Forecasting the Weather Conditions in Jordan Using Machine Learning Techniques

By

Laith Odah Bani Khalid

Supervisor

Dr. Gheith Ali Abandah, Prof.

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer and Networks Engineering

School of Graduate Studies

The University of Jordan

Aug, 2022

COMMITTEE DECISION

This thesis/dissertation (Forecasting the Weather Conditions in Jordan Using Machine Learning Techniques) was successfully defended and approved on ------------------------

Examination Committee	**Signature**
Dr. Gheith Abandah, (Supervisor) Prof. of Computer Engineering	------------------------
Dr. Iyad Jafar, (Member) Prof. of Computer Engineering	------------------------
Dr. Waleed Dweik, (Member) Assist. Prof. of Computer Engineering	------------------------
Dr. Ahmad AlJaafreh, (Member) Prof. of Computer Engineering (Tafila Technical University)	------------------------

AKNOWLEDGEMENT

I am extremely thankful to my thesis supervisor ~~prof~~Prof. Gheith Aban~~a~~dah for his support, supervision, and patience throughout my thesis work at the ~~university~~ University of Jordan. I would also like to express my gratitude to my parents for their support and encouragement throughout the study period.

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVIATION AND SYMBOLS

Abbreviation	Clarification
ANN	Artifical Neural Network
BiLSTM	Bidirectional Long Short-Term Memory
CC	Correlation Coefficient
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
ED-BiLSTM	Encoder-Decoder BILSTM
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
JMD	Jordanian Meteorological Department
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NSE	Average Nash-Sutcliffe Efficiency
NWP	Numerical Weather Prediction
PCC	Pearson Correlation Coefficient
R^2	Coefficient Of Determination
RMSE	Root Mean Square Error
RMSProp	Root Mean Squared Propagation
RNN	Recurrent Neural Network
Seq2Seq	Sequence to Sequence
SVM	Support Vector Machine
SVR	Support Vector Regression
TCN	Temporal Convolutional Networks
TL	Transfer Learning

Forecasting the Weather Conditions in Jordan Using Machine Learning Techniques

By

Laith Odah Bani Khalid

Supervisor

Dr. Gheith Ali Abandah, Prof.

ABSTRACT

Weather forecasting is an important research field due to its impact in a wide variety of applications like agriculture, industry, navigation, tourism, and everyday human activities. The traditional way of weather forecasting is based on complex physical models that describe the hydrodynamic behavior of the atmosphere. This way is costly, take much time, often inaccurate and requires supercomputers to make predictions. ~~In fact,~~ Using~~the deployment of~~ machine learning (ML) algorithms in weather forecasting ~~can~~ could increase the forecast accuracy and provide a complementary way to the traditional way of ~~the~~ weather forecasting. Recurrent neural networks (RNN) and bidirectional long short-term memory (BiLSTM) are ~~a~~ widely used for solving problems that have non-linear sequences and time series datasets.

In this thesis, we propose~~d~~ Encoder-Decoder BiLSTM (ED-BiLSTM) model to predict the weather conditions in Jordan over a short term. The evaluation results ~~exhibited~~ demonstrate that the proposed model gives very accurate results with low mean squared error (MSE) value, less than 1%. We ~~have~~ also ~~proved~~ show the importance of integrating the data~~set~~ of ~~the~~ nearby areas to the target area on the model performance, as the model performance has improved significantly

Formatted: Line spacing: Double

Formatted: Font: (Default) Times New Roman, Complex Script Font: Times New Roman, (Complex) Arabic (Jordan), English (United Kingdom)

Formatted: Font: (Default) Times New Roman, Complex Script Font: Times New Roman, (Complex) Arabic (Jordan), English (United Kingdom) when utilizing the data~~set~~ of all Jordan governorates and some areas near Jordan to predict the weather ~~state~~ condition in Amman.

The weather datasets of all Jordanian governorates and some areas near Jordan: Damascus, Cyprus, Jerusalem, and Cairo were utilized in building and assessing the proposed model. The weather parameters that were used in this thesis include hourly readings of average air temperature, ultraviolet index, atmospheric pressure, feels-like temperature, humidity, heat index and total precipitation from 2009 to 2021.

Formatted: Font: (Default) Times New Roman, Complex Script Font: Times New Roman, (Complex) Arabic (Jordan), English (United Kingdom)

Chapter 1: Introduction

This chapter includes a background of my thesis, research problem, thesis contributions, thesis importance, thesis objectives, thesis questions, thesis methodology, and thesis outline.

1.1 Background

Jordan is a country situated in the north of the Arabian Peninsula and in western Asia, it extends at a longitude from 29 to 33 degrees north and latitude from 36 to 19 degrees east. It has an estimated population of 10 million and an area of about 89,342 square kilometers. The land constitutes about 89% of the entire area, while the water constitutes about 0.7% (Arabeyyat *et al.,* 2018). Jordan is a semi-dry country, and it is considered one of the 10 poorest countries in the world in terms of water. In fact, the desert covers more than three-quarters of Jordan area, while the western part of the country contains arable land because it receives more rainfall over the winter season (Zahran *et al.,* 2015). In general, Jordan climate is a combination of the Mediterranean and desert climates, where the Mediterranean climate prevailing in the northern and western regions of the country, while the desert climate prevails in the rest of the country. The weather is hot, dry in summer and pleasant, humid in winter (Zahran *et al.,* 2015).

Commented [GA1]: Check. 89 +.7 does not equal 100%

Weather forecasting refers to the process of determining the atmospheric elements like air temperature, cloud cover, and precipitation over a certain ~~period of time~~period in the future (Rahayu *et al.,* 2020). Weather forecasting is done by gathering information about the current and recent past state of the atmosphere~~.~~. Then, utilizing several tools and techniques to study the gathered data and produce predictions based on it. In fact, accurate weather forecasting can help to reduce the losses that result from bad weather conditions such as floods, cyclone, and frost.

RNNs are a special form of artificial neural network (ANN) that are commonly applied in applications that have sequential and time series dataset like, text translation, audio recognition, and weather forecasting due to their ability to remember and utilize information from previous time steps. Long short-term memory (LSTM) and BiLSTM are modern versions of RNN that were designed to overcome shortcomings in traditional RNN. In this thesis, we proposed ED-BiLSTM model to predict the weather conditions in Jordan over a short-term.

1.2 Research Problem

Weather forecasting is a difficult process due to the constant and rapid changes in the state of the atmosphere over short periods of time. The accuracy of the weather predictions depends on several factors, such as the quantity and quality of available data, and the complexity of the weather patterns. In fact, the Jordanian Meteorological Department (JMD) and the Arab Weather Company are the two official weather forecasting entities in Jordan. These two entities use the traditional way of weather forecasting called numerical weather prediction (NWP) to predict the weather state in Jordan. In fact, predicting the weather conditions using NWP consists of three steps, firstly taking weather parameters readings about the present state of the atmosphere using land weather stations and satellite stations (Booz *et al.,* 2019). Secondly, the readings are entered to set of differential equations that describes the dynamic behavior of the atmosphere. Finally, the differential equations are solved by complex forecast models using supercomputers to produce predictions (Booz *et al.,* 2019). This way has several drawbacks like, take much time to make predictions, requires supercomputers to solve the equations, sometimes gives inaccurate results due to the difficulty of representing the chaotic nature of the atmosphere or incorrect initial measurements of the weather conditions (Doroshenko *et al.,* 2020). These reasons have motivated the researchers to look for new ways to improve the accuracy and reliability of predictions.

Deep learning (DL) is a robust data modelling tool that can be used to process huge amounts of datasets and learn the patterns and relationships between them in a way similar to that of the human brain (Rani *et al.,* 2020). In fact, DL techniques outperform the ML techniques in many applications like natural language processing, healthcare, and time series forecasting due to its high learning capabilities (Rani *et al.,* 2020). In this thesis, we will develop a DL model to predict the weather conditions in Jordan for a short period of time.

1.3 Research Contributions

In this thesis, we introduced ED-BiLSTM deep learning model (illustrated in Chapter 3) for the weather forecasting in Jordan over a short-term. In fact, this is the first work that predicts the weather conditions in all Jordanian governorates, but most of the previous works have tried to predict the weather conditions for certain regions or certain governorates in Jordan insufficiently. In addition, this is the first attempt that ~~takes into account~~considers the locations that around the targeted area during training the model. Also, none of the previous works designed a DL model to predict all the weather elements together. In fact, this is the first work that designs a DL model to predict all the weather elements together. The proposed model offers quick and accurate predictions with low cost and less computing resources.

1.4 Research Importance

The following four points highlight the importance of this thesis:

1. Providing a complementary way to the traditional way of the weather forecasting by utilizing DL techniques. And that will increase the accuracy of weather forecasting in Jordan.

2. This thesis will offer a valuable help for the researchers, especially those who interest in deploying DL techniques in the weather forecasting.
3. This thesis offers a detailed explanation of the weather patterns and behaviour in Jordan.

1.5 Thesis Objectives

The main purpose of this thesis is to design a DL model that can provide reliable and accurate predictions of the weather state in Jordan. Also, exploring and analyzing the weather ~~behaviour~~behavior in Jordan, as well as clarifying its major features, trends, and visualizing them.

Formatted: English (United States)

Formatted: English (United States)

1.6 Thesis Questions

1. Which ML model can be better for the weather forecasting in Jordan?
2. Is taking into consider the datasets of the locations surround the target can enhance the model accuracy or not?
3. What are the most effective weather parameters to ~~take into account~~consider when designing forecast models?

1.7 Thesis Methodology

The following steps represent our plan to achieve the objectives of this thesis:

1. Obtaining the dataset: getting and gathering the weather dataset from several sources.
2. ~~Surveying~~Surveying the previous studies: studying the prior research~~es~~ and papers that relate to our thesis topic, focusing on the latest ones, and summarize the techniques that have been used in those studies.
3. Checking the quality of dataset: applying several techniques and tools to assess the quality of the dataset, ensuring that it is suitable for DL, detecting and solving the problems, if any, such as noise, missing values, and anomalies values.

4. Visualizing the dataset: visualizing the dataset in several charts and figures that illustrate the weather patterns and trends in Jordan.
5. Designing forecast model: build a DL model by incorporating ED architecture with BiLSTM cell to predict the weather conditions in Jordan.
6. Preparing the dataset for training: making several procedures on dataset to make it ready for training, like normalization and constructing sliding windows.
7. Training and testing the proposed model: training the proposed model on the whole dataset and getting the results. Then, assessing them using the MSE metric.
8. Documenting our work: Writing and presenting our work and results in an easy and understandable way.

1.8 Thesis Outline

The rest of this thesis is organized in the following way: Chapter 2 contains some of the previous relevant works to our thesis topic. Chapter 3 explains our proposed forecasting model and some of the main concepts in ML in details. Chapter 4 includes dataset analysis, ~~description~~description, and the steps of preparing the dataset for training. Chapter 5 contains the experiments of enhancing the base model and the ~~final results~~results. Finally, Chapter 6 summarizes the conclusions and some of the future ideas to improve the model performance.

Chapter 2: Related Work

In this chapter, we will briefly explain some of the prior studies that deal with deploying ML techniques in the weather forecasting. This chapter consists of three sections, the first one includes papers that are closely related to our proposed model. The second section includes some research~~es~~ about the weather forecasting in Jordan using ML techniques. Whereas the last section contains papers about the weather forecasting outside Jordan using ML techniques.

2.1 Targeted Research Work

In this section, we will concentrate on four research papers which my research depends on. In fact, the proposed model consists of merging several ideas from these four papers.

Hossain *et al.* (2015) suggested two different DL models: stacked denoising auto-encoders (SDAE) and traditional ANN for air temperature forecasting at a certain future hour in Northwestern Nevada in the USA. Authors implemented two different approaches for training the models, the first one is univariate which means taking a series of past temperature values to predict a set of future values of air temperature, while the second one is multivariate which means taking a set of past values of multiple weather features to predict a range of future values of air temperature. The dataset features include: air temperature, wind speed, humidity and atmospheric pressure between November 2013 and December 2014. The results showed that the SDAE model achieved root mean square error (RMSE) by 3.81~~-~~% for univariate, and 2.06% for multivariate in 10-fold cross validation and 3.32%, 1.38% in one-fold cross validation. Whereas the traditional ANN model achieved RMSE by 6.36% and 5.08% for univariate and multivariate respectively in 10-fold cross validation and 5.23% and~~,~~ 4.19% in one-fold cross validation. Therefore, the multivariate approach achieves better prediction accuracy than univariate approach.

Zaytar and Amrani (2016) suggested ED-LSTM model to predict three weather elements: air temperature, wind speed and humidity in nine Moroccan cities over the next 24 and 72 hours. The suggested model composed of two LSTM layers with repeat vector and time distributed dense layers. The used dataset in training and testing the proposed model includes hourly readings of air temperature, wind speed and humidity between 2000 and 2015. The suggested model achieved MSE values between 0.00516 and 0.00839 for the next 24 hours, and between 0.00675 and 0.01053 for the next 72 hours.

Jaseena and Kovoor (2020) suggested a stacked autoencoder LSTM (SAE–LSTM) model to predict the wind speed at Tamil Nadu state in India. The used dataset in this study includes every 10 minutes readings of wind speed between December 2013 and December 2017 at Tamil Nadu. Also, the authors made a comparison between the suggested model and two ML models, support vector regression (SVR) and ANN, and two DL models, RNN and LSTM. The results exhibited that the suggested SAE-LSTM model outperformed the previous mentioned models. Also, the suggested model achieved mean absolute error (MAE), RMSE, and coefficient of determination (R^2) by 0.3982, 0.5969, and 96.20, respectively.

Hewage *et al.* (2021) suggested two different DL models: LSTM and temporal convolutional networks (TCN) to predict 10 weather elements over the near future. The suggested two models were implemented in two different approaches: multi-input multi-output (MIMO) and multi-input single-output (MISO). Also, the authors investigated the effectiveness of BiLSTM and transfer learning (TL) in making predictions. The used weather dataset in this study includes the following features: air temperature, surface pressure, X and Y components of wind directions, humidity, convective and non-convective rain, snow water equivalent, soil temperature, and soil moisture. The results revealed that the LSTM model outperformed the TCN model, and the MISO approach achieved better performance than MIMO approach. In addition, the BiLSTM model achieved better results than the other two suggested models and TL.

