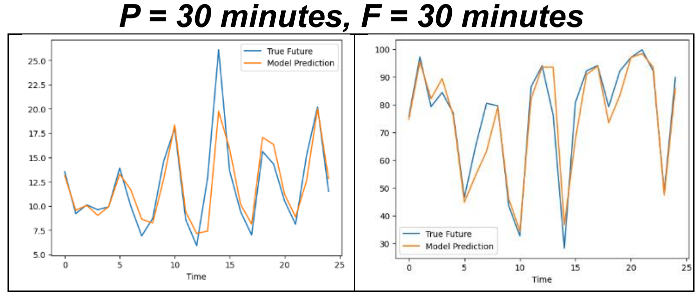
P = 15 minutes, F = 4 hours

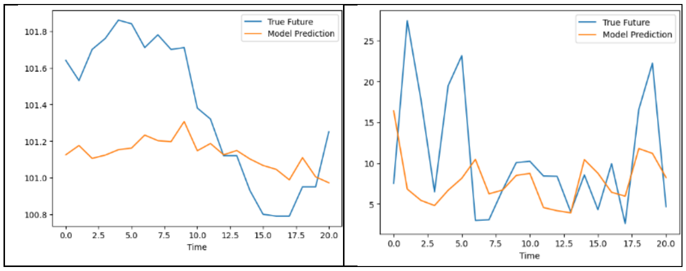


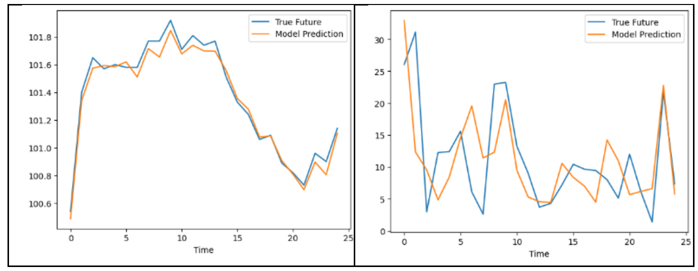
P = 15 minutes, F = 2 days



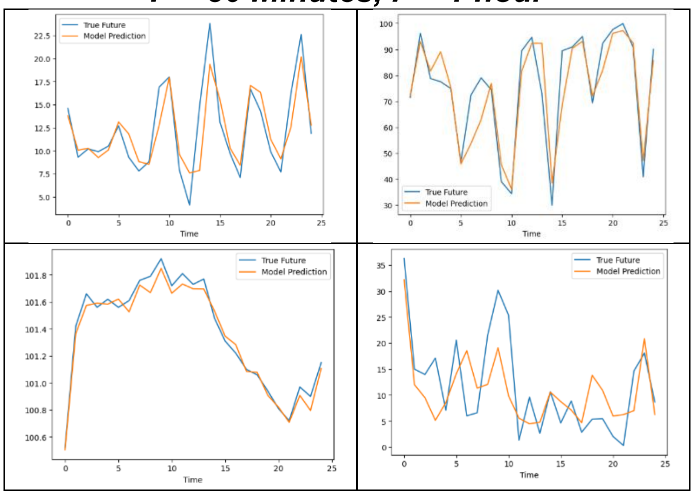
P = 30 minutes, F = 30 minutes



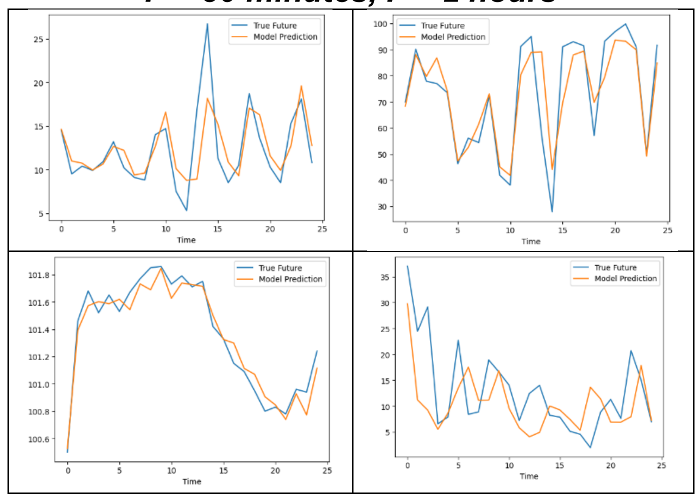




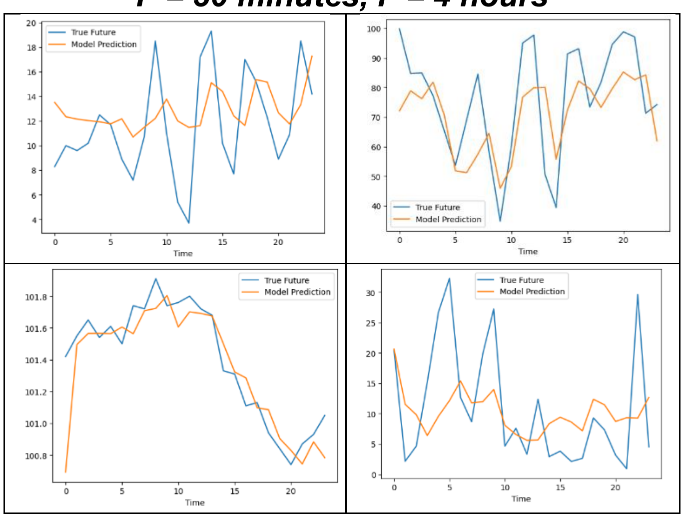
P = 30 minutes, F = 1 hour



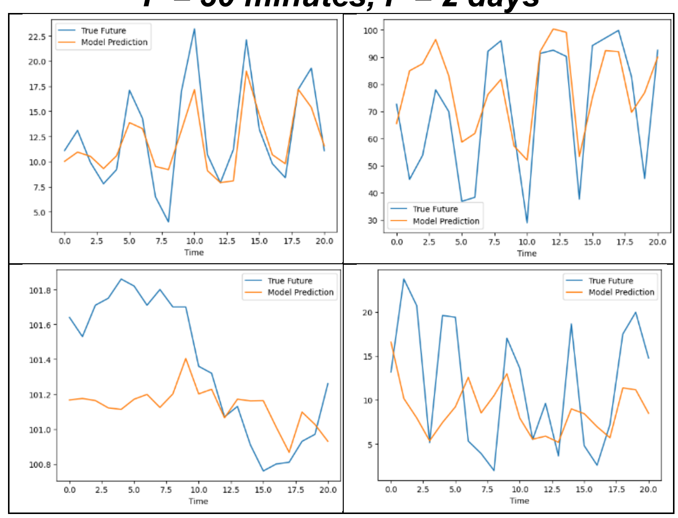
P = 30 minutes, F = 2 hours



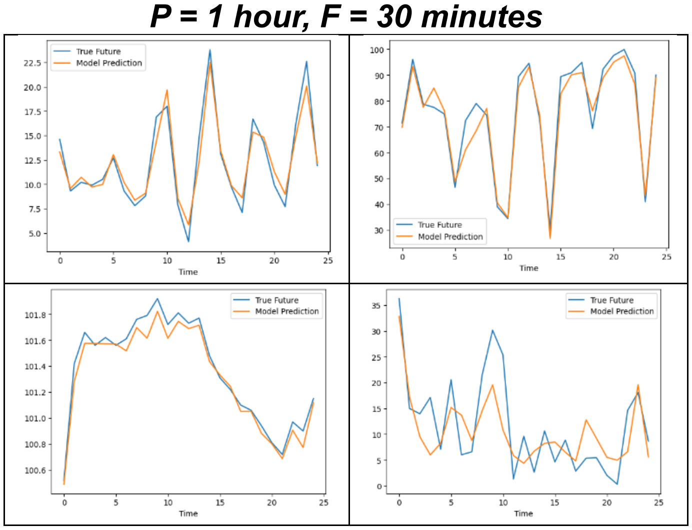
P = 30 minutes, F = 4 hours



