

Forecasting Point Observation of Temperature and CO₂ with RNN and LSTM Models

Master Thesis

A thesis submitted in fulfilment of the requirements for the academic degree of
Master of Science (M.Sc.) in Meteorology
at the Institute for Atmospheric and Environmental Sciences
Johann Wolfgang Goethe University, Frankfurt am Main

submitted by
Onurhan Aydin
from Frankfurt am Main

submitted on 16. August 2024

Supervisor: Prof. Dr. Juerg Schmidli

Co-supervisor: Dr. Shweta Singh

Declaration

I hereby confirm that I have independently written the present thesis and have not used any sources other than those listed in the bibliography. I have adhered to the rules of good scientific practice.

All passages that are quoted or paraphrased from published or unpublished sources are clearly identified as such.

The drawings or illustrations in this thesis have been created by me or are provided with an appropriate source reference.

This thesis has not been submitted in the same or a similar form to any other examination authority.

Frankfurt am Main, den 16 August 2024

Onurhan AYDIN

Acknowledgment

I would like to express my deepest gratitude to my advisors, Prof. Dr. Juerg Schmidli and Dr. Shweta Singh, for their continuous support, guidance, and encouragement throughout the duration of this thesis. Their insights and expertise were invaluable in shaping my research and bringing this work to completion.

I am also grateful to my professors and instructors at Johann Wolfgang Goethe University for their valuable teachings and support during my studies. Their knowledge and dedication have greatly contributed to my academic growth.

Special thanks go to my family and friends for their unwavering support and encouragement during my studies. Their patience and understanding provided me with the strength and motivation needed to complete this work.

I would also like to acknowledge the support of my colleagues and fellow students at Johann Wolfgang Goethe University who offered helpful discussions and shared their knowledge and experiences with me.

Lastly, I thank all those who have contributed to my academic journey, whether directly or indirectly. Your support has been instrumental in helping me achieve this milestone.

Table of Contents

1 Introduction.....	8
2 Review Literatures.....	10
2.1 Physical methods for weather forecasting.....	10
2.2 Statistical methods for weather forecasting.....	12
2.2.1 Machine Learning.....	14
2.2.2 Deep Learning	16
3 Neural Networks	19
3.1 Artificial Neural Networks	19
3.2 Recurrent Neural Network.....	20
3.3 Long Short-Term Memory	21
4 Methodology.....	24
4.1 Data collection and site	24
4.2 Data Analysis and Visualisation.....	25
4.3 Model setup.....	32
4.4 Performance evaluation	36
5 Results	38
5.1 Model sensitivity	38
5.2 Sensitivity Parameters	39
5.2.1 Neuron and Features.....	39
5.2.2 Layer.....	47
5.2.3 Scaler	51
5.2.4 Activation function.....	55
5.3 Loss Metrics for Overfitting.....	60
5.3.1 Temperature.....	60
5.3.2 CO ₂	63
6 Conclusion.....	66
7 References	68
8 Appendix.....	73
8.1 Forecast Results of Temperature	73
8.1.1 LSTM	73
8.1.2 RNN.....	83
8.2 Forecast Results of CO₂	93
8.2.1 LSTM	94
8.2.2 RNN.....	98
8.3 Tools and Literature	103

List of Figures

Figure 1. The mathematical and physical laws that govern the interactions of elements advance in time, updating each block's temperature, pressure, and other variables (Arizona University. Accessed: Jan. 10, 2024).....	11
Figure 2. The diagram represents the practical process of machine learning, illustrating the systematic approach in computational learning methodologies (Chen et al., 2020)	15
Figure 3. The figure depicts the architecture of a Deep Learning model designed for sequential data, consisting of interconnected input, hidden, and output layers (Mishra et al., 2018).	18
Figure 4. The figure depicts the architecture of a Recurrent Neural Network (RNN) for prediction, with input, hidden state, and output layers for analyzing sequential data (Tran et al., 2021).	21
Figure 5. The LSTM cell architecture, crucial for air temperature forecasting, includes cell states, gates, and activation functions to capture temporal dependencies (Tran et al., 2021).	23
Figure 6. Geographical overview of Payerne, Switzerland, marked with a blue circle, showing key geographical features from a satellite image (Google Earth. Accessed: Jan. 24, 2024)...	24
Figure 7. Heatmap representation of correlations between weather variables, with darker colors indicating stronger relationships.	27
Figure 8. Boxplots of variables depicting variability and central tendencies, with taller boxes signaling greater variability and outliers marked.....	28
Figure 9. Time series plots of temperature, CO ₂ , and correlated variables, showing distinct diurnal cycles and recurring patterns over 24 hours.	29
Figure 10. Wind rose diagrams for Payerne station, illustrating the distribution of wind speed and direction for each month.	30
Figure 11. Internal structure of an LSTM unit, highlighting the roles of the forget, input, and output gates in regulating information flow (Jiang et al., 2019).....	32
Figure 12. Graphical representation of MAE for the sensitivity test results of the layers for temperature.	48
Figure 13. Evaluation of the impact of varying the number of layers on the MAE for CO ₂ forecasting, showing better performance with simpler architectures.	50
Figure 14. Comparison of the MAE for temperature predictions using different data scaling technics, highlighting MinMaxScaler's effectiveness.	53

Figure 15. Comparison of the MAE for CO ₂ predictions using different scaling techniques, highlighting MinMaxScaler's superior performance.	54
Figure 16. Evaluation of the impact of different activation functions on the MAE for temperature forecasting.....	57
Figure 17.Evaluation of how different activation functions affect the MAE for CO ₂ forecasting, indicating varying optimal functions for LSTM and RNN models.	59
Figure 18. Training and validation loss over epochs for the LSTM model, illustrating the model's learning and potential signs of overfitting in 14 times running.....	61
Figure 19. Training and validation loss over epochs for the RNN model in 11 times running, showing the model's learning process and potential overfitting.	62
Figure 20. Training and validation loss over epochs for the LSTM model in 13 times running, illustrating the model's learning and potential signs of overfitting for CO ₂	64
Figure 21. Training and validation loss over epochs for the RNN model in 14 times running, showing the model's learning process and potential overfitting for CO ₂	65

List of Tables

Table 1. Available meteorological and trace gas	25
Table 2. Architectures of LSTM and RNN models. The boxes represent the architecture of LSTM for temperature and CO ₂ and the architecture of RNN for temperature and CO ₂ respectively	35
Table 3. Impact of varying the number of features and neurons on the MAE for temperature prediction	42
Table 4. Impact of varying the number of features and neurons on the MAE for CO ₂ prediction	45
Table 5. Layer sensitivity results for temperature	48
Table 6. Layer sensitivity results for CO ₂	49
Table 7. Temperature forecasting MAE with different scalers.....	52
Table 8. MAE for CO ₂ forecasting with different scalers	54
Table 9. Sensitivity results of activation function for temperature.....	57
Table 10. Sensitivity results of activation function for CO ₂	58
Table 11. Final sensitivity setup for temperature.....	61
Table 12. Final sensitivity setup for CO ₂	63

1 Introduction

Weather forecasting is crucial for many areas such as farming, transportation, energy management, and disaster preparedness. Accurate predictions help make better decisions and reduce risks [1]. Among the weather factors, temperature and carbon dioxide (CO₂) levels are particularly important as they affect many human activities and environmental processes [2]. Despite advances in weather science and technology, predicting temperature and CO₂ accurately remains challenging due to the complex interactions in the atmosphere and factors like land and ocean conditions [2].

Traditional forecasting methods, such as numerical weather prediction models and statistical techniques, have long been used [3]. However, these methods often struggle to capture the complex patterns in temperature and CO₂ data over time. They also have difficulties modeling non-linear relationships, handling data uncertainty, and providing accurate forecasts across different periods. They require significant computing power, which can be a limitation for real-time forecasting [3].

Recently, deep learning techniques, especially Long Short-Term Memory (LSTM) networks, have brought new possibilities to weather forecasting. These methods can capture complex patterns in data and learn from past information [4]. LSTM networks, a type of recurrent neural network (RNN), are particularly good at handling sequential data and time series forecasting tasks [5]. Unlike traditional methods, LSTM models can automatically find important features in raw data, adapt to changing patterns, and make accurate long-term predictions. This makes LSTM networks especially useful for forecasting temperature levels in areas with complex climate conditions and rapid weather changes [6].

Machine learning (ML) and deep learning (DL) techniques have significantly improved weather forecasting. Research by Scher and Messori [6] has shown that algorithms can improve short-term precipitation forecasts. These algorithms are good at capturing complex relationships between weather variables. Deep learning techniques, especially RNNs and LSTM networks, have shown great potential in understanding complex non-linear relationships in atmospheric data, which is crucial for accurate weather forecasting.

The main goal of this thesis is to study how effective LSTM and RNN models are in forecasting temperature and CO₂ over time and to see if they can improve forecast accuracy and lead time. Specifically, we used the last five days of data to predict the weather for the next day. For example, we used data from August 21, 22, 23, 24, and 25 to predict the weather for August 26 (see the appendix). By using historical temperature, CO₂ and additional meteorological data, weather predictors, and advanced deep learning techniques, this research aims to address key challenges in weather forecasting, such as model understanding, data processing, and adjusting parameters.