2.2 Weather Forecasting in Jordan Using Machine Learning Techniques

Zahran *et al.* (2015) suggested feed forward ANN based on backpropagation (BP) model to forecast the future rainfalls in Jordan. The datasets that were used in this study include rainfall readings for four regions in Jordan: Amman, Irbid, Ruwaished and AL-Ghour between 1977 and 2008.The suggested model consists of three layers: input layer, hidden layer, and output layer. The results showed that the suggested model achieved prediction accuracy by 85.63%, 86.26%, 77.73%, and 79 % for Amman, Irbid, AL-Ghour, and Ruwaished, respectively.

Odeh (2016) suggested feed forward network (FFN) for the minimum air temperature forecasting for four months: January, April, July, and October in Amman. The author used daily readings of minimum temperature for four months from 1973 to 2003 in building and testing the proposed model. The suggested model composed of input and output layers with three hidden layers between them. The results revealed that the FFN achieved R^2 around 0.99 and MSE ~~by~~ of 1%.

Arabiyyat *et al.* (2018) suggested a combined model between ANN and nonlinear autoregressive exogenous (NARX) to predict the precipitation patterns in Jordan for the next decade. The meteorological dataset that was used in this study includes annual precipitation readings for 26 stations in Jordan between 1985 and 2015. The suggested model composed of input and output layers, with eight hidden layers between them. The results revealed that the suggested model achieved MSE by 0.0038 and R^2 by 0.95041 for Amman airport station. The results also showed that a decrease in total rainfall is expected over the next decade, particularly in the regions that usually receive large amounts of rainfall. While, in the southern part of the country the amounts of rainfall are expected to remain constant or possibly increase. On the other hand, the rainfall in the eastern part of the country will be more than usual.

Dahamsheh and Suleiman (2020) suggested four ANN models to predict the total monthly rainfall at four areas in Jordan valley: Baqura, Deir Alla, Ghor Al-Safi and Wadi Al-Rayyan for long periods of time. Each model was used to predict the total monthly rainfall for each mentioned region. The used datasets in training and testing the proposed models include monthly rainfall reading from 1975 to 2005 and they obtained from JMD. The suggested models achieved MAE ~~by~~ of 7.5, 5.4, 4.6, and 5.7 and R^2 ~~by~~ of 0.98, 0.97, 0.66, and 0.97 for Baqura, Deir Alla, Ghor Al-Safi, and Wadi Al-Rayyan, respectively.

2.3 Weather Forecasting Outside Jordan Using Machine Learning Techniques

Anjali *et al.* (2019) proposed three ML models: multiple linear regression (MLR), support vector machine (SVM) and ANN for the air temperature forecasting in Kerala state in India. The weather dataset includes daily readings of the following features: air temperature, humidity, wind speed, wind direction and atmospheric pressure between 2007 and 2015. The results revealed that the MLP achieved mean error (ME), MAE, RMSE, average error (AE) and correlation coefficient (CC) by 0.0003, 1.3777, 1.8568, 1.0782, and 0.8119, respectively. While ANN achieved ME, MAE, RMSE, AE, and CC ~~by~~ of 0.2558, 1.5174, 2.0100, 1.2958, and 0.7932, consecutively. Finally, SVM achieved ME, MAE, RMSE, AE, and CC by 0.1720, 1.3667, 1.8727, 1.1371, and 0.8110, respectively. Hence, based on the error values we can say that the MLR model gives more accurate results than the other two models, ANN and SVM.

Chhetri *et al.* (2020) proposed a combined model of BiLSTM and gated recurrent unit (GRU) (BiLSTM-GRU) to predict the monthly rainfall in the Simtokha district in Bhutan. The proposed model composed of seven layers: pass-through input layer, one BiLSTM layer, one batch normalization layer, one GRU layer, one batch normalization layer, one dense layer, and an output layer. The climate dataset that was used in this study includes daily samples of maximum temperature, sunshine, minimum temperature, rainfall, wind speed and relative humidity from 1997 to 2017. Also, the authors compared the proposed model with other six models: linear regression, LSTM, GRU, BiLSTM, multi-layer perceptron (MLP), and convolutional neural network (CNN) to show the effectiveness of the proposed model. The results showed that the proposed model gave better results than other six models. The proposed model achieved MSE, RMSE, R^2, and Pearson correlation coefficient (PCC) value ~~by~~ of 0.0075, 0.087, 0.870, and 0.938, respectively.

Khan and Maity (2020) proposed a DL model combines between a one-dimensional convolutional neural network (Conv1D) and MLP (Conv1D+MLP) to predict the amount of precipitation from one to five days ahead in the Maharashtra state in India. The used dataset in this study includes: maximum air temperature, maximum relative humidity, geopotential height, longwave radiation, minimum relative humidity, u-wind speed, v-wind speed, sea level pressure, and rainfall between 1941 and 2005.The authors made a comparison between the proposed model with the other two models, MLP and SVR. The results revealed that the proposed model achieved better performance than the other two models, MLP and SVR. For one day ahead prediction, Conv1D-MLP achieved a range of R^2 values, average Nash-Sutcliffe efficiency (NSE) and average RMSE from 0.41 to 0.76, from 0.17 to 0.57, and from 5.81 to 17.45, respectively. Whereas, during the next five days prediction, the proposed model achieved a range of R^2, NSE, and average RMSE from 0.25 to 0.55, from 0.28 to 0.11, and from 6.6 to 24.19, respectively.

Rahayu *et al.* (2020) proposed RNN with LSTM model to predict the air temperature with five classifications: cold, cool, normal, warm, and hot over the next three days in Bandung city in Indonesia. The used dataset in this study includes daily readings of air temperature, humidity, rainfall and wind speed between 2000 and 2019. The results of this study include several conclusions: firstly, using adaptive moment estimation (Adam) optimization gives better accuracy of 80.36% than stochastic gradient descent (SGD) of 76.48%. Secondly, training the model on large dataset gives better prediction accuracy with lower error rate than training it on small datasets. Finally, dividing the dataset into 80:20 training: testing gives better accuracy than dividing it into 60:40 and 70:30.

Roy (2020) proposed three types of DL techniques: ~~:~~ MLP, LSTM, and a hybrid model of CNN and LSTM (CNN+LSTM) to predict the air temperature in New York state in the USA over the following day and ten days. The meteorological dataset that was used in this study includes daily readings of the following features: wind speed, snow depth, average temperature, snowfall, maximum temperature, ~~precipitation~~precipitation, and minimum temperature between 2009 and 2019. The results revealed that the CNN+LSTM model achieved higher accuracy in forecasting than the other two models. Over the one day ahead, the predictions accuracy of CNN+LSTM, LSTM and MLP are 97.42%, 95.03%, and 89.15%, consecutively. Whereas, over the ten days ahead the predictions accuracy of CNN+LSTM, LSTM, and MLP are 71.58%, 70.32%, and 63.23%, consecutively.

Chapter 3: Theoretical Background of The Proposed Model

3.1 Introduction

In this chapter, we will provide a theory explanation of the proposed model. Firstly, we will give a brief overview of RNN, LSTM, BiLSTM, and deep neural network (DNN). After that, we will explain ED--BiLSTM architecture in detail. Finally, we will discuss the evaluation metric that is used to assess the model effectiveness.

3.2 Basic RNN

Recurrent neural network (RNN) is a powerful kind of neural network that was developed to deal with sequences of data like speech recognition and time series applications (Schmidt, 2019). In fact, RNN contains feedback loops where the RNN inputs at each time step include the information of the current and previous time steps (Karevan and Suykens, 2020). Figure 1 shows simple RNN architecture. Figure 2 illustrates the basic structure of RNN cell.

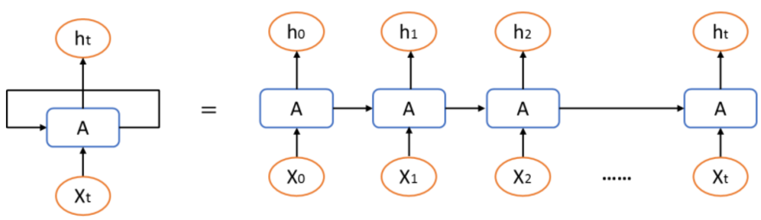

Figure 1. Simple RNN architecture.

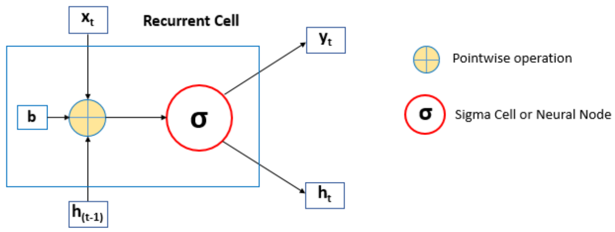

Figure 2. Basic structure of RNN cell.

The mathematical calculations of output are in the following two equations:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b) \quad (1)$$

$$y_t = h_t \quad (2)$$

Where, x_t is the input, W_h and W_x are the weights, b is the bias, y_t and h_t are the output and recurrent output respectively. Despite the RNN ability to recognize short-term dependencies, the vanishing gradient problem (VGP) prevents them from capturing the long-term ones. The VGP occurs during training neural networks when the gradients that are used to update the weights shrink dramatically. As a result, the process of updating weights stops, and the learning process ends. To address the VGP in RNN, Hochreiter and Schmidhuber devised LSTM cell, which will be explained in the next section (Hochreiter and Schmidhuber, 1997).

3.3 LSTM

Long short-term memory (LSTM) is a special type of RNN that was designed to address the VGP (Hochreiter and Schmidhuber, 1997). LSTM ~~has the ability to~~can capture the dependencies between long sequences such as those between time series datasets by memorizing information for lengthy periods of time (Siami-Namini *et al.,* 2019). There are several applications of LSTM like text translation, music composition and recently time series forecasting.

The basic structure of LSTM called a memory block, where each memory block contains two cell states and three gates: forget, input, and output. The purpose of forget gate is to decide what information should be allowed to enter or removed from the memory block. The input gate decides what information should be stored or updated in the cell state, while the output gate is responsible for identifying the next value of the hidden state, where the hidden state contains information about the previous input (Azzouni and Pujolle, 2017). Figure 3 reveals the basic structure of LSTM cell which composed of two cell states and three gates: input gate, output gate, and forget gate.

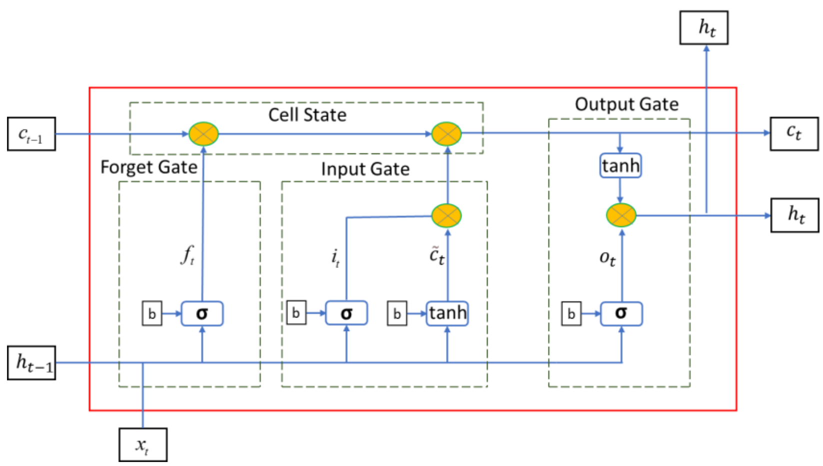

Figure 3. LSTM architecture.

The mathematical equations of LSTM cell are the following:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (3)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (4)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (6)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (7)$$

$$h_t = o_t * tanh(c_t) \quad (8)$$

Where f_t , i_t , and o_t refer to forget gate, input gate, and output gate. W and U denote to the weight matrices, while b denotes to the bias. h_t and h_{t-1} represent the outputs at the current and previous time steps consecutively. x_t refers to the input of LSTM cell, c_t , c_{t-1} denote to the output and input of the cell state, $\tilde{c}_t$ is the output of tanh function. One of the issues that encountering LSTM is that it does not have the ability to see the future time samples, BiLSTM, which will be explained in the next section, was developed to address this issue.

3.4 BiLSTM

BiLSTM is a modern kind of RNN that integrates the traditional RNN with LSTM cell. BiLSTM was invented by Graves and Schmidhuber in 2005 to overcome the limitations in unidirectional LSTM by deploying two LSTM cells (Graves and Schmidhuber, 2005). These two cells are responsible for processing the sequences in both directions, one from front to back and the other from back to front (Althelaya *et al.,* 2018).

Applying the LSTM twice improves the learning of dependencies and correlations between the long sequences and therefore obtaining more accurate outputs (Siami-Namini *at al.,* 2019). In fact, BiLSTM networks achieve more accurate results than unidirectional LSTM and simple RNN in many applications like phoneme classification and time series prediction. The structure of BiLSTM is illustrated in ~~figure~~ Figure 4.

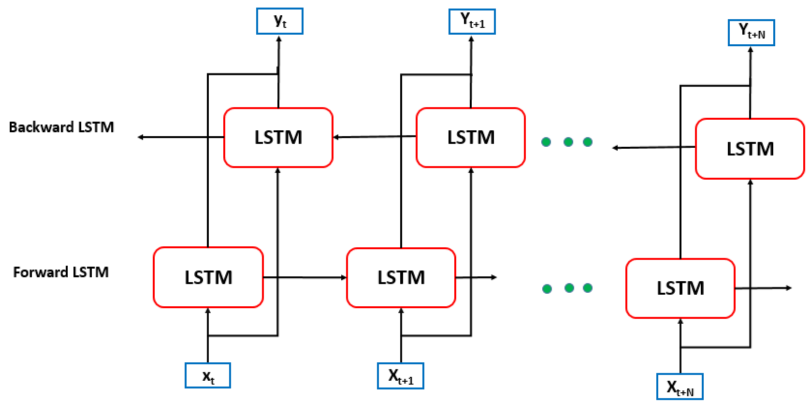

Figure 4. BiLSTM structure.

3.5 Deep ~~neural~~ Neural ~~network~~ Network (DNN)

DL is a part of ML inspired by the patterns of information processing found in the human brain. DL utilizes a large amount of data to detect and discover the relationships between the given inputs and a specific output rather than using human designed rules (Sharker, 2021). DNN is defined as an ANN that have more than one hidden layer between the input and output layers. DNNs have been successfully applied in a several applications like healthcare, machine translations, handwriting analytics, and natural language processing due to their high learning capabilities from dataset. CNNs, DNNs, deep belief networks, deep reinforcement learning, and RNNs are the most prominent types of DL. In this thesis, we used a DNN to design a weather forecasting model with five BiLSTM layers.