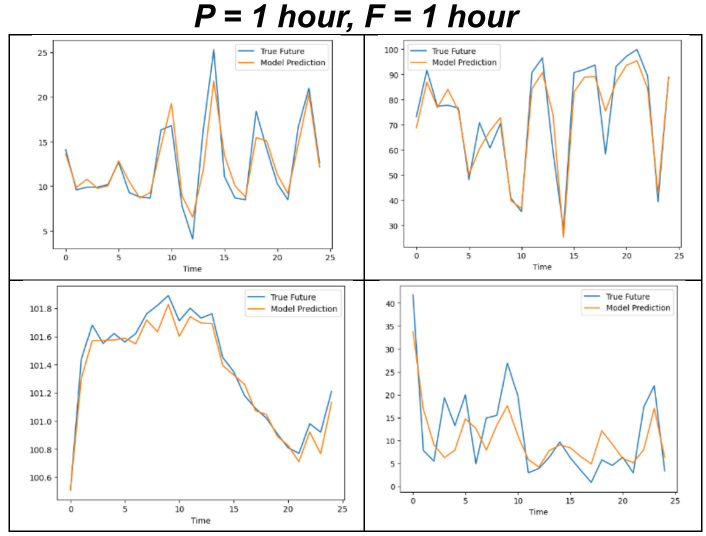
P = 30 minutes, F = 2 days



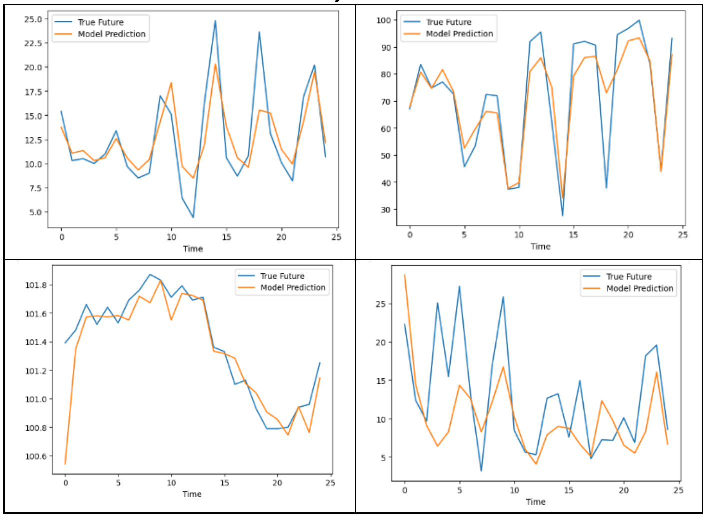
P = 1 hour, F = 30 minutes



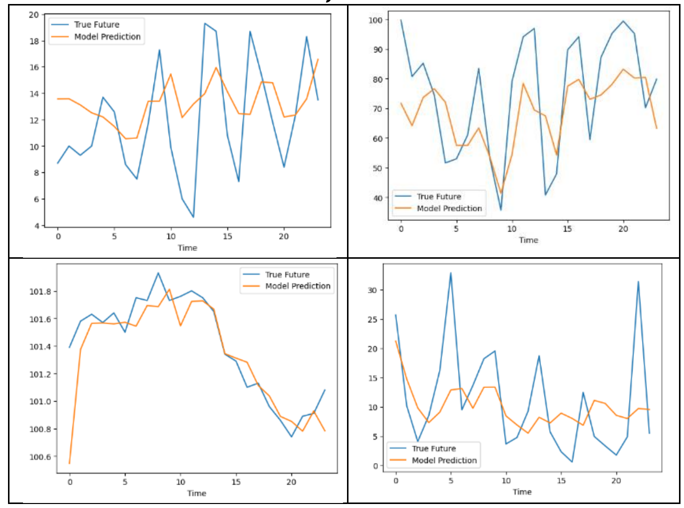
P = 1 hour, F = 1 hour



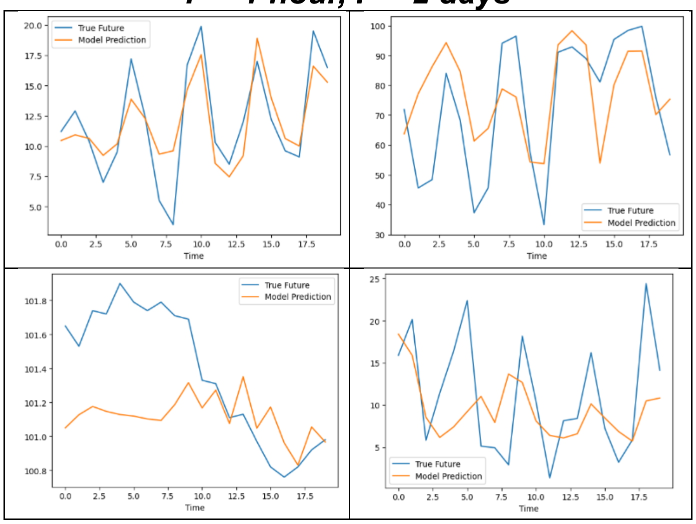
P = 1 hour, F = 2 hours



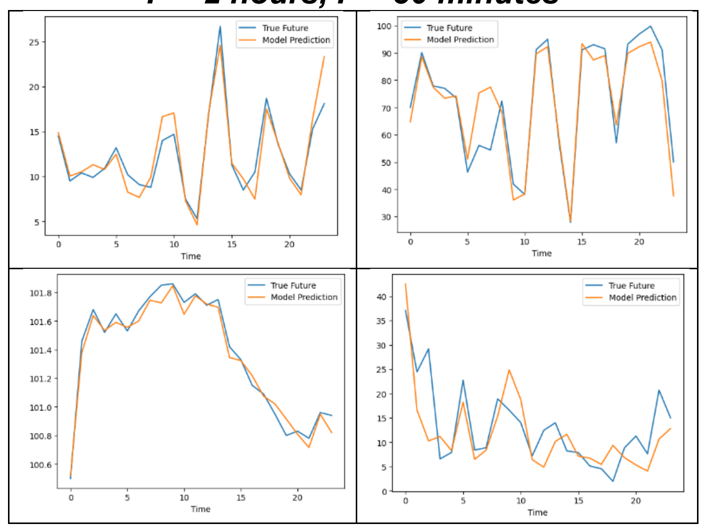
P = 1 hour, F = 4 hours



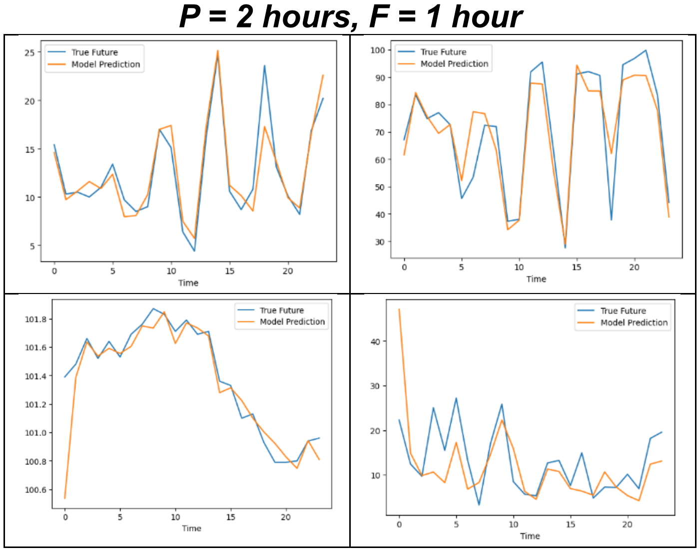
P = 1 hour, F = 2 days



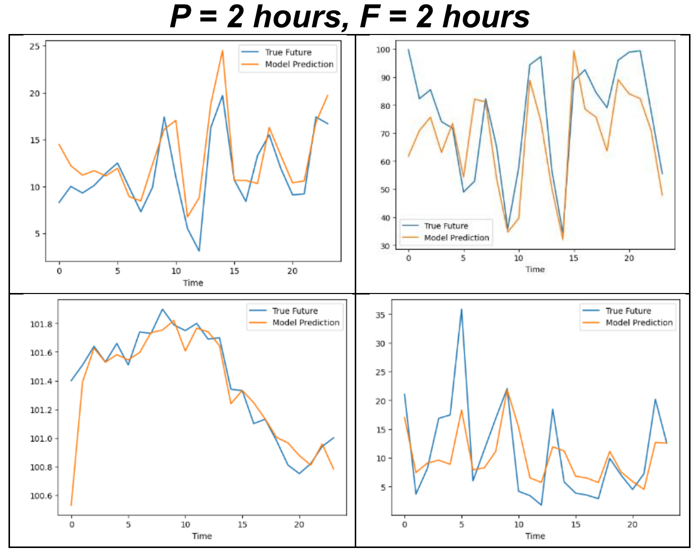
P = 2 hours, F = 30 minutes