To optimize the performance of these models, we performed sensitivity tests by adjusting several parameters such as the number of neurons, the number of features, the number of layers, the type of scaler, and the activation function. These tests help to find the best configuration for the models to improve their accuracy and reliability. Through detailed analysis, experiments, and validation against standard datasets, this study seeks to provide insights into the strengths and limitations of LSTM-based approaches for forecasting temperature and CO₂.

In conclusion, this thesis aims to connect advanced deep learning methods with practical weather forecasting applications. By showing the strengths and addressing the weaknesses of LSTM and RNN models, this study aims to develop more robust and reliable forecasting systems that can better serve society's needs in mitigating the effects of adverse weather conditions.

2 Review Literatures

2.1 Physical methods for weather forecasting

Physical weather forecasting methods, particularly Numerical Weather Prediction (NWP), form the backbone of modern meteorological modeling, employing physics and mathematics to simulate complex atmospheric processes [3]. NWP models, grounded in fluid dynamics and thermodynamics, aim to replicate the behavior of the atmosphere by integrating partial differential equations, notably the Navier-Stokes equations [3]. These equations govern fundamental fluid motion principles, allowing for the representation of dynamic processes such as advection, diffusion, and the conservation of mass, momentum, and energy. Early pioneers in the field, such as Lorenz and Charney et al., laid the groundwork for NWP, emphasizing the significance of numerical methods in approximating atmospheric physics and developing initial models capable of simulating atmospheric dynamics [8], [9].

Advancements in computational techniques have driven the evolution of NWP models, enabling meteorologists to achieve higher spatial and temporal resolutions [10]. Modern NWP models, exemplified by those developed by institutions like the European Centre for Medium-Range Weather Forecasts (ECMWF) and the National Center for Atmospheric Research (NCAR), employ sophisticated grid systems such as spectral transform or finite-volume grids to enhance spatial resolution and capture smaller-scale atmospheric features [11][12].

In addition to numerical techniques, physical forecasting methods integrate observational data into model simulations through data assimilation [13]. Data assimilation involves the continuous integration of observations from various sources, including satellites, radar, and ground-based instruments, into NWP models to initialize and update their simulations. By assimilating observational data, NWP models can correct errors and biases in initial conditions, leading to improved forecast accuracy over time. The synergy between advanced numerical techniques, adherence to physical principles, and the integration of observational data assimilation constitutes a comprehensive approach to weather forecasting, empowering meteorologists to better understand and predict complex atmospheric phenomena.

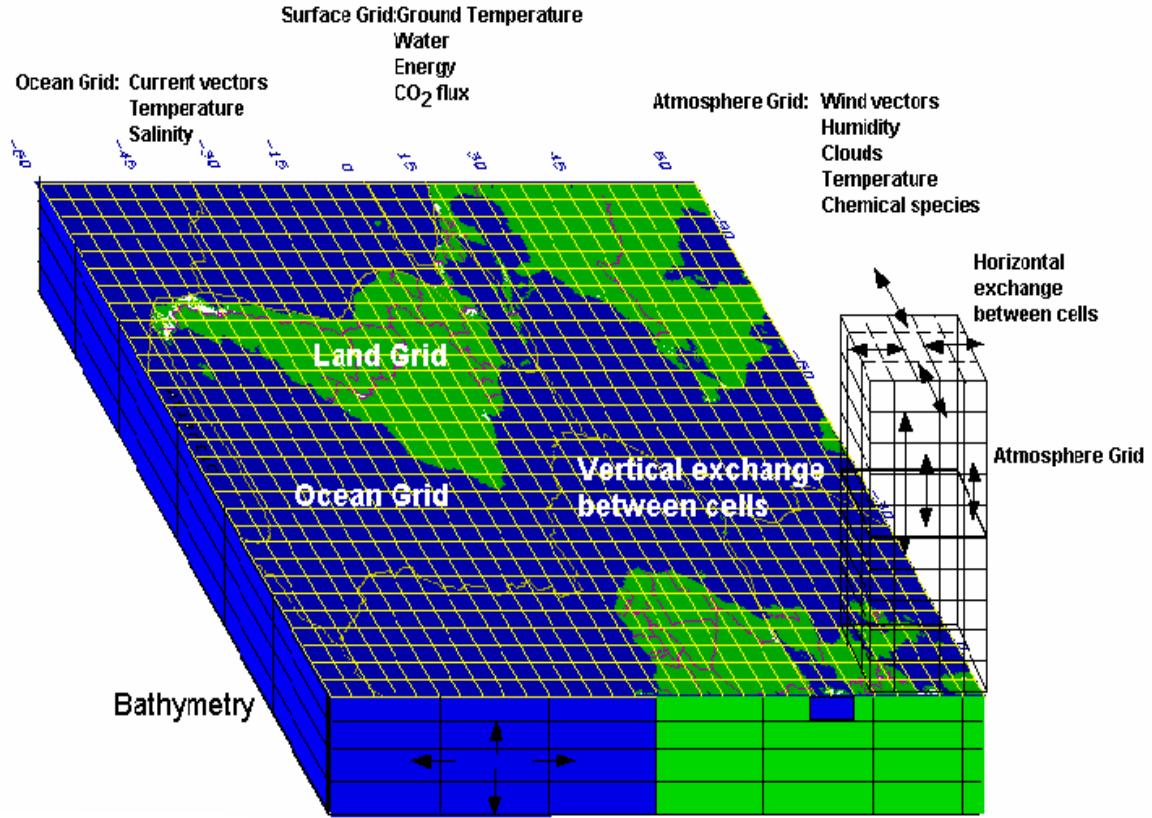


Figure 1. The mathematical and physical laws that govern the interactions of elements advance in time, updating each block's temperature, pressure, and other variables (Arizona University. Accessed: Jan. 10, 2024).

Additionally, Figure 1 obtained from the website of Arizona University [14] shows how a model works by specifying all these numbers for every block in the model. This is the initial condition of the model and defines the state of the model at the starting time. From this point, the model operates autonomously. The mathematical and physical principles that dictate the interactions between elements are projected forward over time. Essentially, we compute the changes in temperature, pressure, and other variables for each segment due to significant physical processes, including the effects of adjacent segments. After completing these calculations, the model is slightly altered from its initial state. Each segment has updated values for temperature, pressure, density, humidity, wind direction, speed, and other relevant parameters.

2.2 Statistical methods for weather forecasting

The integration of statistical methods into weather forecasting has historically relied on a variety of techniques that model the relationships between different meteorological variables. These methods have evolved significantly with advancements in computing power and the advent of more sophisticated algorithms. Traditional statistical methods include regression analysis, autoregressive integrated moving average (ARIMA) models, and ensemble techniques, which have been foundational in weather prediction efforts.

Regression analysis, one of the most basic statistical techniques, involves identifying the relationships between a dependent variable, such as temperature, and one or more independent variables, such as humidity or wind speed. This method has been widely used due to its simplicity and the ease of interpreting its results. However, regression analysis often falls short in capturing the non-linear relationships present in meteorological data. For instance, non-linear interactions between temperature and humidity may not be effectively modelled using simple regression techniques, necessitating more sophisticated approaches to improve forecast accuracy [2].

ARIMA models extend the capabilities of regression by incorporating elements of autoregression (AR) and moving averages (MA) to address time series data. These models are particularly effective in short-term forecasting and have been used extensively in predicting weather patterns. ARIMA models are capable of dealing with the stationary aspects of time series data while also integrating the most recent observations with long-term historical trends, thus providing more accurate and reliable forecasts. Studies have shown that ARIMA models can outperform other statistical techniques in providing interval forecasts and are more reliable in capturing the temporal correlation and skewness inherent in meteorological variables [14].

Ensemble techniques, which combine multiple models to improve predictive performance, have gained popularity in weather forecasting. Methods such as bagging and boosting aggregate the outputs of several models to create a more accurate and robust forecast. Ensemble methods are particularly useful in reducing the variance of predictions and improving forecast reliability [2].

The advent of machine learning (ML) and deep learning (DL) techniques has further revolutionized statistical methods in weather forecasting. ML algorithms, such as support vector machines (SVMs) and random forests, have been employed to refine short-term precipitation forecasts.

Deep learning techniques, particularly recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks, have shown significant promise in modeling temporal dependencies within weather data. LSTM networks, a specialized type of RNN, are designed to handle long-term dependencies and are particularly effective in time series forecasting tasks. These networks can capture the sequential nature of meteorological data, making them well-suited for predicting weather patterns that evolve over time. For instance, the study by Su et al. demonstrated the effectiveness of an object-based probabilistic deep learning model using multichannel infrared GOES-16 satellite observations to predict convective initiation, highlighting their precision in capturing complex atmospheric phenomena [19].

Furthermore, deep learning has also improved the understanding of extreme weather events. For example, Bose et al. [17] used recurrent neural networks (RNNs) to analyze satellite imagery and identify patterns associated with the formation and intensity of hurricanes. By processing sequential data from satellite images, these networks can discern subtle atmospheric cues indicative of impending extreme weather phenomena, thereby improving the lead time and accuracy of forecasts. This approach leverages the vast amount of data available from modern satellites, which can sense multiple spectral bands, to provide a more detailed and accurate understanding of hurricane behavior [17].