3.6 Sequence to Sequence Prediction Problem (Seq2Seq)

The term "*sequence prediction*" refers to the process of predicting the set of next values, labels or sequence based on a series of the previous values. There are four types of sequence prediction:

Formatted: Font: Italic, Complex Script Font: Italic one input time step to one output time step (one-to-one), multiple input time steps to one output time step (many-to-one), one input time step to multiple output time steps (one-to-many), and multiple input time steps to multiple output time steps (many-to-many) or Seq2Seq (Sutskever *et al.,* 2014). In fact, we interest in this thesis with the fourth type of sequence prediction, Seq2Seq. A major concern in seq2seq prediction is that the length of the input sequence is different from the length of the output sequence. (Sutskever *et al.,* 2014). There are many applications for seq2seq such as, text translation, speech recognition, and time series forecasting where the input length differs from the output length (Sutskever *et al.,* 2014).

3.7 Encoder-Decoder BiLSTM (ED-BiLSTM)

The Encoder-Decoder (ED) is a RNN architecture that was developed recently in 2014 to address the Seq2Seq prediction problem. The architecture of the ED model composed of two parts: encoder and decoder, where each part has multiple layers of any type of cell like MLP, CNN, RNN, LSTM, and GRU. In our thesis, we incorporated BiLSTM cell with an ED architecture to construct the proposed model. The proposed model composed of five BiLSTM layers, two for the encoder and three for the decoder, as well as a repeat vector layer and a time distributed dense layer.

The encoder composed of two BiLSTM layers and the aim of it is to discover and learn the relationships between the elements in the input sequence and represent it as a fixed-size vector called context vector, where this vector is given to the decoder as input (Cho *et al.,* 2014). The decoder part composed of three BiLSTM layers and is responsible for producing predictions by processing and decoding the context vector (Malhotra *et al.,* 2016).

In fact, the output of the encoder is in 2-dimensional, while the input of the decoder should be 3D which includes samples, time steps and number of features. In fact, to make the output of encoder adapts with the input of decoder, RepeatVector layer is utilized between them to reshape the context vector from 2D to 3D by repeating the provided 2D input by n-steps (n is the number of future steps that we want to predict) to create a 3D output (Zaytar and Amrani, 2016). The time distributed dense layer is responsible for applying a fully connected dense layer on each element in the output sequence (Zaytar and Amrani, 2016). Figure 5 exhibits the basic architecture of ED-BiLSTM model.

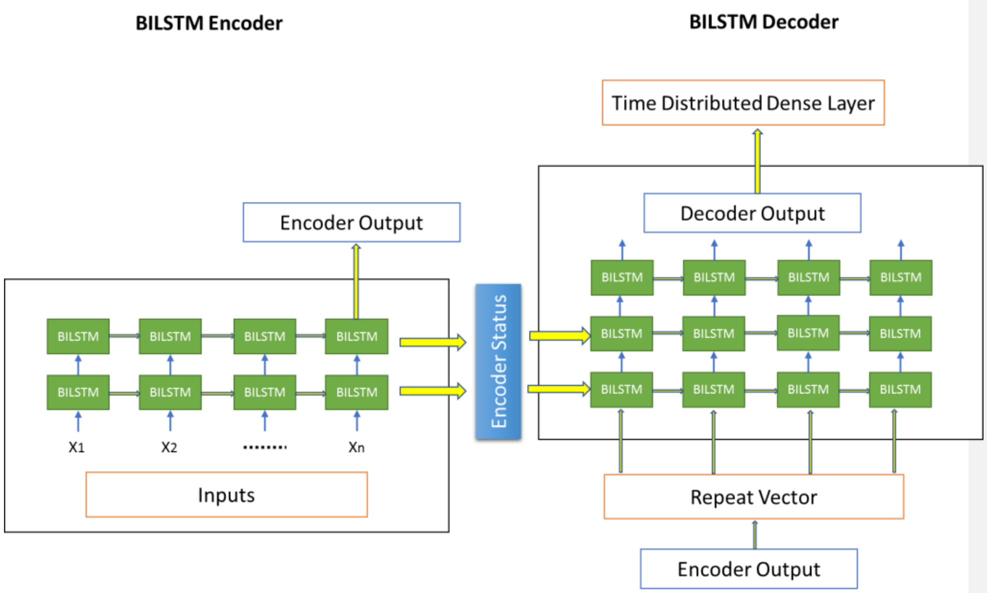

Figure 5. ED-BiLSTM model architecture.

3.8 Evaluation Metrics

The performance of ML models is measured using evaluation metrics. In our thesis, we use MSE to assess the performance of the proposed model.

3.8.1 Mean ~~squared~~ Squared ~~error~~ Error (MSE)

MSE is a common error metric for regression and time series problems and is an estimator that measures how close the expected value is to the actual value (Jierula *et al.,* 2015). MSE is always a positive value due to error value is squared between the actual value and predicted value. The relationship between MSE and prediction accuracy is inverse. The lower MSE value, the better the prediction accuracy and vice versa. Basically, the mathematical equation of MSE is represented as follows:

$$MSE = \frac{1}{N}\sum_{1}^{N}\left(y_i - \hat{y}_i\right) \tag{9}$$

Where N refers to the number of data samples, while y_i and $\hat{y}_i$ denote to the actual values and predicted values.

Formatted: Font: Italic, Complex Script Font: Italic

Chapter 4: Datasets Analysis and Model Training

4.1 Introduction

In this thesis, we used weather dataset for all Jordanian governorates which include Amman, Irbid, Zarqa, Jerash, Ajloun, Mafraq, Madaba, Balqa, Karak, Tafilah, Aqaba, and Maan with some areas surrounding Jordan, like Jerusalem, Damascus, ~~Cyprus~~Cyprus, and Cairo. The dataset was obtained from www.worldweartheronline.com, www.ncdc.noaa.gov, power.larc.nasa.gov and www.arabiaweather.com. The dataset consists of hourly readings of the following parameters: average air temperature, ultraviolet index, atmospheric pressure, feels-like temperature, humidity, heat index and total precipitation from 1/1/2009 to 31/12/2021. Air temperature (°C) is a scale that indicates whether the air is cold or warm. Ultraviolet index is a measure of the level of ultraviolet radiation. Atmospheric pressure (mb) is the force with which the weight of the air column affects the unit of area. Feels like temperature (C) is a measurement of how hot or cold it really feels like outside. Humidity (%) refers to the amount of water vapor present in the air. Heat index (C) is an indicator that combines the air temperature with the humidity to determine the temperature that a human really feels. Precipitation (mm) is defined as all forms of water that fall from the clouds towards the ground, whether in the form of liquid rain, freezing rain, snow, frost, or hail. The weather dataset is divided into three subgroups: 70% training, 15 % validation, and 15% testing. The training dataset is used to train and build the model, whereas the testing and validation datasets are used to assess the model. Table 1 gives a glimpse of the weather dataset.

Commented [GA2]: Change all C to °C

Table 1. Glimpse of the weather dataset.

Location ID ~~id~~	Date	Average Temperature (C)	Precipitation (mm)	Humidity (%)	Pressure (mb)	Feels Like Temperature (C)	Heat Index (C)	Ultraviolet Index
Amman	2009.01.01 00:00:00	2	0	90	1020	-1	3	1
Amman	2009.01.01 01:00:00	2	0	91	1021	-1	3	1
Amman	2009.01.01 02:00:00	1	0	92	1021	-2	2	1
Amman	2009.01.01 03:00:00	0	0	94	1021	-2	1	1
Amman	2009.01.01 04:00:00	1	0	91	1021	-2	2	1

Formatted Table

4.2 Visualizing Dataset

This section deals with representing the dataset in several graphs. Different types of graphs were produced to visualize the trends and correlations the between the weather features. Figure 6 shows the yearly average temperature in all Jordanian governorates from 2009 to 2021. Figures 7 and 8 exhibit the monthly average maximum and minimum temperature in all Jordanian governorates between 2009 and 2021.

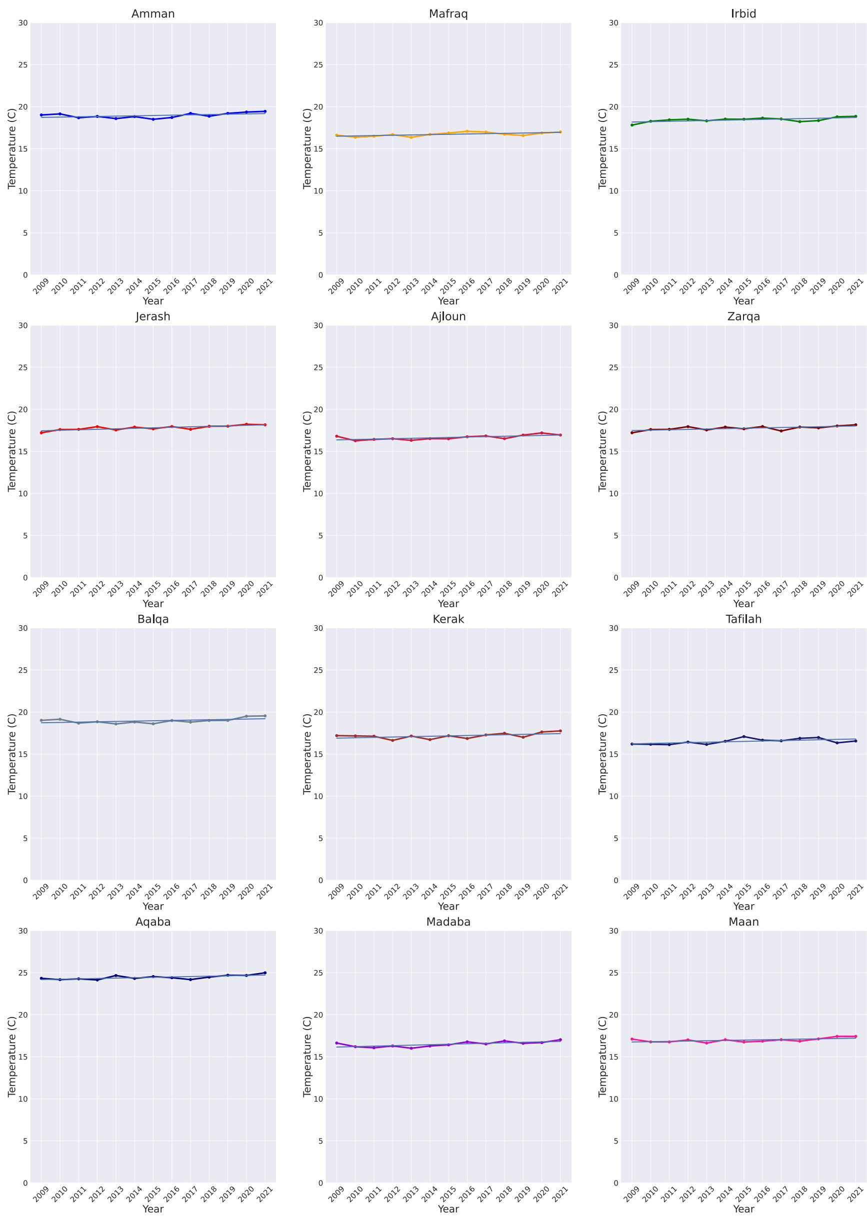

Figure 6. Yearly average temperature in all Jordanian governorates between 2009 and 2021.

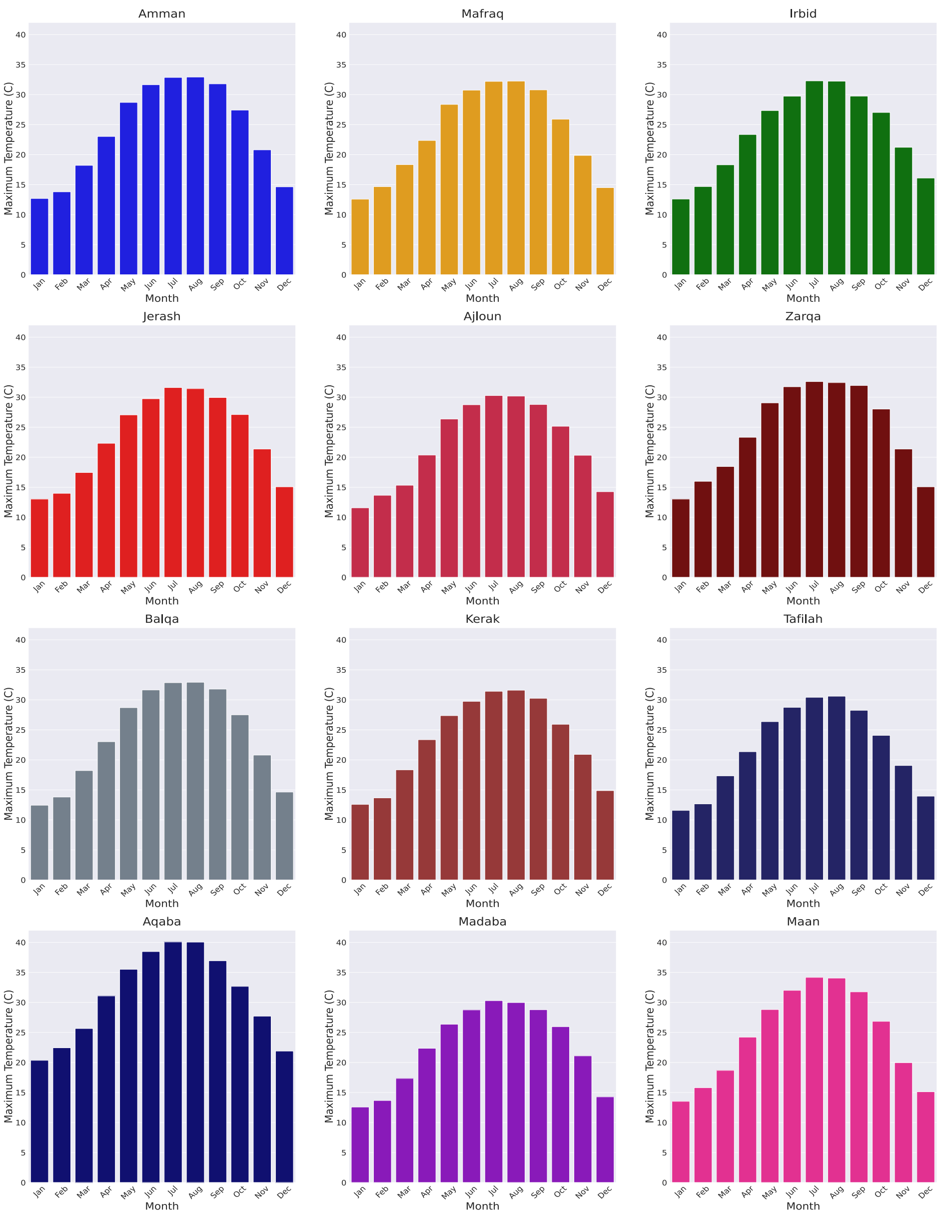

Figure 7. Monthly average maximum temperature in all Jordanian governorates between 2009 and 2021.