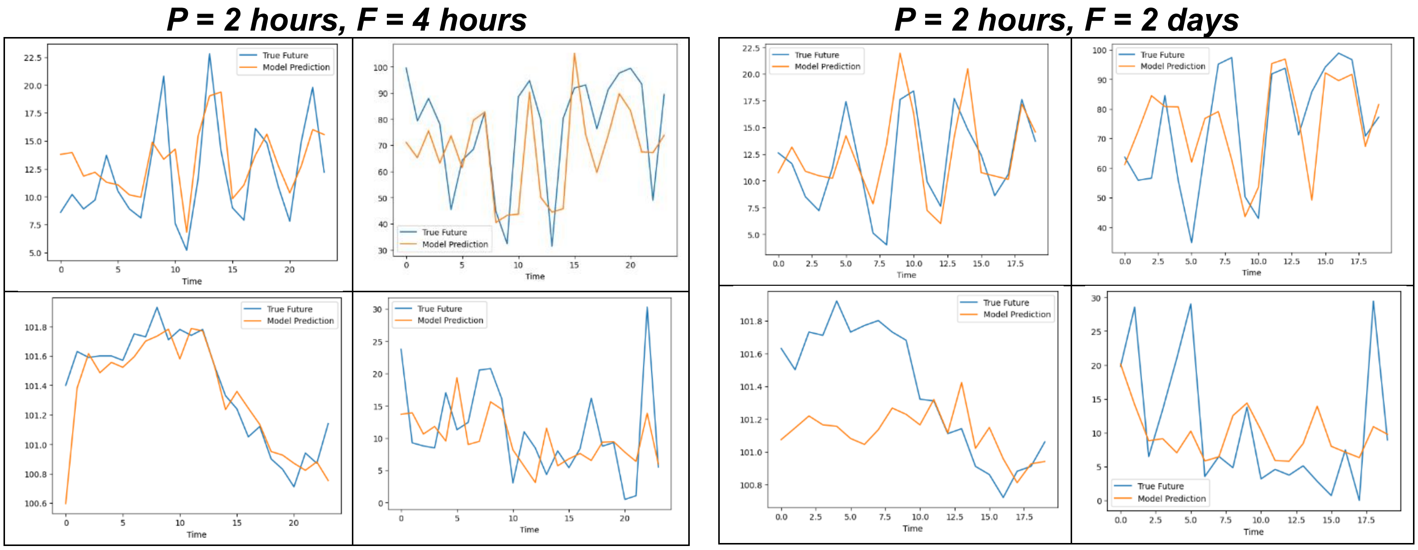


P = 2 hours, F = 1 hour



P = 2 hours, F = 2 hours





Conclusions

Given the successful and relatively accurate outcomes achieved in this implementation for short-range forecasting, spanning from a few minutes to a few hours, its applicability extends to a broad spectrum of scenarios. The primary goal of this study was to explore the viability of such algorithms when coupled with weather data from sources like the UOA Weather Station Network System. The proposed architecture is capable of providing reliable prediction on the short range of 30 minutes to 2 hours with historical recordings from 1 hour up to 2 hours.

An API will be developed to facilitate the utilization of this program. While the implementation is specifically tailored for the UOA WSN System, the API will be designed to be versatile and not tied to a specific data source. This approach empowers users to choose between using the provided dataset or opting for an alternative one.

Future research will focus on integrating satellite data into the training dataset of the algorithm. This integration aims to enhance the algorithm's predictive capabilities by incorporating more dynamic features, thus yielding even more precise predictions.

References

[1] Sepp Hochreiter, “Long Short-term Memory”, Dec. 1997
[2] Colah’s blog, “Understanding LSTM Networks”, Aug. 2015, *link*
[3] C. Donald Ahrens, “Essentials of Meteorology – An Invitation to the Atmosphere”, Third Edition
[4] Kera’s API *link*