The collaboration between statistical methods and machine learning holds promising potential for advancing our comprehension of atmospheric dynamics and elevating the precision of weather forecasts. By leveraging the complementary strengths of traditional statistical methods and modern ML and DL techniques, meteorologists can harness the wealth of available data to generate more accurate and reliable predictions. This integration not only enhances forecast accuracy but also aids in mitigating the impacts of adverse weather conditions, ultimately contributing to greater societal resilience.

In summary, while traditional statistical methods have provided a strong foundation for weather forecasting, the incorporation of machine learning and deep learning techniques has

significantly enhanced the accuracy and reliability of predictions. As these technologies continue to evolve, their integration into operational weather forecasting models promises to deliver even greater advancements in predictive capabilities, ultimately improving our ability to respond to and prepare for weather-related risks.

2.2.1 Machine Learning

Machine learning (ML) has emerged as a transformative force revolutionizing the landscape of weather forecasting, offering innovative solutions to formidable challenges in predicting atmospheric phenomena. The procedural steps of machine learning techniques, as depicted in Figure 2, can be succinctly outlined as follows: (1) Identify a problem involving an unknown mapping function f and define a hypothetical solving model set H [20]. (2) Gather and organize a finite training set D . (3) Specify the loss function for the model. (4) Choose the learning algorithm A . (5) Optimize the parameters to minimize the loss function, resulting in the selection of optimal parameters for the model. (6) Save the model g with the optimal parameters and utilize it for prediction and analysis of new data.

Machine learning algorithms are categorized based on learning tasks such as prediction, feature selection, and dimensionality reduction. In the context of this review focusing on tropical cyclone forecast modeling, only predictive algorithms will be discussed. Typically, if the model aims to predict discrete values, it pertains to a "classification" task, while predicting continuous values corresponds to a "regression" task. Moreover, learning tasks can be broadly classified into "supervised learning" and "unsupervised learning," depending on whether training data are labeled. Classification and regression represent supervised learning, while clustering represents unsupervised learning.

The prediction task involves establishing a mapping f from the input space X to the output space Y , where $f: X \rightarrow Y$. This mapping function f depends on a vector of nonlinear parameters ω , where $y = f(x, \omega)$. The parameters ω are obtained during training, involving optimization of performance criteria (e.g., minimizing mean square error) for classification or regression/mapping problems. Additionally, machine learning algorithms may have hyperparameters, such as the number of hidden neurons and learning rates for neural networks,

which also impact the training process of an appropriate machine learning-based forecast model.

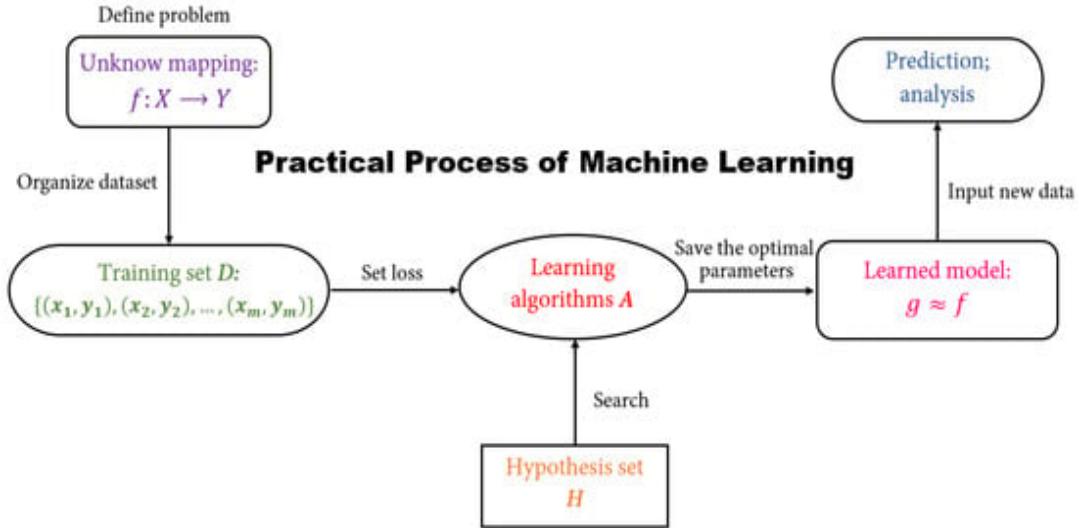


Figure 2. The diagram represents the practical process of machine learning, illustrating the systematic approach in computational learning methodologies (Chen et al., 2020).

Casati et al. [21] demonstrate the adaptability and effectiveness of various ML algorithms, each offering unique advantages in enhancing weather forecast accuracy.

Support Vector Machines (SVMs), a powerful supervised learning algorithm, have shown promising results in weather forecasting tasks. Research by Zhang et al. [22] highlights the effectiveness of SVMs in predicting precipitation patterns, leveraging their ability to delineate complex decision boundaries and handle high-dimensional data effectively.

K-Nearest Neighbors (KNN), another popular ML algorithm, has also found applications in weather forecasting. For instance, a study by Combinido et al. [23] showcased the utility of KNN in predicting localized weather phenomena, highlighting its simplicity and ability to identify patterns based on proximity to neighboring data points.

Ensemble learning techniques, such as Random Forests and Gradient Boosting Machines (GBMs), have gained traction in weather forecasting due to their ability to combine multiple models to improve prediction accuracy. Research by Benáček et al. [24] demonstrates the effectiveness of ensemble methods in forecasting temperature extremes, harnessing the

collective intelligence of diverse models to produce more robust predictions. Specifically, the study utilized natural gradient boosting and quantile random forests to adjust hourly temperature ensemble predictions, showing superior performance for short-term forecasts.

Decision Trees, known for their interpretability and ease of use, have also been applied in weather forecasting scenarios. Studies by Chen et al. [25] showcase the utility of decision trees in predicting wind speed and direction, offering valuable insights into localized weather patterns and facilitating decision-making processes.

Naive Bayes classifiers, while relatively simple, have found niche applications in weather forecasting, particularly in probabilistic forecasting tasks. Research by Liu et al. [26] explores the use of Naive Bayes classifiers in predicting the likelihood of specific weather events, providing valuable probabilistic forecasts essential for risk assessment and decision-making.

As machine learning techniques continue to evolve, their integration into operational weather forecasting models promises to usher in a new era of predictive accuracy. The versatility and adaptability of ML algorithms to dynamic meteorological conditions establish them as reliable tools for unraveling the intricacies of the Earth's atmosphere, thereby contributing to more resilient and responsive forecasting systems.

2.2.2 Deep Learning

Deep learning (DL) has emerged as a transformative force in weather forecasting, revolutionizing traditional approaches and significantly enhancing predictive capabilities, particularly in time series forecasting. DL encompasses various neural network architectures, including recurrent neural networks (RNNs) and their variant, long short-term memory networks (LSTMs), which have proven pivotal in handling sequential data. These models are particularly suitable for the complexities of time series forecasting in meteorology. RNNs and LSTMs are designed to capture temporal dependencies in data, allowing them to model the sequential nature of meteorological phenomena effectively [27]. This capability is crucial for accurately predicting weather patterns, which often depend on past data to forecast future conditions [28].

Moreover, DL techniques offer versatility and adaptability, allowing for the integration of additional data sources and the extraction of meaningful features for forecasting tasks. For instance, integrating data from satellite imagery, sensor networks, and historical weather records can significantly enhance the models' forecasting accuracy. This multi-source data integration allows DL models to learn from various inputs, providing a comprehensive view of the weather system and improving the precision of weather predictions [27], [29].

Attention-based mechanisms in DL models further enhance weather forecasting. These mechanisms empower models to focus on relevant information, improving their ability to identify and prioritize important features within the data. This selective focus is particularly beneficial in weather forecasting, where specific atmospheric patterns or anomalies can significantly impact the predictions. Attention mechanisms enable models to dynamically adjust their focus based on evolving weather patterns, leading to more accurate forecasts [29].

The advancements in DL have significant practical implications for weather forecasting. Enhanced predictive capabilities enable better disaster preparedness, allowing authorities to issue timely warnings and take preventive measures to mitigate the impact of extreme weather events. Accurate weather forecasts also support various sectors, such as agriculture, aviation, and energy, by providing critical information for decision-making processes. For example, precise weather predictions can help farmers plan their activities, optimize irrigation schedules, and protect crops from adverse weather conditions. The continuous improvement and integration of advanced DL techniques in weather forecasting promise a future where predictions are more reliable, timely, and actionable, ultimately contributing to the safety and well-being of communities worldwide.

In addition to the textual explanation, a visual representation of these concepts can be found in Figure 3, which illustrates the integration of attention-based mechanisms and transformer architectures with RNNs and LSTMs in weather forecasting models.