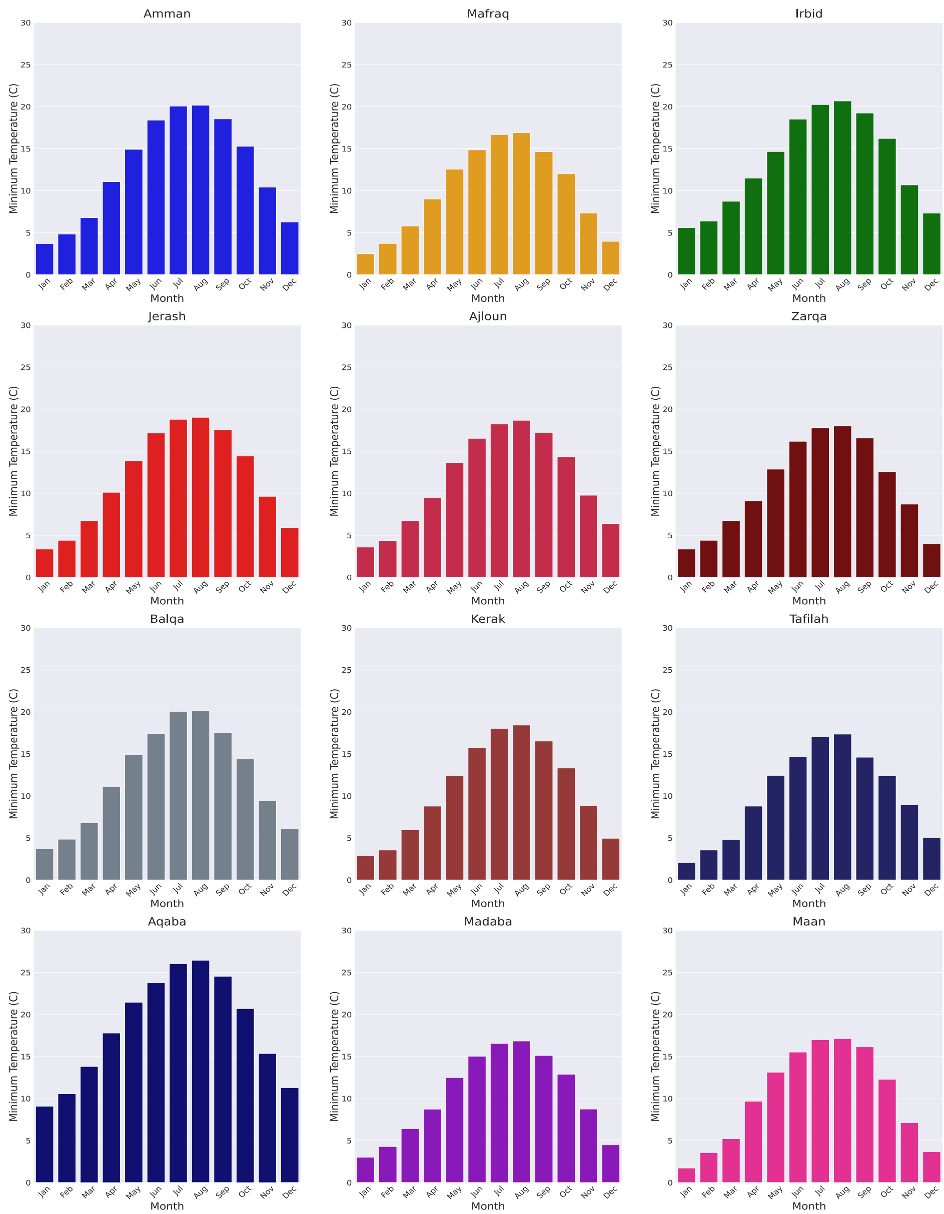

Figure 8. Monthly average minimum temperature in all Jordanian governorates between 2009 and 2021.

It can be noticeable from ~~figure~~ Figure 6 that there is a very slight increase in the average air temperature in all Jordanian governorates during the past 13 years. In fact, the average temperature during the period 2009-2021 in most Jordanian governorates fluctuates between 16.5 C and 19 C except Aqaba, where the average temperature is around 24.5 C. Also, from ~~figure~~ Figures 7 and 8 we can say that the monthly average minimum and maximum temperature in Jordan follows the same trend. The lowest value of minimum and maximum temperature is in January. Then, it starts to increase until it reaches the highest value in August. After that, it begins to decline until the end of the year. The following two ~~figures~~ Figures 9 and 10 illustrate the yearly and monthly average humidity in all Jordanian governorates between 2009 and 2021.

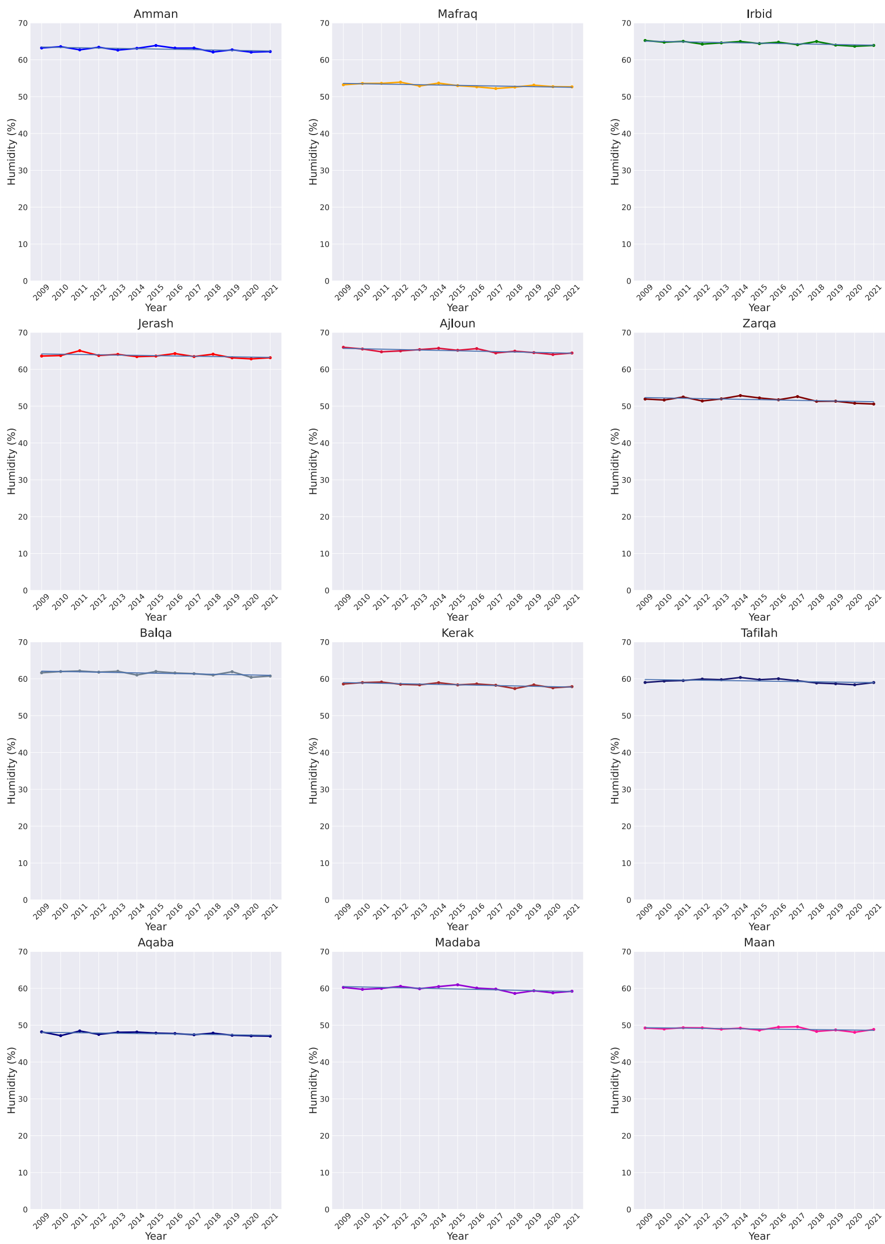

Figure 9. Yearly average humidity in all Jordanian governorates between 2009 and 2021.

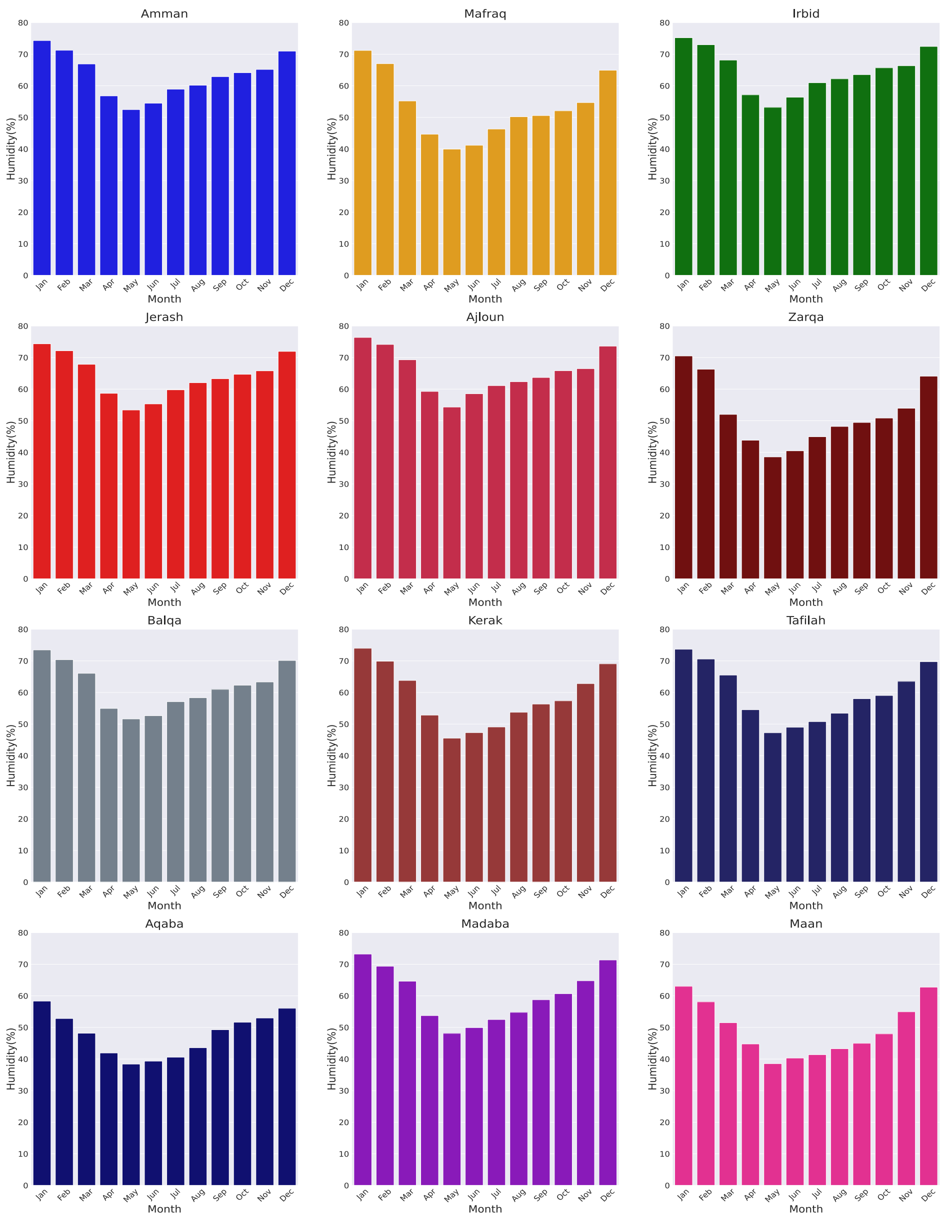

Figure 10. Monthly average humidity in all Jordanian governorates between 2009 and 2021.

Figure 9 shows that the changing in the average humidity follows the same trend in all Jordanian governorates. The average humidity fluctuates between 58 and 66 in most Jordanian governorates, except Mafraq, Zarqa, Aqaba and Maan, where the average humidity around 53 ,52, 48, and 50, respectively. Also, we can conclude from figure 10 that the highest average humidity is in January, then it begins to decline till reaching the minimum value in May. After that, it returns to increase until the end of the year. Figure 11 displays the yearly total precipitation in all Jordanian governorates from 2009 to 2021. Whereas figure 12 exhibits the monthly total precipitation in all Jordanian governorates during the period between 2009 and 2021.

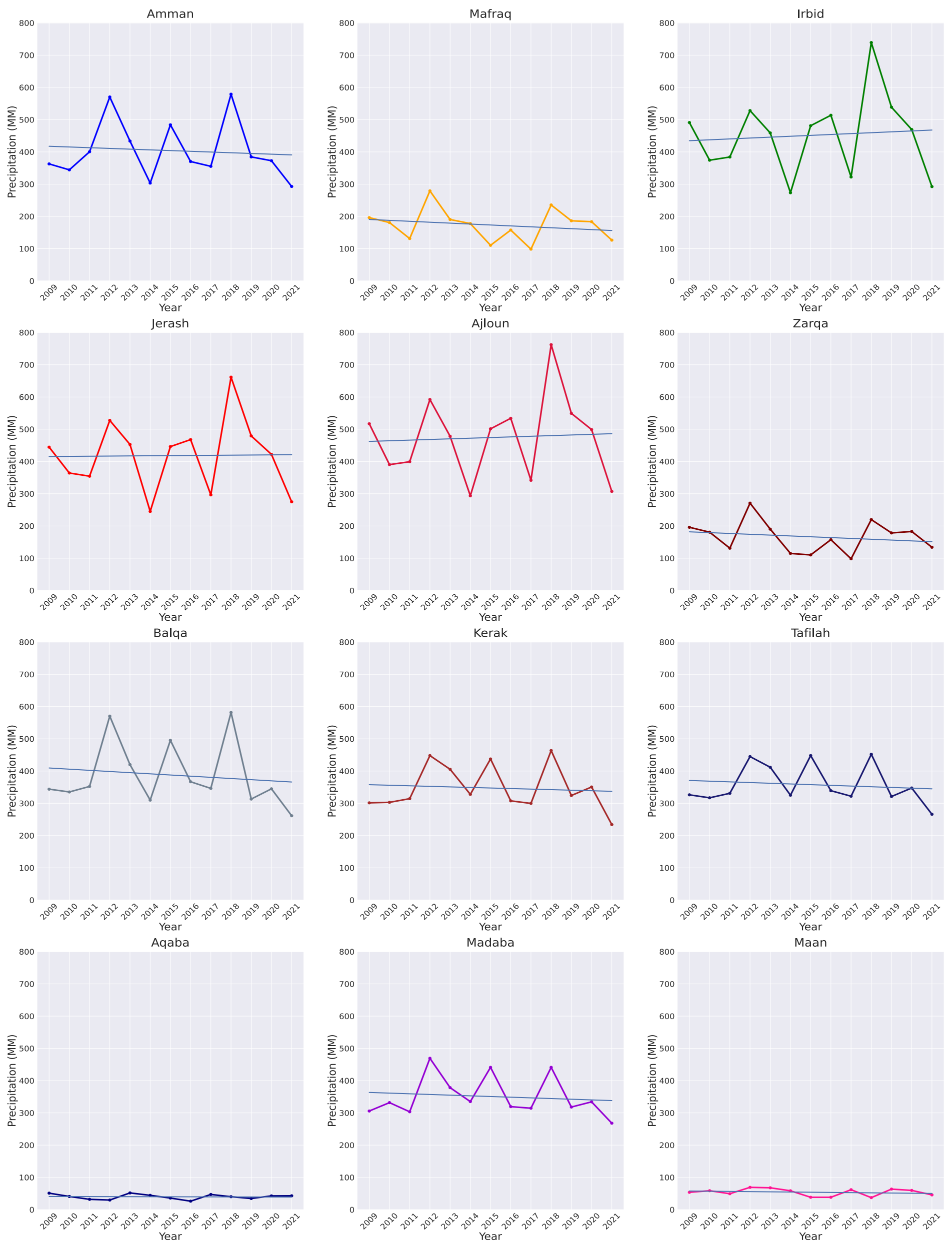

Figure 11. Yearly total precipitation in all Jordanian governorates between 2009 and 2021.

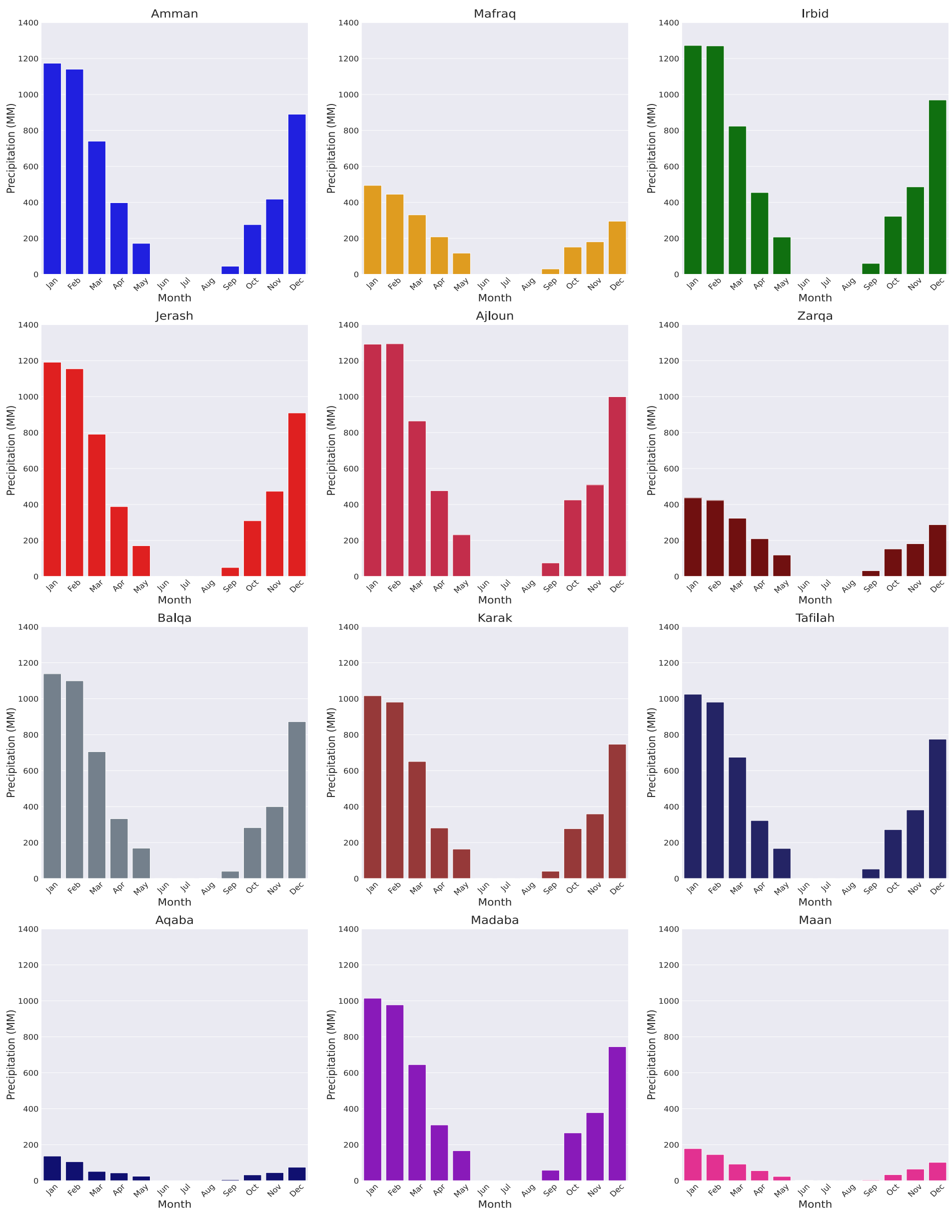

Figure 12. Monthly total precipitation in all Jordanian governorates between 2009 and 2021.

Figure 11 shows that the amounts of precipitation have decreased in all governorates except Irbid, Jerash, and Ajloun, where these three governorates have witnessed an increase in the precipitation over the past 13 years. In fact, there are fluctuations in year-to-year precipitation in Jordan. In fact, the two governorates that have the highest precipitation rates are Ajloun and Irbid. While the lowest precipitation rate is in Aqaba. Additionally, we can conclude from ~~figure~~ Figure 12 that the highest precipitation rates are during January and February, and there is no precipitation during the summer period, from June to August. We can see the correlation between the weather features as a heatmap in ~~figure~~ Figure 13.

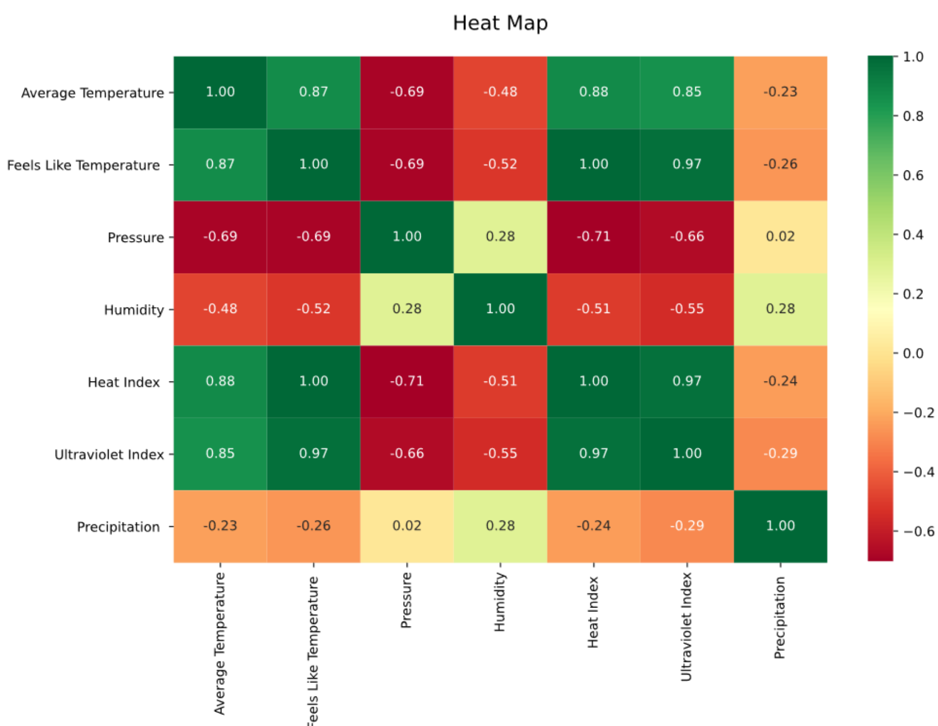

Figure 13. Heatmap showing the correlation among the weather features.

As the ~~figure~~ Figure 13 shows, there are high correlations between some features like those between the average temperature, feels like temperature, heat index and ultraviolet index. On the other hand, there are very weak correlations such as those between the precipitation and atmospheric pressure. Figure 14 illustrates the types of correlations between the weather features.

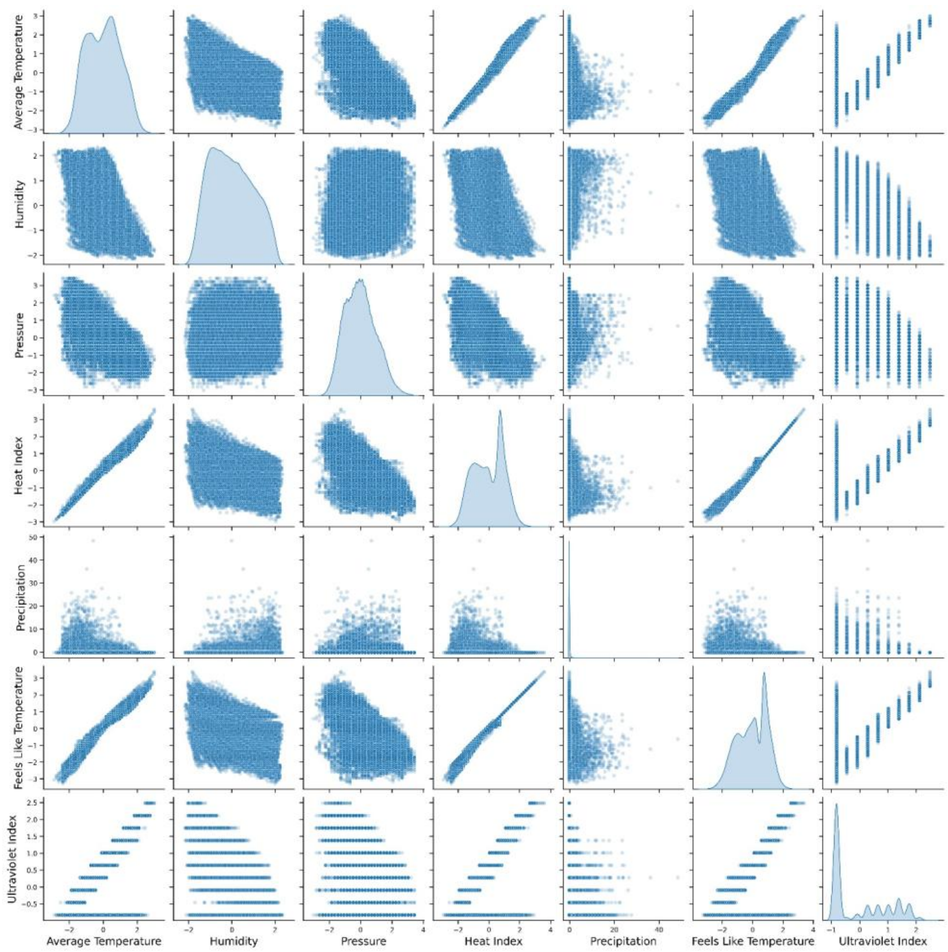

Figure 14. Scatter diagram showing the correlation between the weather features.

4.3 Data Preprocessing

As described in the earlier section 4.1. The dataset consists of hourly readings of seven weather parameters for all Jordanian governorates and some areas near Jordan like Jerusalem, Damascus Cyprus, and Cairo from 1/1/2009 to 31/12/2021. In fact, preparing the dataset for training consists of several steps. Firstly, reading the weather dataset from a comma separated values (CSV) file. Secondly, normalizing the dataset using min and max scaler to maintain all the values in the range from -1 to 1 by applying the following equation:

$$p_i = 2 \times \left(\frac{p_i - \min(p_i)}{\max(p_i) - \min(p_i)} \right) - 1 \tag{10}$$

Thirdly, constructing temporal sliding windows to make the dataset suitable for the model training. Where each window uses the consecutive past 120 hours of all-weather features to predict the range of future hours. Then, the sliding windows are shuffled randomly to avoid overfitting issue. Finally, the shuffled sliding windows are split into three parts, 70% for training, 15% for validation and 15% for testing. The training dataset is utilized to teach and build the proposed model, the validation dataset is used to give an initial evaluation of the proposed model about new dataset, whereas the testing set is used to give the final evaluation of the model accuracy.

4.4 Model Training

The following hyperparameters for the model were selected after preparing the dataset: batch size by 128, and seed of the random shuffle by 13 to ensure that the data shuffling is performed in the same way each time.

When we implemented the experiments of enhancing the base model and training the proposed model on each Jordanian governorate individually, we fixed the number of epochs by 1000 and early stopping by five to stop the training after five epochs of increasing in the validation loss. In contrast, we were unable to make a continuous training on the whole dataset due to the limitations in Colab session length of 24 h. So that, we resorted to training the model in the following way. First, the length of Colab session is divided by the time of one epoch to get the number of epochs suitable with the session length. Next, the proposed model is trained for that number of epochs and saving the best weight. After that, the best obtained weight is loaded, and retraining process is carried out by that number of epochs. This process is implemented numerous times until reaching the minimum loss.

4.5 Experiments Platforms

This section describes the experimental environments that were used in building the base model, making the experiments to enhance it, and training the proposed model to get the ~~final results~~results. In fact, two experimental platforms were used in this thesis: Kaggle and Colab pro plus. Graphics processing unit (GPU) accelerators in both platforms were used in training and testing the proposed model to reduce training time. Tables 2 and 3 show the specifications of Kaggle and Colab pro plus platforms, consecutively.

Table 2. Kaggle platform specifications.

Components	Specification
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz, 2 cores, 56 MB cache
TPU	v3-8
GPU	Nvidia-Tesla P100-PCIE @ 1.32GHz, 16 GB
Memory	16 GB
Libraries	Python 3.7.10, TensorFlow 2.4.1 seaborn 0.11.2, Pandas 1.3.2, Matplotlib 3.4.3, NumPy 1.19.5 Sklearn 0.23.2

Table 3. Colab Pro Plus platform specifications.