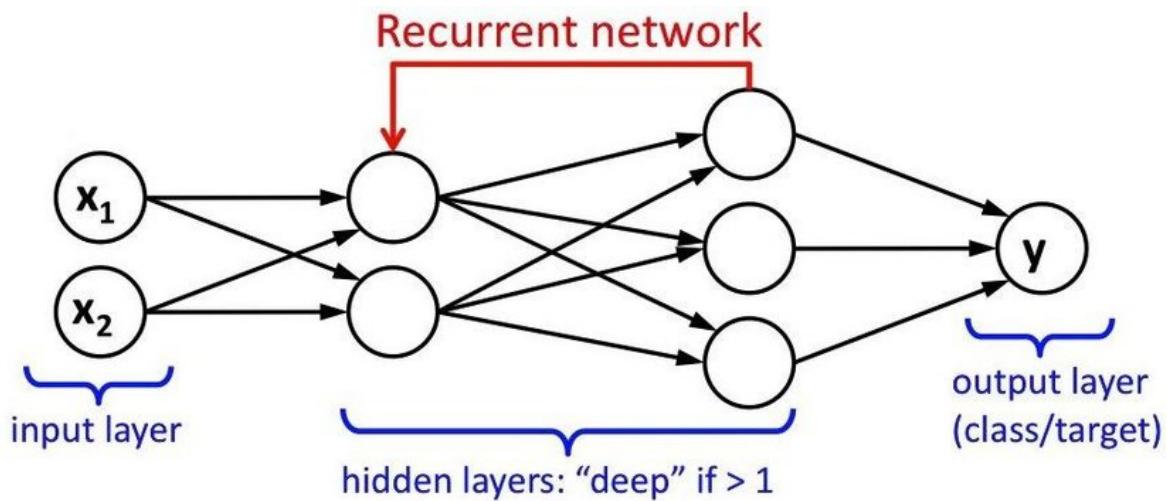


Figure 3. The figure depicts the architecture of a Deep Learning model designed for sequential data, consisting of interconnected input, hidden, and output layers (Mishra et al., 2018).

As DL methods continue to evolve, the potential for achieving unprecedented levels of accuracy in weather predictions grows. These advancements not only advance our scientific understanding of atmospheric processes but also have significant implications for practical applications such as disaster preparedness, climate modeling, and sustainable resource management. By leveraging DL models, forecasters can better anticipate and mitigate the impacts of extreme weather events, thereby enhancing societal resilience and fostering sustainable development in the face of dynamic and evolving weather patterns.

3 Neural Networks

3.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) represent a class of computational models inspired by the structure and function of biological neural networks, capable of learning complex patterns and relationships within data [31]. ANNs consist of interconnected nodes organized into layers, including input, hidden, and output layers, where each node performs simple computations and transmits signals to nodes in subsequent layers. These networks are trained using iterative optimization algorithms, such as backpropagation, to adjust the weights and biases of connections between nodes, minimizing the error between predicted and actual outputs.

The versatility and effectiveness of ANNs have been demonstrated across various domains, including pattern recognition, classification, and time series forecasting. They offer flexibility in modeling nonlinear relationships and can handle large volumes of data efficiently. Moreover, advancements in hardware and algorithmic techniques have fueled the development of deep neural networks, enabling the extraction of intricate features from raw data with unprecedented accuracy and speed [32]. These deep architectures, comprising multiple hidden layers, allow ANNs to automatically learn hierarchical representations of data, leading to superior performance in tasks such as image and speech recognition, natural language processing, and financial forecasting.

Furthermore, recent research explores the application of ANNs in climate modeling, showcasing their potential in capturing complex interactions between various meteorological variables and providing more accurate long-term projections. By leveraging the capabilities of ANNs, researchers can gain deeper insights into the dynamics of the Earth's climate system, facilitating better-informed decision-making in areas such as environmental policy and resource management.

As the field of artificial neural networks continues to evolve, ongoing advancements in algorithmic techniques, model architectures, and computational infrastructure promise to further enhance their capabilities and expand their applicability across diverse domains.

3.2 Recurrent Neural Network

Recurrent Neural Networks (RNNs) constitute a class of neural network architectures specifically designed to process sequential data by maintaining a form of memory across time steps [4]. Unlike traditional feedforward neural networks, RNNs possess cyclic connections, allowing them to exhibit dynamic temporal behavior by capturing dependencies within sequential data. These networks have found extensive applications in diverse fields such as natural language processing, speech recognition, and time series forecasting [33].

RNNs are particularly adept at modeling time-dependent data because they can maintain information about previous inputs through their recurrent connections. This makes them highly suitable for weather forecasting, where understanding the temporal relationships between data points is crucial. For example, RNNs have been used effectively to forecast atmospheric temperature by modeling the sequential dependencies in weather data [35].

The versatility of RNNs allows for the integration of additional data sources and the extraction of meaningful features for forecasting tasks. Integrating data from satellite imagery, sensor networks, and historical weather records can significantly enhance the forecasting accuracy of RNN models. This multi-source data integration allows RNNs to learn from various inputs, providing a comprehensive view of the weather system and improving the precision of weather predictions [34].

While RNNs are powerful, they can struggle with long-term dependencies due to issues like the vanishing gradient problem. Long Short-Term Memory (LSTM) networks, a variant of RNNs, address this issue by incorporating specialized memory cells that can retain relevant information over extended time horizons [4]. This makes LSTMs particularly effective in capturing long-range dependencies in weather data, further enhancing the accuracy of forecasts [35].

The advancements in RNN and LSTM research have significant practical implications for weather forecasting. Enhanced predictive capabilities enable better disaster preparedness, allowing authorities to issue timely warnings and take preventive measures to mitigate the impact of extreme weather events. Accurate weather forecasts also support various sectors,

such as agriculture, aviation, and energy, by providing critical information for decision-making processes. The continuous improvement and integration of advanced RNN and LSTM techniques in weather forecasting promise a future where predictions are more reliable, timely, and actionable, ultimately contributing to the safety and well-being of communities worldwide [33], [34].

In the context of this discussion, Figure 4 illustrates the integration of RNN in time series forecasting models, showcasing their role in capturing temporal dependencies and improving forecast accuracy. This figure is inspired by the comprehensive review conducted by Tran et al. [36], which provides valuable insights into the application of neural networks for weather forecasting.

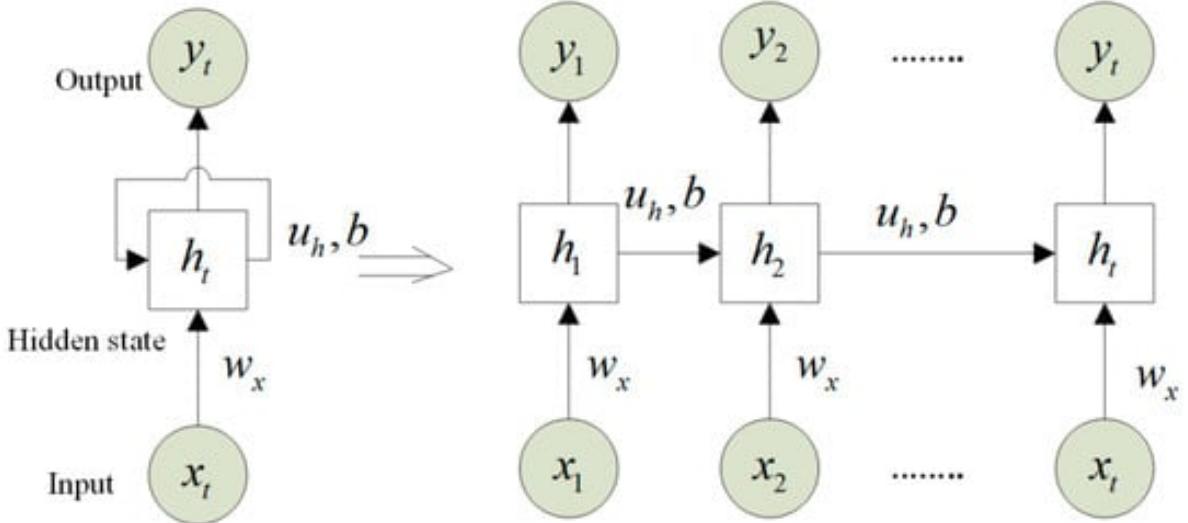


Figure 4. The figure depicts the architecture of a Recurrent Neural Network (RNN) for prediction, with input, hidden state, and output layers for analyzing sequential data (Tran et al., 2021).

As LSTM-based methodologies continue to evolve and adapt to new challenges, they hold great promise for revolutionizing time series forecasting across a wide range of applications.

3.3 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks have emerged as a cornerstone in time series forecasting due to their remarkable ability to capture and model complex temporal dependencies over extended sequences while alleviating the vanishing gradient problem

inherent in traditional Recurrent Neural Networks (RNNs) [4]. LSTMs are distinguished by their unique architecture featuring memory cells equipped with input, forget, and output gates, allowing for the selective retention or discarding of information over time. These gates facilitate the learning of both short-term patterns and long-term dependencies within sequential data, making LSTMs particularly well-suited for capturing the dynamic nature of time series datasets.

In meteorology, where accurate and reliable weather forecasting is crucial, LSTM networks have demonstrated significant potential. For instance, in temperature forecasting, LSTMs can effectively capture the intricate relationships between various meteorological variables such as humidity, air pressure, wind speed, and historical temperature data. By analyzing these complex interactions, LSTM models can provide valuable insights into temperature trends and fluctuations, aiding meteorologists in making informed decisions for a wide range of applications, including agriculture, energy management, and disaster preparedness.

Furthermore, the hierarchical architecture of LSTM networks enables them to learn abstract representations of sequential meteorological data, facilitating the extraction of meaningful features for forecasting tasks. This versatility and adaptability render LSTM networks highly effective for modeling complex and nonlinear relationships inherent in meteorological time series data, leading to superior forecasting performance compared to conventional statistical methods across a myriad of domains including climate prediction and weather forecasting.