Components	Specification
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz, 4 cores, 56 MB cache
TPU	v2
GPU	Nvidia-Tesla V100-SXM2 @ 1.32GHz, 16 GB Nvidia-Tesla P100-PCIE @ 1.32GHz, 16 GB
Memory	53 GB
Libraries	Python 3.7.13, TensorFlow 2.8.2 Seaborn 0.11.2, Pandas 1.3.5, Matplotlib 3.2.2, NumPy 1.21.6 Sklearn 1.0.2

Chapter 5: Model Experiments and Results

5.1 Introduction

In this chapter, we will discuss the base model of the weather forecasting, the trials and experiments that have been conducted to reach the proposed model, and the results of weather forecasting in Jordan. We used MSE as an evaluation metric to show the high ability of the proposed model in predicting the weather conditions. In fact, Amman dataset was used in all the experiments of model enhancement to find the effect of each trail and fine tuning hyperparameter on the model performance quickly and easily. The dataset was split into three subsets: training (70%), validation (15%), and testing (15%).

5.2 Base Model

The base model of weather forecasting called Encoder-Decoder LSTM (ED-LSTM) and it is based on state-of-the-art LSTM cell (Hewage *et al.,* 2020). More specifically, the base model consists of the following five layers respectively: pass-through input layer, LSTM hidden layer with 128 nodes, repeat vector layer for the number of future hours to be predicted, two LSTM hidden layers with 512, 256 nodes consequently, and time distributed layer with a dense of seven nodes, which is the number of weather features, as shown in figure 15. This type of model called MIMO, where all the weather parameters are entered into the model to predict all of them. The activation function that is used with LSTM cell is the default tanh function. The SGD optimizer with learning rate of 0.01 and a batch size of 128 are used in training the model. Also, we constructed sliding windows, where each one uses the previous 120 hours of all-weather parameters to predict the next nine hours. Figure 15 shows the architecture of the ED-LSTM base model. Table 4 shows the performance of the base model in terms of MSE value.

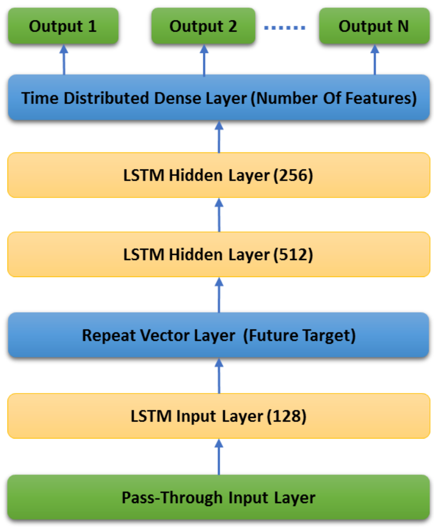

Figure 15. The architecture of the ED-LSTM base model.

Table 4. The performance of the base model in terms of MSE value.

Type Of Cell	Optimizer	MSE on Testing Dataset	Number of Epochs	Execution Time
LSTM	SGD	0.0136	288	1.48 hrs.

5.3 Model Experiments

During this part, we will discuss the experiments that have been carried out on the base model to enhance its performance and to achieve the optimal results. Also, we have performed numerous fine ~~tuning~~tunings of hyperparameters and testing several architectures of the base model by changing the type of optimizer, type of cell, number of hidden layers, number of nodes and so on to get the best possible architectural configuration.

5.3.1 Type of Optimizer

In this experiment, we have tested several types of optimizers: Adam, stochastic gradient descent with Momentum (M=0.9) and Nestrov, and root mean squared propagation (RMSProp), then comparing them with SGD to see which is better for the base model. In fact, the aim of optimizers is to update the attributes of neural networks like weights and biases during training ~~in order to~~to reduce the losses. Table 5 shows a comparison between the four tested optimizers in terms of MSE values.

Table 5. Comparison between four optimizers in terms of MSE values.

Type Of Optimizer	MSE	Number of Epochs	Execution Time
SGD	0.0136	288	1.48 hrs.
SGD with Momentum and Nestrov	0.0051	28	9.11 min
RMSProp	5.5263e-04	70	23.12 min
Adam	3.4209e-04	93	28.86 min

It can be noticeable from the above table that the Adam optimizer proves superior performance in prediction than other three optimizers. It demonstrates an MSE of 3.4209e-04 on testing dataset. RMSProp achieves the second best MSE by 5.5263e-04. However, SGD and SGD with Momentum and Nestrov achieve bad performance during our forecasting task. So that, Adam was chosen as the default training optimizer for the next experiments.

5.3.2 Type of Cell

The aim of this experiment is to find the optimal type of cell for the base model by experimenting two other types of cells, GRU and BiLSTM and compared them with LSTM. Table 6 shows the performance of each type of cell in terms of MSE values.

Table 6. Performance comparison between three types of cells in terms of MSE values.

Type Of Cell	MSE	Number of Epochs	Total Training Time
GRU	5.2314e-04	95	29.88 min
LSTM	3.4209e-04	93	28.86 min
BiLSTM	1.9357e-04	79	49.78 min

Formatted Table

It can be noticeable from the above table that the BiLSTM achieves better performance than other two types of cells, LSTM and GRU, this is because the BiLSTM can see and process the past and future elements in the sequences at same time. The only shortcoming of BiLSTM is that it takes a little more time in training compared to LSTM and GRU. Also, LSTM performs better than GRU. Among three types of cells, the GRU model achieves the lowest performance. It demonstrates MSE by 5.2314e-04 on testing dataset. Hence, BiLSTM with Adam optimizer were chosen for the rest of experiments.

5.3.3 Number of Layers

In this experiment, we have changed the architecture of the best obtained model from previous experiments, BiLSTM with Adam, by adding and removing some hidden layers to the base model to reach the optimal topology. Figure 16 shows the architecture of the base model and other three tested models that have different number of layers. Also, table 7 shows a comparison between the four tested models in terms of MSE values.

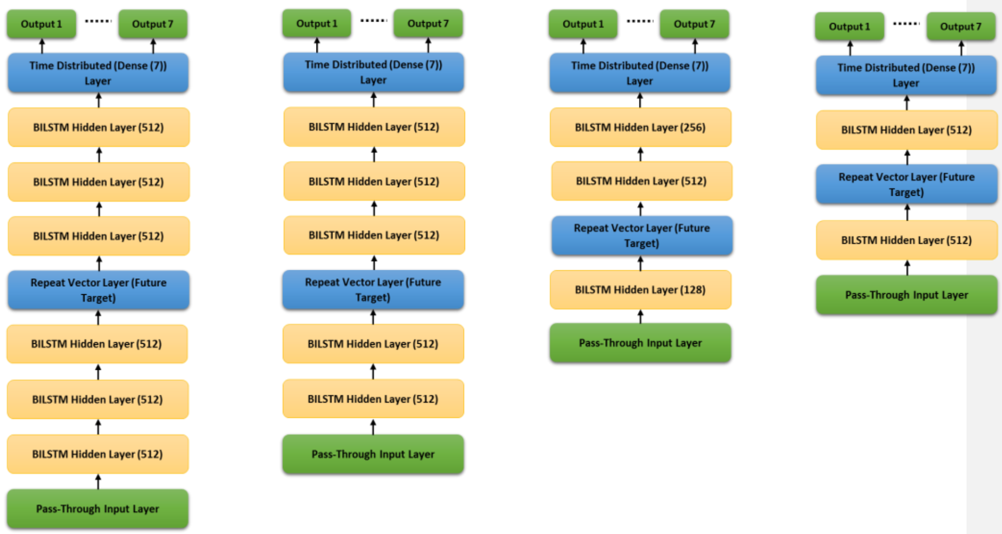

Figure 16. The architecture of the base model and three other models that have different number of layers.

Table 7. Comparison between the four tested models in terms of MSE values.

Number Of BiLSTM Layers	MSE	Number of Epochs	Execution Time
2	2.2411e-04	92	1.88 hrs.
3	1.9357e-04	79	49.78 min
5	1.2732e-04	65	2.61 hrs.
6	1.3869e-04	62	4.09 hrs.

From table 7, ~~it is clear that the~~the removing some hidden layers from the base model negatively affects the model performance. While, adding some BiLSTM layers up to five layers increases gradually the effectiveness of the model. In fact, the model that contains five BiLSTM layers, two layers in the encoder and three layers in the decoder, with 512 nodes per layer gives the best performance with the lowest error of 1.2732e-04 on testing dataset. So that, we have chosen this model as a base model for the next experiment.

5.3.4 Number of Nodes in Each Layer

An additional experiment was performed to test the effect of changing the number of nodes in the hidden layers on the performance of the base model. During this experiment, we have changed the number of nodes in the encoder part to 256 and 128 respectively. Whereas, in the decoder part the numbers of nodes were changed into 128,256, and 512 consecutively. Figure 17 shows the architecture of the base model and another tested model that has different number of nodes. The results obtained from this experiment in terms of MSE values are shown in table 8.

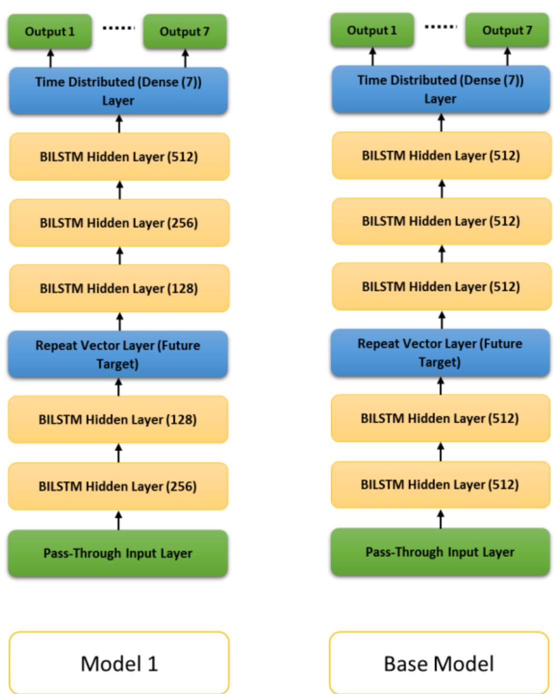

Figure 17. The architecture of the base model and another tested model that has a different number of nodes.

Table 8. Performance comparison between the base model and another tested model that has a different number of nodes in terms of MSE values.

Type of Model	MSE	Number of Epochs	Execution Time
Base Model	1.2732e-04	65	2.61 hrs.
Model 1	2.6230e-04	76	1.57 hrs.

Formatted Table

According to the presented results in ~~table~~ Table 8, ~~it is clear that giving~~giving different numbers of nodes to hidden layers does not improve the model performance, while giving the same large number of nodes (512) to all BiLSTM layers improves the model performance significantly. In fact, the model that contains five BiLSTM layers each with 512 nodes, with repeat vector layer and time distributed dense layer gives high performance with little error value compared to another tested model. Therefore, it was chosen as a base model for the next experiment.

5.3.5 Different Size of Encoder and Decoder

This experiment was conducted to find the suitable size of encoder and decoder of the base model that gives the best performance. In fact, several sizes of the encoder and decoder were tested in this experiment by adding and removing some hidden layers to each part. Figure 18 illustrates the structure of the four models, including the base model, that have different sizes of encoder and decoder. Also, ~~table~~ Table 9 presents a comparison between the proposed model and other three tested models that have different sizes of encoder and decoder based on MSE values.

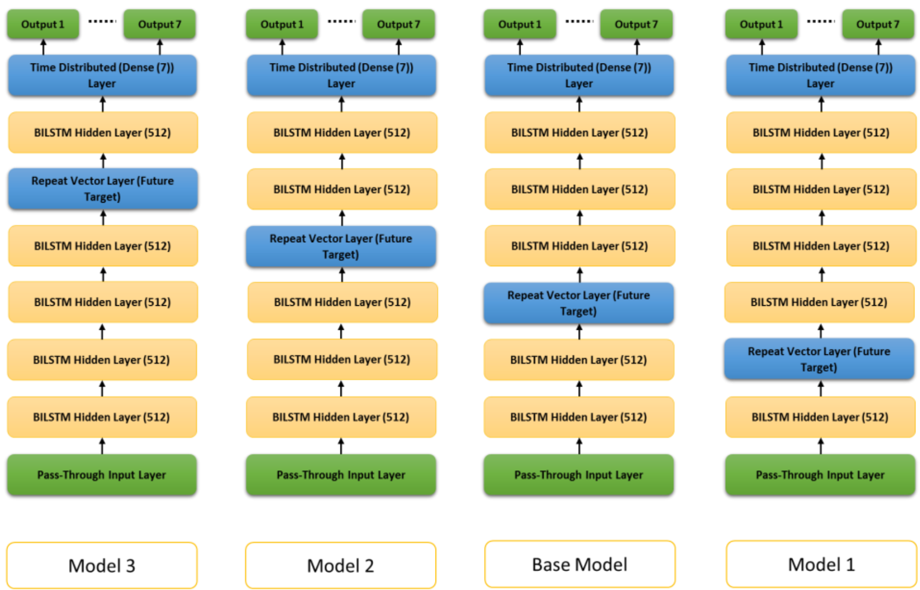

Figure 18. The structure of the base model and other three models that have different sizes of encoder and decoder.

Table 9. Comparison between the base model and other three models that have different size of encoder and decoder based on MSE values.

Model Number	MSE	Number of Epochs	Execution Time
Model 3	2.1607e-04	56	4.68 hrs.
Model 2	1.7278e-04	55	3.15 hrs.
Model 1	1.6956e-04	65	1.51 hrs.
Base Model	1.2732e-04	65	2.61 hrs.

Formatted Table

It can be concluded from the ~~table~~ Table 9 that the base model that have two BiLSTM layers in the encoder and three BiLSTM layers in decoder achieves better prediction accuracy than the other three tested models. It achieves MSE by 1.2732e-04. Then it followed by ~~model~~ Models 1 and 2 with 1.6956e-04 and 1.7278e-04 respectively. Whereas the ~~model~~ Model 3 that contains four layers in encoder and one layer in decoder achieves the lowest performance with MSE of 2.1607e-04.

5.3.6 Different Normalization Techniques

Normalization refers to the process of converting the feature values into a common scale without affecting the difference between them and the quality of the results. In fact, the weather parameter values have different ranges, so that the normalization is required to make the data values uniform. In fact, we have tested three different normalization techniques: min-max scaler, standard scaler and default scaler of the base model to choose the one that gives the best performance. Table 10 shows a performance comparison between three different normalization techniques based on MSE values.