Recent research has further underscored the effectiveness of LSTM networks in various time series forecasting tasks. For instance, the work by Malhotra et al. [7] demonstrated the application of LSTM networks for anomaly detection in time series data, showcasing their ability to capture subtle deviations from expected patterns. Similarly, Greff et al. [5] conducted a comprehensive exploration of LSTM architectures, highlighting their efficacy in handling diverse temporal sequences and providing insights into their underlying mechanisms.

Additionally, Figure 5 illustrates the structure of an LSTM cell, showcasing the internal mechanisms of input, forget, and output gates that enable the network to retain and utilize information over time. This figure is inspired by the comprehensive review conducted by Tran et al. [36], which provides valuable insights into the application of neural networks, including

LSTMs, for air temperature forecasting. Parameters in the LSTM cell is explained in Model Setup part.

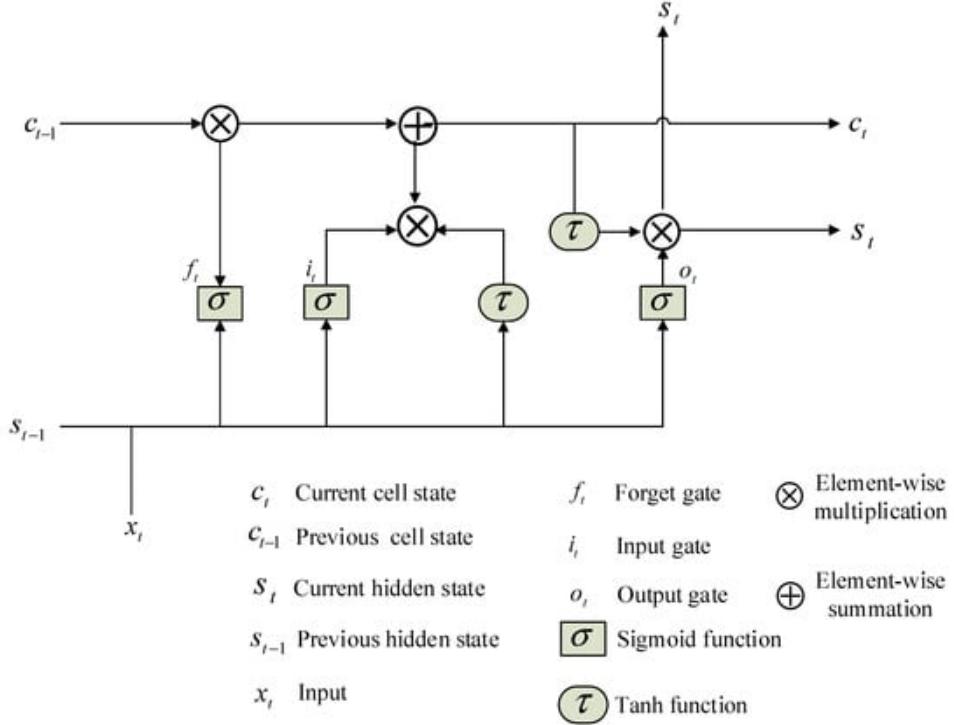


Figure 5. The LSTM cell architecture, crucial for air temperature forecasting, includes cell states, gates, and activation functions to capture temporal dependencies (Tran et al., 2021).

As the field of LSTM-based time series forecasting continues to evolve, ongoing research efforts aim to enhance model performance, explore novel architectures, and address practical challenges in real-world applications.

4 Methodology

4.1 Data collection and site

In this section, we present the data employed in our empirical study, which is dedicated to quantitative temporal weather data analysis. Table 1 outlines the availability of the parameters utilized, obtained from the Payerne measurement station. The dataset was sourced from the Federal Office of Meteorology and Climatology MeteoSwiss (<https://gate.meteoswiss.ch/idaweb>) and cover the period from March 2020 to December 2022. The dataset contains surface variations recorded at the Payerne station through measurement sensors. It is crucial to note that our proposed LSTM model is designed for pointwise weather forecasting at the specific station where the target variable is collected.



Figure 6. Geographical overview of Payerne, Switzerland, marked with a blue circle, showing key geographical features from a satellite image (Google Earth. Accessed: Jan. 24, 2024).

Payerne is located in the Vaud canton of Switzerland at approximately 46.8167° N latitude and 6.9333° E longitude. Situated at an elevation of around 480 meters (1,575 feet) above sea level,

the town benefits from its proximity to Lake Neuchâtel, which influences its local climate (“Could you give me information about Payerne?”, answer by ChatGPT, 24 January 2024).

Payerne experiences a temperate climate typical of Switzerland. Winters are generally cold, while summers tend to be warm. The local weather can be influenced by the surrounding mountains and the nearby lake, with potential effects on precipitation patterns and temperature variations. For the most accurate and current meteorological information, it is advisable to consult official meteorological sources or local weather authorities.

4.2 Data Analysis and Visualisation

In this section, a dataset sourced from Payerne measurement tower is analysed. Since the dataset is not publicly available, it was downloaded from MeteoSwiss database. Table x summarizes the variables and units. The dataset comprises 8 meteorological variables and a tracer gas variable with hourly time resolution.

Table 1. Available meteorological and trace gas variables and their units in the dataset

Variable	Unit
Temperature	°C
Dewpoint Temperature	°C
Relative Humidity	%
Pressure (AGL)	hPa
Precipitation	mm
Radiation	W/m ²
Wind Speed	m/s
Wind Direction	Degree (°)
CO ₂	ppm

The selection of input variables (features) for prediction tasks is crucial, with the choice dependent on data availability and their correlation. To ensure the reliability of our findings, rigorous data preprocessing steps were undertaken, addressing issues such as missing values and outliers. Descriptive statistics, including measures of correlation coefficient matrix, mean, and standard deviation, were computed to offer an initial understanding of the dataset's

characteristics. Consequently, an initial statistical analysis was conducted to assess the correlation between each available weather variable, as outlined below in Figure 7.

A heatmap is used to display correlations among the variables. Darker colors signify stronger relationships. For instance, a deep red hue indicates a notable positive correlation and blue refers negative correlation to simplifying the interpretation of complex relationships within our dataset. It can be observed that the temperature variable is highly and moderately positively correlated with the dewpoint temperature and radiation respectively, while the relative humidity and CO₂ variables are moderately negatively correlated with the temperature. On the other hand, the CO₂ variable demonstrates a moderate positive correlation with relative humidity, while the variables of wind speed and radiation exhibit moderate negative correlations with CO₂. The correlation results show that the hourly temperature is likely to be high as the increased dewpoint temperature and the decreased humidity. For CO₂, the analysis indicates that elevated humidity and reduced wind speed or radiation are associated with higher hourly CO₂ levels. Other weather variables have low correlation values. Thus, only using the individual weather variable and ignoring the time-domain relations of consecutive hourly data, we cannot build an effective relationship among the variables.

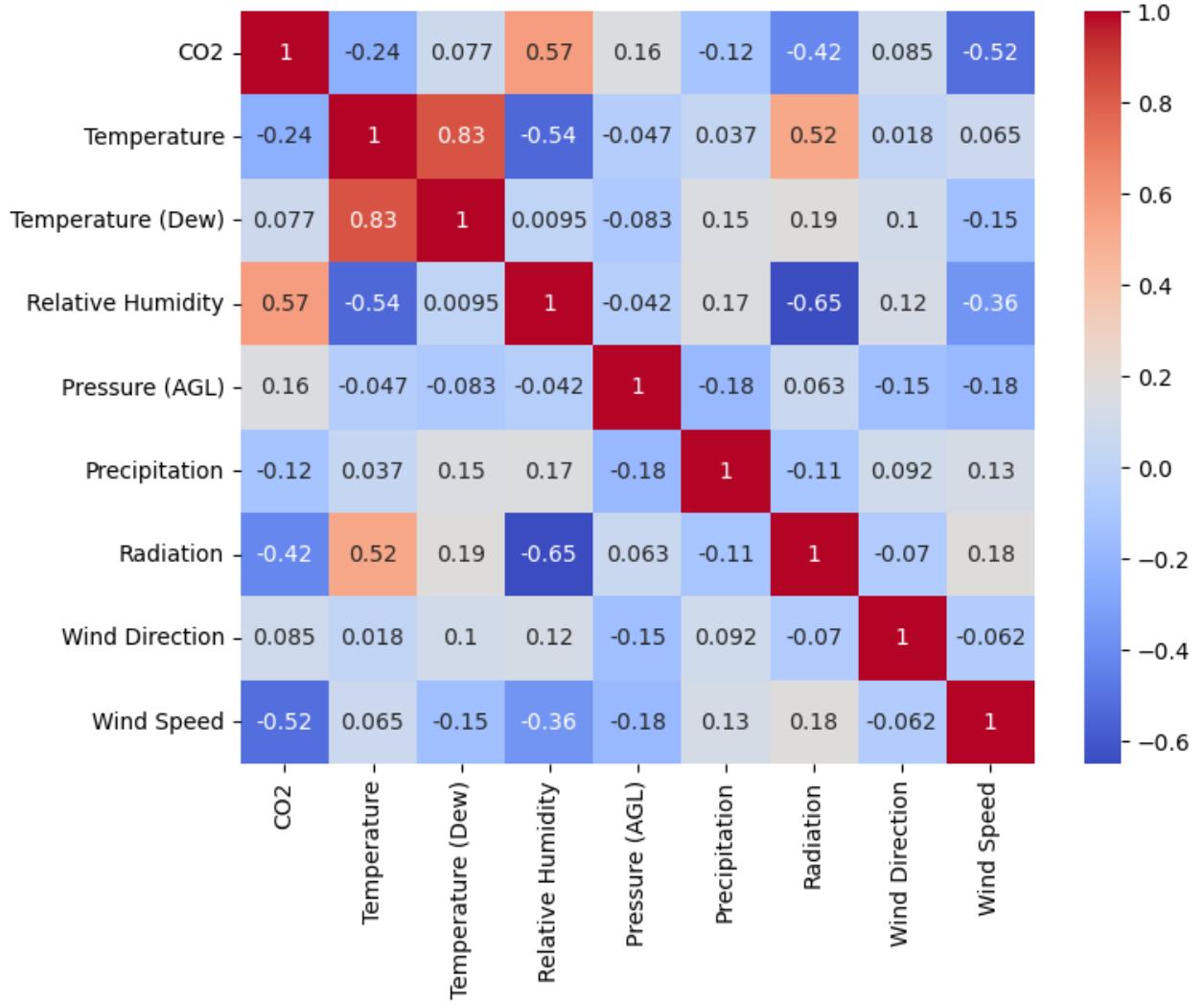


Figure 7. Heatmap representation of correlations between weather variables, with darker colors indicating stronger relationships.