Table 10. Performance comparison between three different normalization techniques based on MSE values.

Normalization Techniques	MSE	Number of Epochs	Execution Time
Standard Scaler	0.0114	47	1.91 hrs.
Default Scaler	1.2732e-04	65	2.61 hrs.
Min-Max Scaler	3.3928e-06	142	3.84 hrs.

Formatted Table

Based on ~~tables~~ Table 10, it can be noticeable that the Min-Max scaler outperforms the other two normalization techniques in terms of MSE values, as it achieves MSE of 3.3928e-06. While the standard scaler and the default scaler achieves MSE by 0.0114 and 1.2732e-04, respectively. Therefore, we have chosen the Min-Max as a default scaler for our proposed model.

5.3.7 Different Datatypes

This experiment was performed to test the effect of changing the datatype on the model performance. The default datatype of the dataset is `float64`. It is known that the higher datatype, the more RAM it requires. However, we have limited resources, RAM, and GPU RAM. So that,

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New we have reduced the datatype to be suitable with available resources. The two datatypes that were tested in this experiment include: `float16` and `float64`. Table 11 summarizes the results of implementing two different datatypes on the model performance in terms of MSE values.

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Table 11. Performance comparison between two datatypes based on MSE values.

Datatypes	MSE	Number of Epochs	Execution Time
Float 16	4.9726e-06	111	2.44 hrs.
Float 64	3.3928e-06	142	3.84 hrs.

Formatted Table

As shown in ~~table~~ Table 11, the model performance decreases as the datatype decreases. The model shows the best performance with a datatype of `float64`. It achieves MSE by 3.3928e-06 on testing dataset. Whereas the `float16` gives MSE by 4.9726e-06. Although `float64` achieves slightly better results compared to `float16`, we have adopted `float16` as a default datatype, due to its compatibility with available resources, RAM and GPU RAM.

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

5.3.8 Different ~~periods~~ Periods of ~~prediction~~Prediction

This experiment was performed to identify the effect of increasing the time prediction on the model performance. In fact, we have tested two different ~~period of time~~periods: 9 and 24 hours. Table 12 shows the model performance with two different periods of time.

Table 12. The performance of the base model based on two different period of times, 9h and 24h in terms of MSE values.

Time periods	MSE	Number of Epochs	Execution Time
9 hours	4.9726e-06	111	2.44 hrs.
24 hours	3.4811e-05	72	3.69 hrs.

Formatted Table

It can be noticeable from ~~table~~ Table 12 that the performance of the base model decreases with the forecast time increases from 9-hour to 24-hour ahead. However, the model performance is still excellent when predicting 24 hours ahead. In general, the forecasts become less accurate and more expensive as the range of them increases.

5.4 Proposed Model

After trying, testing, and experimenting several hierarchies and parameters of the base model, we have adopted the best obtained one from previous experiments as a proposed model for the weather forecasting. The architecture of the proposed model consists of the following layers respectively: pass-through input layer, two BiLSTM layers each with 512 nodes, repeat vector layer, three BiLSTM hidden layers each with 512 nodes, and time distributed dense layer with seven nodes as shown in Figure 19. The resulting neural network has 27,319,303 parameters. Table 13 illustrate the values of the proposed model parameters that achieve the best results.

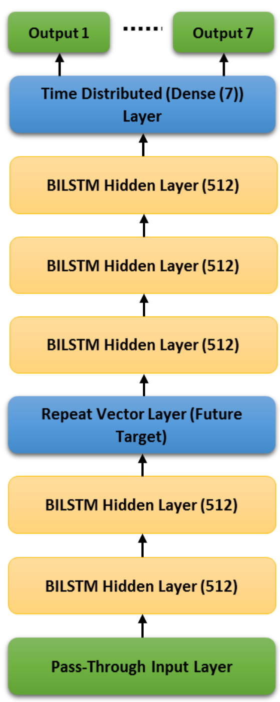

Figure 19. The architecture of the proposed model.

Table 13. The values of the proposed model parameter that achieve the best results.

Parameters	Value
Number of BiLSTM Layers	5 Layers
Number of Nodes in Each Layer	512 nodes
Optimizer	Adam
Learning Rate	0.001
Normalization	Min-Max
Evaluation Metric	MSE
Number of Model Parameters	27,319,303
Batch Size	128

Formatted Table

5.4.1 Training the Proposed Model on Each Jordanian Governorate

In this experiment, we have trained the proposed model on each Jordanian governorate individually to predict the weather conditions over the next 24h. Firstly, we have split the dataset of each governorate into three part which are training: validation: testing by 70:15:15. Then~~-~~, the dataset of each governorate was entered into the model for training and getting the result~~-~~. Table 14 exhibits result of the weather forecasting over the next 24h in all Jordanian governorates. In fact, we call this experiment Each-Gov-Inp.

Table 14. The result of the weather forecasting over the next 24h in all Jordanian governorates.

Governorates	MSE	Number of Epochs	Execution Time
Balqa	4.1259e-05	63	3.03 hrs.
Jerash	3.8685e-05	67	3.37 hrs.
Zarqa	3.8617e-05	64	3.37 hrs.
Amman	3.4811e-05	72	3.69 hrs.
Irbid	2.6153e-05	78	3.78 hrs.
Madaba	2.3771e-05	92	4.82 hrs.
Karak	1.3621e-05	101	5.22 hrs.
Mafraq	1.3554e-05	112	5.70 hrs.
Aqaba	1.1012e-05	130	6.34 hrs.
Ajloun	1.0998e-05	132	6.76 hrs.
Maan	7.1198e-06	166	8.13 hrs.

Formatted Table

Tafilah	6.0177e-06	166	8.14 hrs.

According to ~~table~~ Table 14, the proposed model achieves high prediction accuracy for all Jordanian governorates, as the MSE values for all governorates are close to each other and less than 1%. In addition, our proposed model achieves the best performance with Tafilah, it achieves MSE by 6.0177e-06. Maan has the second best MSE of 7.1198e-06. Then, it is followed by Ajloun, Aqaba, Mafraq, Karak, Madaba, Irbid, Amman, Zarqa, and Jerash. While Balqa achieves the lowest performance with MSE of 4.1259e-05.

5.2.2 ~~The Influence of~~ Taking Governorates Surrounding the Target Governorate ~~on the Prediction Accuracy~~

In this experiment, we have employed the transfer learning ~~in order to~~to predict the weather conditions in Amman by training the proposed model in two stages. In the first stage, the model was trained on the whole dataset of all Jordanian governorates, then saving the best weight of the trained model. In the second stage, the best obtained weight was utilized to train the model on Amman dataset. Table 15 exhibits the results of weather forecasting in Amman over the next 24h when taking other governorates into account during training the model. We call this experiment All-Gov-Inp.

Table 15. The result of weather forecasting in Amman when taking other governorates into consideration during training the model.

Region	MSE	Number of Epochs	Execution Time
Amman Only	3.4811e-05	72	3.69 hrs.
Amman with Other Governorates	5.2378e-06	51	2.25 hrs.

Formatted Table

The results shown in ~~table~~ Table 15 emphasize that there is a noticeable improvement in the performance of the proposed model when neighbouring governorates are included in training. In fact, training the model twice, one for all governorates and the other for each governorate individually improves the model performance significantly. This is because the model can see and process the data twice.

5.4.3 ~~The Influence of~~ Taking Governorates Surrounding the Target Governorate and Some Areas Surrounding Jordan ~~on the Prediction Accuracy~~

Since the weather systems can move long distances over time, the weather of a certain area is highly influenced by that of the near areas. In fact, using the weather dataset of some areas surrounding Jordan that have different geographical locations may allow the proposed model to learn more patterns, and thus improving the accuracy of the predictions. In this experiment, we have trained the proposed model on all Jordanian governorates with some areas around Jordan including Damascus, Cyprus, Jerusalem, and Cairo to show the effect of taking areas surrounding Jordan on the prediction accuracy. The model was trained in two stages, the first one on the whole dataset, while the second one on Amman dataset. Table 16 shows the effect of taking areas surrounding Jordan on the prediction accuracy for Amman. We call this experiment All Gov- Some Areas-Inp.

Table 16. The result of weather forecasting in Amman when taking other governorates and some areas surrounding Jordan into consideration during training the model.

Region	MSE	Number of Epochs	Execution Time
Amman Only	3.4811e-05	72	3.69 hrs.
Amman with Other Governates	5.2378e-06	51	2.25 hrs.
Amman with Some Areas Surround Jordan	2.3304e-06	79	3.73 hrs.

Formatted Table

According to ~~table~~ Table 16, the MSE values decreases further when incorporating some areas surrounding Jordan in weather forecasting for Amman. When comparing the last three experiments, Each-Gov-Inp, All-Gov-Inp, and All Gov- Some Areas-Inp, we can notice that the All Gov- Some Areas-Inp model provides relatively the best results compared to other two models, it achieves MSE by 2.3304e-06. While All-Gov-Inp model achieves the second-best performance with MSE of 5.2378e-06. Whereas Each-Gov-Inp model achieves the highest MSE value of 3.4811e-05. Overall, taking the information of all Jordanian governorates and some areas surrounding Jordan gives better performance than taking all Jordanian governorates only. Also, considering all Jordanian governorates when making prediction for Amman gives better prediction accuracy than taking Amman only.

5.5 Visualization of Results

Visualizing the results is an important step in every ML framework to show the ability of proposed model in performing the task. In fact, we have taken a random sample from the test set of Amman and made a prediction for it by three models, Each-Gov-Inp, All-Gov-Inp, and All Gov- Some Areas-Inp, to show the improvement in prediction accuracy by each model. The sample consists of previous 120 hours of all-weather variables as input data, and next 24 hours as predicted output. Figures 20, 21, and 22 show a comparison between the actual and predicted values for each weather parameter using the three models: Each-Gov-Inp, All-Gov-Inp, and All Gov- Some Areas-Inp.

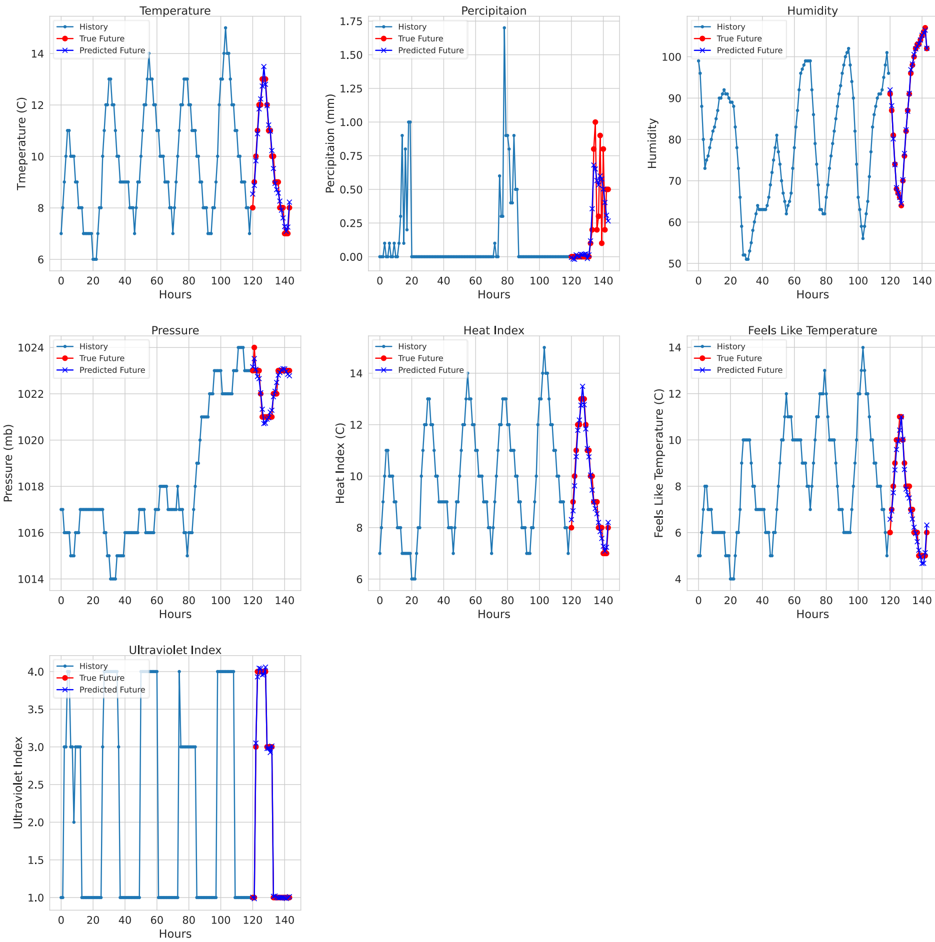

Figure 20. Actual versus predicted values of each weather parameter of a random sample using Each-Gov-Inp model.

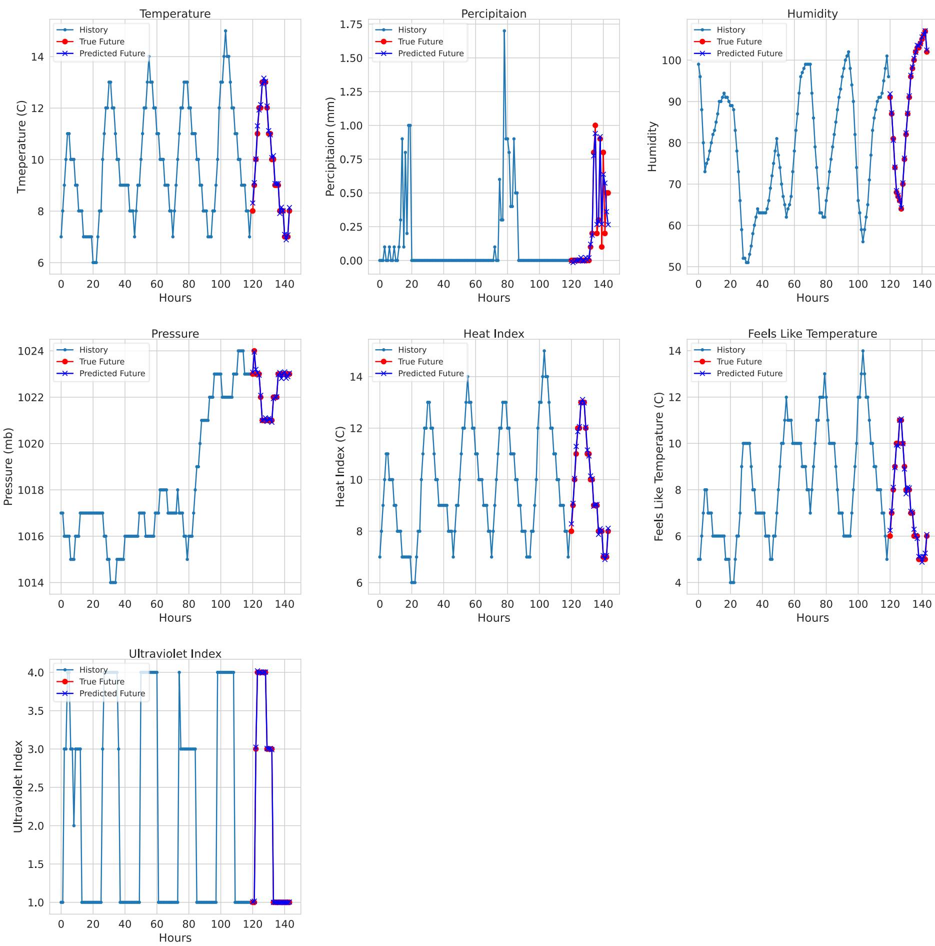

Figure 21. Actual versus predicted values of each weather parameter of a random sample using All-Gov-Inp model.