Subsequently, Exploratory Data Analysis (EDA) dived into the complexities of the data, employing visualizations such as box plots, wind rose for wind speed, and various timeseries plots such as overall, monthly, and hourly (“What does EDA stand for?”, answer by ChatGPT, 28 February 2024.). Figure 8 depicts the boxplots of the variables.

In the boxplot visualization, each box represents the interquartile range (IQR) by the variables, with the median denoted by a central line. Outliers beyond the whiskers offer insights into potential extreme values. Variable labels clarify the representation of each box, and the widths of the boxes convey the spread of data. Wider boxes signify larger variability, while narrower boxes suggest more concentrated distributions. While the wind direction is having the highest IQR, the precipitation is quite narrow. Medians of temperature, dewpoint temperature and

pressure are relatively centered. That refers to symmetric distribution, mean and median alignment, relatively small skewness, and balanced spread of the data. Although data preprocessing steps were implemented to deal with outliers, it seems that the dataset couldn't get rid of the outliers. Dewpoint temperature seems as the optimal variable with respect to outlier characteristics.

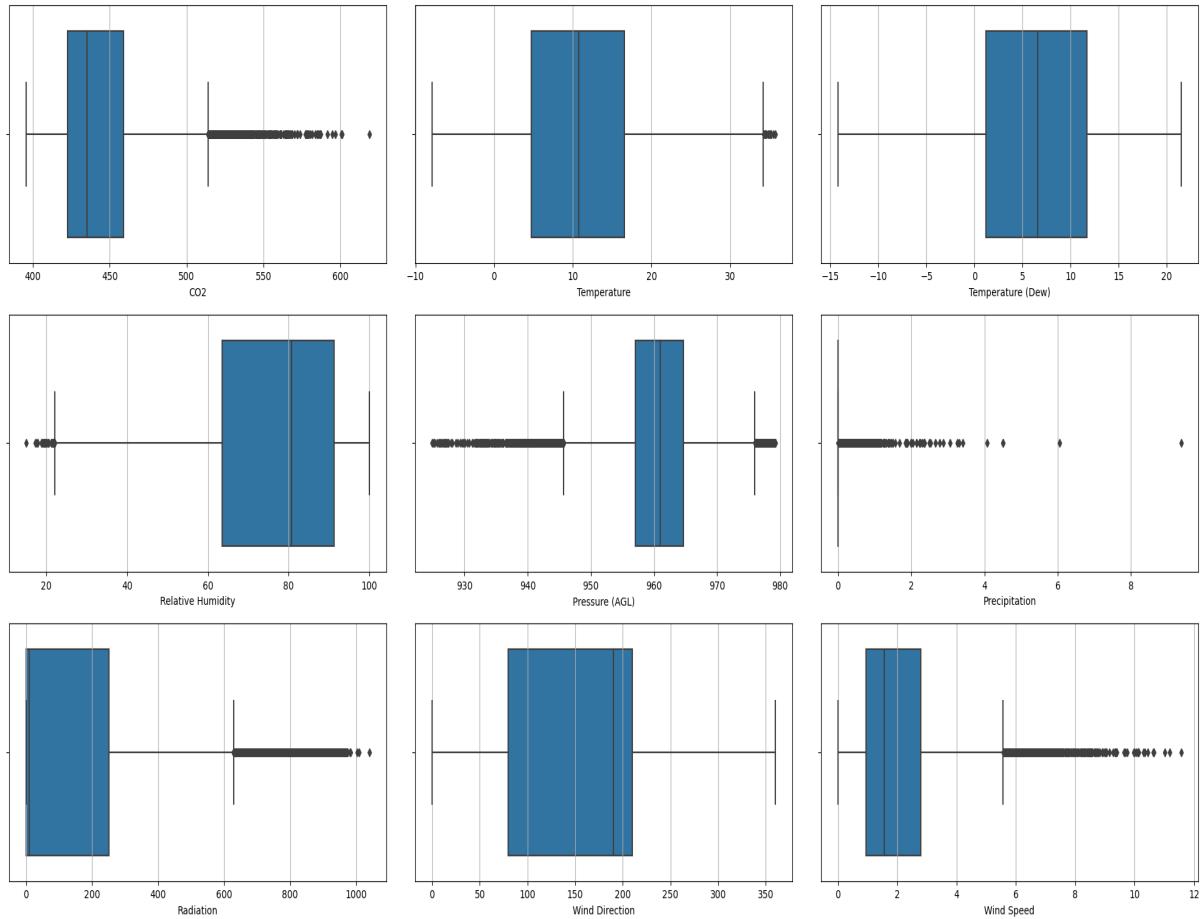


Figure 8. Boxplots of variables depicting variability and central tendencies, with taller boxes signaling greater variability and outliers marked.

Figure 9 presents the time series plots illustrating distinct diurnal cycles of the target variables and their correlated variables, revealing recurring patterns over a 24-hour period. These cyclic fluctuations capture the systematic variations in the observed variables, providing valuable insights into the temporal dynamics of the phenomena under the study. Analyzing these plots allows for a comprehensive understanding of the recurring patterns and trends characterizing the diurnal rhythms inherent in the data.

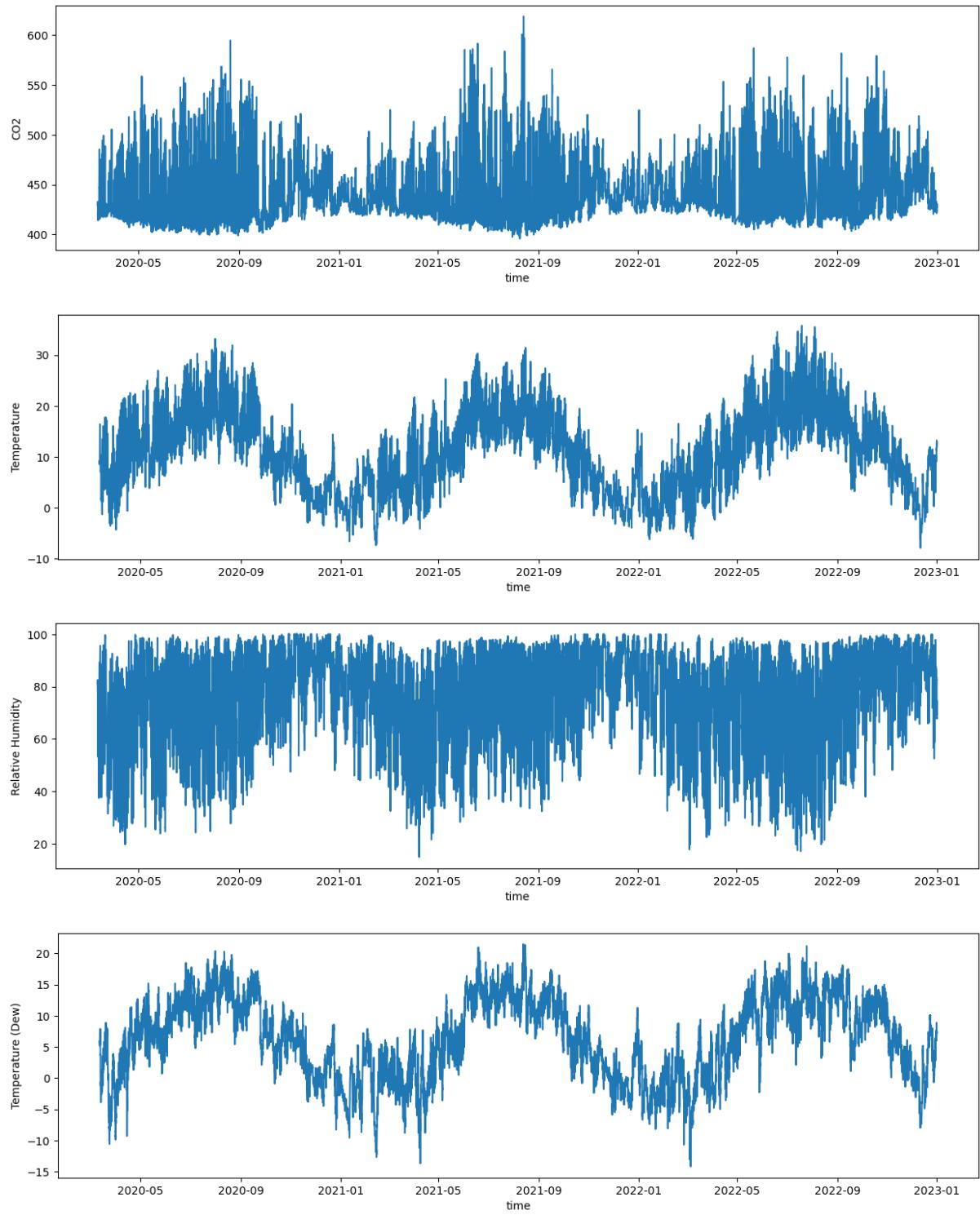


Figure 9. Time series plots of temperature, CO₂, and correlated variables, showing distinct diurnal cycles and recurring patterns over 24 hours.

In almost three-years time series plot of temperature, the data reveals patterns and fluctuations. Seasonal variations become apparent as temperature values exhibit periodic rises and falls, aligning with the cyclical nature of climatic conditions. In contrast, carbon dioxide (CO₂) and

relative humidity exhibit comparatively smoother cycles. We are going to see the impacts of the diurnal cycle range in the results section.

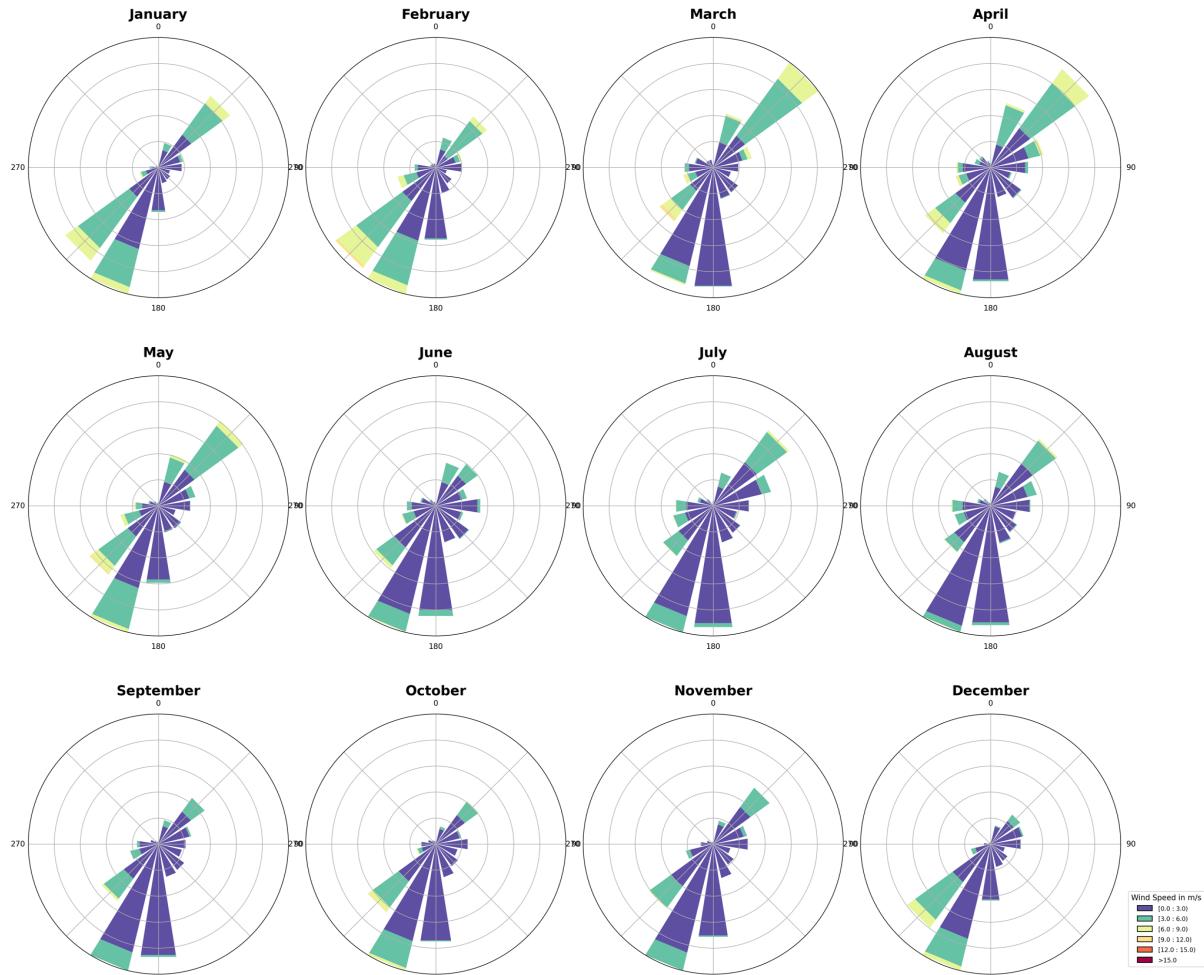


Figure 10. Wind rose diagrams for Payerne station, illustrating the distribution of wind speed and direction for each month.

The wind rose diagrams in Figure 10 for the Payerne station provide a detailed monthly overview of wind patterns, including both direction and speed, from January to December. Throughout the year, the predominant wind direction remains fairly consistent, with winds primarily coming from the south and southwest. This directional consistency suggests a stable regional wind pattern influenced by broader atmospheric conditions.

In terms of wind speed, there is a noticeable variation across the months. During the winter months of January and February wind speeds frequently fall within the 3.0 to 6.0 m/s range, with occasional peaks reaching up to 9.0 m/s. This trend continues into the spring, particularly

in April, which also sees some higher wind speeds up to 9.0 m/s. Notably, in March and April, there is a shift in wind direction towards the northeast. These higher speeds could be due to seasonal atmospheric changes and the transitional weather patterns typical of spring.

The summer months, particularly June and July, exhibit lower wind speeds, predominantly between 1.0 and 3.0 m/s. This reduction in wind speed during summer is likely due to more stable weather conditions and reduced atmospheric pressure differences. As the year progresses into autumn, wind speeds start to slightly increase again, with October showing speeds between 3.0 and 6.0 m/s, similar to the spring months.

By November and December, the wind speeds stabilize again within the 3.0 to 6.0 m/s range, mirroring the early winter months. Overall, the wind rose diagrams indicate that Payerne experiences moderate to strong winds mostly from the south and southwest throughout the year, with notable variations in wind speed during the transitional season of spring. These patterns are crucial for understanding the local climate and for applications such as renewable energy planning, where wind consistency and speed are critical factors.

Concurrently, data visualization tools played a pivotal role in conveying complex relationships. In our data analysis, a diverse set of visualizations was employed to effectively communicate findings. Heatmaps illuminated correlation patterns between variables, providing a comprehensive overview of relationships within the dataset. Boxplots were used to visualize the spread and central tendency of numerical variables, offering a detailed depiction of the dataset's variability. Timeseries plots facilitated the exploration of temporal trends, offering insights into dynamic patterns. Wind rose diagrams were particularly useful in illustrating the distribution of wind speed and direction throughout the year, revealing consistent patterns and seasonal variations. Collectively, these visualizations contributed to a more nuanced interpretation of our results, catering to the diverse nature of the dataset.

Interpretation of our findings situated them within the broader context of our research objectives, demonstrating the practical implications of our analysis. Nevertheless, it is crucial to acknowledge the limitations inherent in our dataset and recognize any assumptions made during the analysis process. In summary, the "Data Analysis and Visualization" phase not only provided us with an in-depth comprehension of the dataset, but also set the basis for informed conclusions critical to our overall research goals.

4.3 Model setup

The proposed model is based on LSTM networks and employs temporal weather data to identify patterns and forecast weather. The LSTM model, which has dealt with the RNN model's vanishing gradient problem, is one of the most powerful artificial neural network models for sequential data such as text, signal, and time series [38].