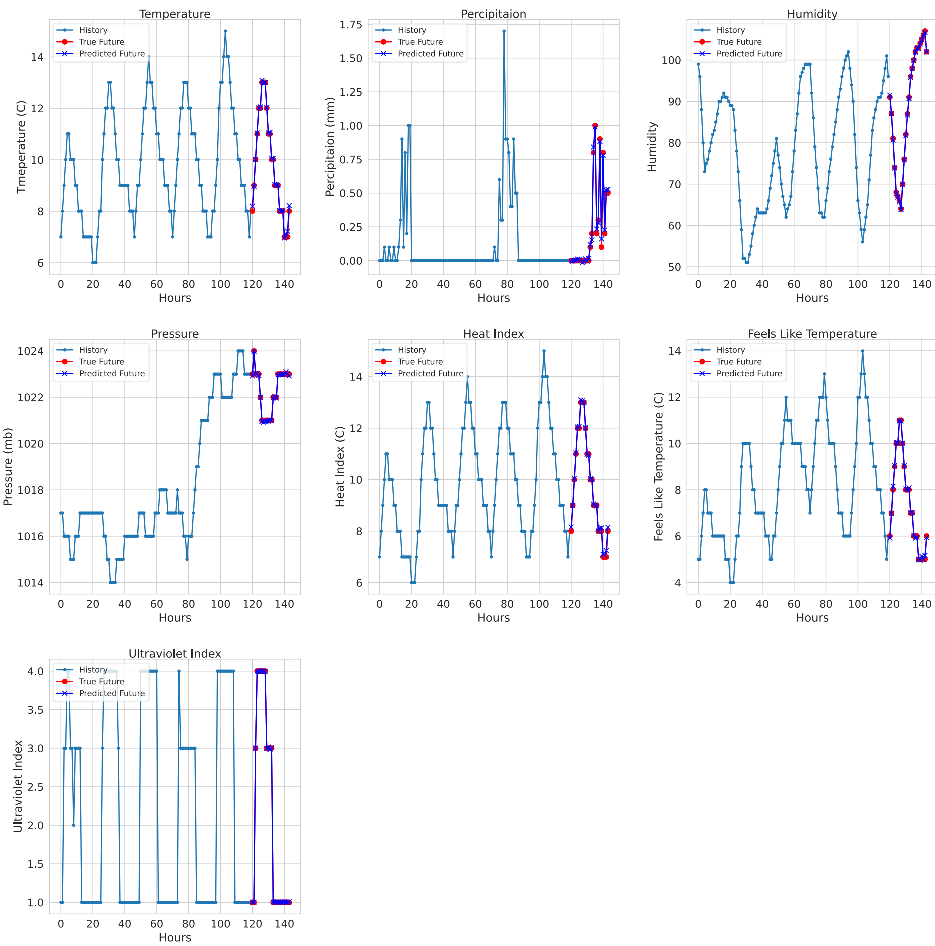

Figure 22. Actual versus predicted values of each weather parameter of a random sample using All Gov-Some Areas-Inp model.

It can be seen from the above three figures the enhancements in the prediction accuracy when implementing each type of model, Each-Gov-Inp, All-Gov-Inp and All Gov-Some Areas-Inp. The red line, predicted values, for all three models is closer to the blue line, actual values. The difference is a quite small. The improvement in performance between the three models can be easily observed in all-weather parameters, especially precipitation, as there is a small difference between the actual and predicted precipitation values by Each-Gov-Inp model as shown in figure 20. This difference decreases slightly when applying All-Gov-Inp model as illustrate in ~~figure~~ Figure 21. Whereas the actual values almost fully match the predicted values when utilizing All Gov-Some Areas-Inp model as shown in ~~figure~~ Figure 22. However, all three models show high effectiveness in predicting the weather conditions in Jordan for the next 24h. In fact, All Gov-Some Areas-Inp model proves superior performance in weather forecasting than All-Gov-Inp and Each-Gov-Inp models. Whereas, Each-Gov-Inp model gives the highest MSE value among the three models.

Chapter 6: Conclusions and Future Works

This chapter includes conclusions of my results and future works to improve the prediction accuracy.

6.1 Conclusions

In this thesis, a DL model based on ED-BiLSTM was proposed for a short-term weather forecasting in Jordan. The proposed model achieved substantial results with high accuracy in predicting the weather conditions. The datasets of all Jordanian governorates and some areas near Jordan were used in building the proposed model. The dataset consists of hourly readings of seven weather variables: average air temperature, ultraviolet index, atmospheric pressure, feels-like temperature, humidity, heat index, and total precipitation from 2009 to 2021.

We have carried out several experiments to enhance the base model including, changing the optimizers, type of cell, number of hidden layers, number of nodes, normalization techniques, datatypes, and size of encoder and decoder. We noticed that changing the optimizers and normalization techniques have the greatest impact on improving the model performance. Furthermore, the obtained results demonstrated that utilizing the historical data from surrounding areas to predict the weather of a particular area significantly enhance the prediction accuracy rather than just looking at the area where the weather forecasting is done.

6.2 Future Works

Future works for my thesis includes investigating the effect of adding more weather variables such as wind speed, visibility, and dew point on the prediction accuracy. Also, we suggest experimenting other types of DL architecture like CNN to check if they can produce better results than RNN. In addition to that, we suggest incorporating more locations around the target to improve the prediction accuracy.

REFERENCES

Althelaya, K. A., El-Alfy, E. S. M., & Mohammed, S. (2018, April). Stock market forecast using multivariate analysis with bidirectional and stacked (LSTM, GRU). **In 2018 21st Saudi Computer Society National Computer Conference (NCC)** (pp. 1-7). IEEE.

Anjali, T., Chandini, K., Anoop, K., & Lajish, V. L. (2019, July). Temperature prediction using machine learning approaches. **In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)** (Vol. 1, pp. 1264-1268). IEEE.

Arabeyyat, O., Shatnawi, N., & Matouq, M. (2018). Nonlinear multivariate rainfall prediction in Jordan using NARX-ANN model with GIS techniques. **Jordan Journal of Civil Engineering**, 12(3).

Azzouni, A., & Pujolle, G. (2017). A long short-term memory recurrent neural network framework for network traffic matrix prediction. **arXiv preprint arXiv**:1705.05690.

Booz, J., Yu, W., Xu, G., Griffith, D., & Golmie, N. (2019, February). A deep learning-based weather forecast system for data volume and recency analysis. **In 2019 International Conference on Computing, Networking and Communications (ICNC)** (pp. 697-701). IEEE.

Chhetri, M., Kumar, S., Pratim Roy, P., & Kim, B. G. (2020). Deep BLSTM-GRU model for monthly rainfall prediction: A case study of Simtokha, Bhutan. **Remote sensing**, 12(19), 3174.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. **arXiv preprint arXiv**:1406.1078.

Dahamsheh, A. M., & Suleiman, G. (2020). PREDICTION OF MONTHLY RAINFALL IN JORDAN VALLEY USING ARTIFICIAL NEURAL NETWORK**. INTERNATIONAL JOURNAL OF CIVIL ENGINEERING AND TECHNOLOGY (IJCIET)**, 11(10). https://doi.org/10.34218/IJCIET.11.10.2020.004

Doroshenko, A., Shpyg, V., & Kushnirenko, R. (2020, November). Machine Learning to Improve Numerical Weather Forecasting**. In 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT)** (pp. 353-356). IEEE.

Graves, A., & Schmidhuber, J. (2005, July). Framewise phoneme classification with bidirectional LSTM networks. In Proceedings. **2005 IEEE International Joint Conference on Neural Networks**, 2005. (Vol. 4, pp. 2047-2052). IEEE.

Hewage, P., Trovati, M., Pereira, E., & Behera, A. (2021). Deep learning-based effective fine-grained weather forecasting model. **Pattern Analysis and Applications**, 24(1), 343-366.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. **Neural computation**, 9(8), 1735-1780.

Hossain, M., Rekabdar, B., Louis, S. J., & Dascalu, S. (2015, July). Forecasting the weather of Nevada: A deep learning approach. **In 2015 international joint conference on neural networks (IJCNN)** (pp. 1-6). IEEE.

Jaseena, K. U., & Kovoor, B. C. (2020). A hybrid wind speed forecasting model using stacked autoencoder and LSTM. **Journal of Renewable and Sustainable Energy**, 12(2), 023302.

Jierula, A., Wang, S., Oh, T. M., & Wang, P. (2021). Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data. **Applied Sciences**, 11(5), 2314.

Karevan, Z., & Suykens, J. A. (2020). Transductive LSTM for time-series prediction: An application to weather forecasting. **Neural Networks**, 125, 1-9.

Khan, M. I., & Maity, R. (2020). Hybrid deep learning approach for multi-step-ahead daily rainfall prediction using GCM simulations. **IEEE Access**, 8, 52774-52784.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). LSTM-based encoder-decoder for multi-sensor anomaly detection. **arXiv preprint arXiv**:1607.00148.

Odeh, I. (2016). Temperature Prediction in Jordan using ANN. **International Journal of Computer Science and Information Technologies**, 7(1), 378-383.

Rahayu, I. S., Djamal, E. C., Ilyas, R., & Bon, A. T. (2020). Daily temperature prediction using recurrent neural networks and long-short term memory. **In Proceedings of the International Conference on Industrial Engineering and Operations Management** (pp. 2700-2709).

Rani, K. G., Wise, D. J. W., Begum, S. S., & Nirosha, S. (2020, July). Designing A Model for Weather Forecasting Using Machine Learning. **In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)** (pp. 385-388). IEEE.

Roy, D. S. (2020). Forecasting the air temperature at a weather station using deep neural networks. **Procedia Computer Science**, 178, 38-46.

Sarker, I. H. (2021). Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. **SN Computer Science**, 2(6), 1-20.

Schmidt, R. M. (2019). Recurrent neural networks (rnns): A gentle introduction and overview. **arXiv preprint arXiv**:1912.05911.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. **IEEE transactions on Signal Processing**, 45(11), 2673-2681.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019, December). The performance of LSTM and BiLSTM in forecasting time series. **In 2019 IEEE International Conference on Big Data (Big Data)** (pp. 3285-3292). IEEE.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. **Advances in neural information processing systems**, 27.

Zahran, B., Mesleh, A., Matouq, M., Alheyasat, O., & Alwada'n, T. (2015). Rainfall prediction in semi-arid regions in Jordan using back propagation neural networks**. International Journal on Engineering Applications**, 3(6), 158-162.

Zaytar, M. A., & El Amrani, C. (2016). Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. **International Journal of Computer Applications**, 143(11), 7-11.

التنبؤ بحالة الطقس في الأردن باستخدام تقنيات تعلم الآلة

إعداد

ليث عوده بني خالد

المشرف

عبندة الأستاذ الدكتور غيث علي ~~عبنده~~

ملخــــــص

يعد التنبؤ بالطقس مجالًا بحثيًا مهمًا نظرًا لتأثيره في مجموعة متنوعة من ~~التطبيقات~~ المجالات مثل الزراعة والصناعة والملاحة والسياحة والانشطة البشرية اليومية. تعتمد الطريقة التقليدية للتنبؤ بالطقس على نماذج فيزيائية معقدة لوصف السلوك الهيدروديناميكي للغلاف الجوي. هذه الطريقة مكلفة، وتستغرق الكثير من الوقت، وغالبًا ما تكون غير دقيقة وتتطلب ~~اجهزة~~ أجهزة كمبيوتر عملاقة لعمل التنبؤات. في الواقع، يمكن أن يؤدي توظيف خوارزميات التعلم الآلي في التنبؤ بالطقس إلى زيادة دقة التنبؤ وتوفير طريقة تكميلية للطريقة التقليدية للتنبؤ بالطقس. يتم استخدام الشبكات العصبية المتكررة (RNN) و الذاكرة طويلة المدى ثنائية الاتجاه (BiLSTM) على نطاق واسع لحل المشاكل التي تحتوي على سلاسل ~~الزمنيه~~ زمنية وانماط غير خطية.

في هذه الرسالة، اقترحنا نموذج Encoder-Decoder BiLSTM للتنبؤ بحالة الطقس على المدى القصير في الأردن. تظهر نتائج التقييم أن النموذج المقترح يعطي نتائج دقيقة للغاية مع نسبة خطأ منخفضة، أقل من 1٪. كما أثبتنا ايضا أهمية دمج مجموعة بيانات المواقع القريبة من الهدف على ~~اداء~~ أداء النموذج، حيث لاحظنا تحسن كبير في ~~اداء~~ أداء النموذج عند استخدام بيانات الطقس لجميع المحافظات الأردنية وبعض المدن المحيطة بالأردن للتنبؤ بحالة الطقس في عمان.

تم استخدام مجموعة بيانات الطقس لجميع المحافظات الأردنية وبعض المناطق ~~القريبه~~ القريبة من الأردن وهي دمشق وقبرص والقدس والقاهرة في بناء واختبار النموذج المقترح. تتضمن معلمات الطقس التي تم استخدامها في هذه الأطروحة قراءات كل ~~ساعة لمتوسط~~ ساعة لمتوسط درجة حرارة الهواء، ومؤشر الأشعة فوق البنفسجية، والضغط الجوي، ودرجة الحرارة الشبيهة بالشعور، والرطوبة، ومؤشر الحرارة، وإجمالي هطول الأمطار من عام 2009 إلى عام 2021.