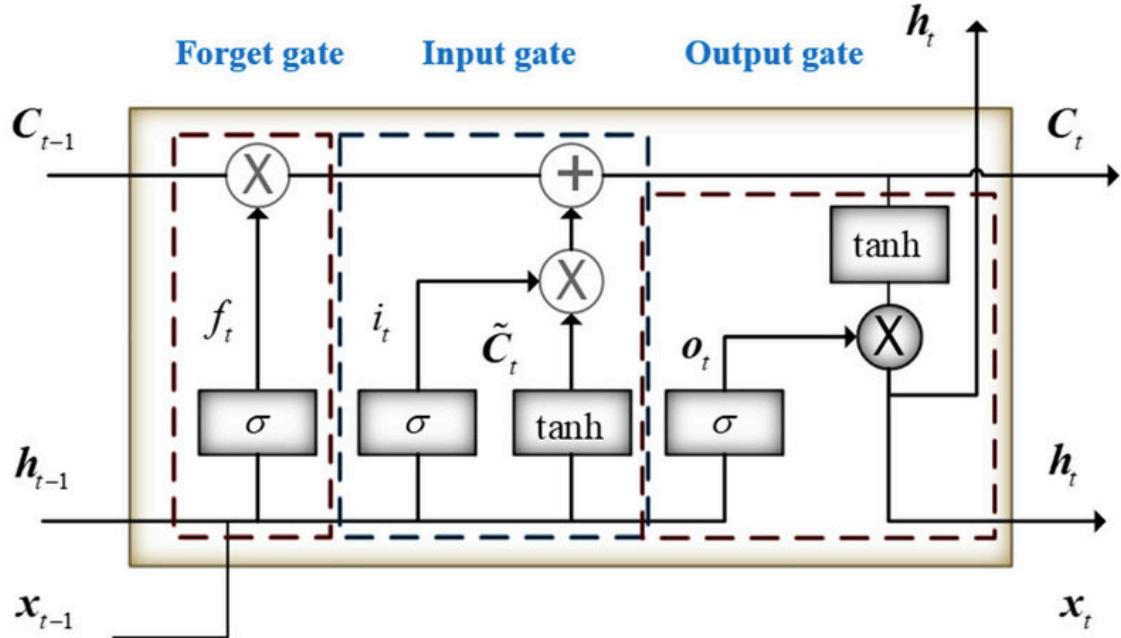


Figure 11. Internal structure of an LSTM unit, highlighting the roles of the forget, input, and output gates in regulating information flow (Jiang et al., 2019).

Memory blocks with three units, which are forget gate, input gate, and output gate, replace neurons in RNN. The forget gate can decide how much information from the previous memory vector should be forgotten. The input gate controls how much information enters the current cell. The output gate determines what information is output from the current cell. Figure 11 displays the technical design of an LSTM unit. C_{t-1} is the cell state at time $t-1$, h_{t-1} is the final output value of LSTM neural unit at time $t-1$, x_t is the input at time t , σ is the activation function of sigmoid, f_t is the output of forget gate at time t , i_t is the output of input gate at time t , and o_t is the output of the output gate at time t , C_t is the cell state at time t , and h_t is the output at time t . The gates can be described as following equations:

$$i_t = \sigma (\mathbf{W}_{i,x} \mathbf{x}_t + \mathbf{W}_{i,h} \mathbf{h}_{t-1} + b_i)$$

$$f_t = \sigma (\mathbf{W}_{f,x} \mathbf{x}_t + \mathbf{W}_{f,h} \mathbf{h}_{t-1} + b_f)$$

$$o_t = \sigma (\mathbf{W}_{o,x} \mathbf{x}_t + \mathbf{W}_{o,h} \mathbf{h}_{t-1} + b_o)$$

$$\tilde{c} = \tanh (\mathbf{W}_{\tilde{c},x} \mathbf{x}_t + \mathbf{W}_{\tilde{c},h} \mathbf{h}_{t-1} + b_{\tilde{c}})$$

Weight matrices $\mathbf{W}_{i,x}$, $\mathbf{W}_{i,h}$, $\mathbf{W}_{f,x}$, $\mathbf{W}_{f,h}$, $\mathbf{W}_{o,x}$, $\mathbf{W}_{o,h}$ along with bias vectors, b_i , b_f and b_o are involved in this calculation. Here, \mathbf{x}_t represents the current input, \mathbf{h}_{t-1} denotes the output of the LSTM at the previous time step $t-1$, and the function $\sigma()$ is the Sigmoid activation function. The forget gate determines the proportion of the previous memory value that should be discarded from the cell state. Likewise, the input gate determines the new input to be incorporated into the cell state. Subsequently, the cell state C_t is computed as follows:

$$C_t = f_t C_{t-1} + i_t \tilde{c}_t$$

The output \mathbf{h}_t of the LSTM at the time t is derived as:

$$\mathbf{h}_t = o_t \tanh(C_t)$$

Finally, we project the output \mathbf{h}_t to the predicted output $\tilde{\mathbf{y}}_t$ as:

$$\tilde{\mathbf{y}}_t = \mathbf{W}_y \mathbf{h}_t$$

where \mathbf{W}_y is a projection matrix to reduce dimension of \mathbf{h}_t . Within the LSTM architecture, a feature vector \mathbf{x}_t is inputted into the network at time t . At the current state, the LSTM cell receives feedback \mathbf{h}_{t-1} from the preceding LSTM cell to capture temporal dependencies. The objective of network training is to minimize the standard squared error objective function f , which is based on the targets \mathbf{y}_t as

$$f = \sum_t || \mathbf{y}_t - \tilde{\mathbf{y}}_t ||^2$$

Through the application of backpropagation coupled with gradient descent, adjustments are made to the weights and biases during training. Upon completion of the optimization process

for one batch of the training dataset, using the backpropagation algorithm, an epoch concludes. Given that training LSTM networks is conducted offline, the computational time required for training is not a crucial factor for the application. Nonetheless, prediction utilizing the trained LSTM networks is executed swiftly.

To train a neural network model, LSTM requires back propagation which is a test for errors that works backward from output nodes to input nodes. It is a useful mathematical tool for improving prediction accuracy in data mining and machine learning. The error between the output value and the actual value generated by the model without training propagates layer by layer along the opposite direction of forward calculation in back propagation. The input of sequence data can be mapped to specific outputs in the forward calculation of LSTM [40].

Our objective is to forecast the hourly temperature and CO₂ values y_t based on the provided feature vector x_{t-5} . The data spanned from March 2020 to December 2022, and it was split into three parts: 50% for the training dataset, and 25% each for testing and validation. In this experiment, we compared the prediction performance of LSTM and RNN models.

This algorithm minimized the sum of squared differences between actual hourly values and those predicted using feature variables from the training dataset. The model's prediction accuracy was then verified using the testing dataset. Notably, this algorithm did not require any parameter tuning.

The LSTM and RNN networks were implemented using the Keras deep learning package, with the model architecture detailed in Table 2. The table describes the architectures and training configurations used for LSTM and RNN models in a study. Specifically, the table outlines four different models, each with distinct configurations designed to predict temperature and CO₂ levels.

Table 2. Architectures of LSTM and RNN models. The boxes represent the architecture of LSTM for temperature and CO₂ and the architecture of RNN for temperature and CO₂ respectively

```

model = Sequential()
model.add(LSTM(units=100, activation="Sigmoid", return_sequences = True))
model.add(LSTM(units=100))
model.add(Dense(units=1))
opt = tf.keras.optimizers.legacy.Adam(lr=0.0001)
model.compile(optimizer = opt, loss = "mse")


---


model.fit(X_train, y_train, epochs=30, batch_size=24)

model = Sequential()
model.add(LSTM(units=100, activation="Tanh"))
model.add(Dense(units=1))
opt = tf.keras.optimizers.legacy.Adam(lr=0.001)
model.compile(optimizer = opt, loss = "mse")


---


model.fit(X_train, y_train, epochs=30, batch_size=24)

model = Sequential()
model.add(SimpleRNN(units=10, activation="ReLU"))
model.add(Dense(units=1))
opt = tf.keras.optimizers.legacy.Adam(lr=0.001)
model.compile(optimizer = opt, loss = "mse")
model.fit(X_train, y_train, epochs=30, batch_size=24)


---


model = Sequential()
model.add(SimpleRNN(units=10, activation="Sigmoid"))
model.add(Dense(units=1))
opt = tf.keras.optimizers.legacy.Adam(lr=0.001)
model.compile(optimizer = opt, loss = "mse")
model.fit(X_train, y_train, epochs=30, batch_size=24)

```

1. **LSTM model for temperature:** This model uses two LSTM layers, the first with 100 units and a Sigmoid activation function, and the second LSTM layer also with 100 units. The output layer is a Dense layer with a single unit. The model is trained using the Adam optimizer with a learning rate of 0.0001, and the loss function is Mean Squared Error (MSE). Training is performed over 30 epochs with a batch size of 24.

2. **LSTM model for CO₂:** This model is simpler, consisting of a single LSTM layer with 100 units and a Tanh activation function. It also uses a Dense output layer with one unit. The Adam optimizer is employed with a higher learning rate of 0.001, and the model also uses MSE as the loss function. Like the first model, it is trained for 30 epochs with a batch size of 24.

3. **RNN model for temperature:** This model utilizes a SimpleRNN layer with 10 units and a ReLU activation function, followed by a Dense output layer with a single unit. The training setup is similar to the LSTM models, with the Adam optimizer (learning rate 0.001) and MSE loss function. Training is carried out over 30 epochs with a batch size of 24.

4. **RNN model for CO₂:** This model also uses a SimpleRNN layer but with a Sigmoid activation function. The architecture and training configuration are otherwise identical to RNN Model 1, with the Adam optimizer (learning rate 0.001) and MSE loss function used for training over 30 epochs with a batch size of 24.

These configurations were designed to evaluate the performance of LSTM and RNN architectures in predicting hourly solar irradiance using different activation functions and hyperparameters. The models are tuned to optimize the accuracy of their predictions by minimizing the mean squared error between predicted and actual values.

4.4 Performance evaluation

The Minimum Absolute Error (MAE) is a fundamental metric used to evaluate the performance of Long Short-Term Memory (LSTM) models in weather time series forecasting [4], [6]. It measures the average absolute difference between the predicted and actual values, providing insights into the accuracy of the model's predictions. Mathematically, MAE is calculated using the following formula:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Where:

- MAE is the Minimum Absolute Error,
- n is the number of observations,
- y_i is the actual value of the observation,
- \hat{y}_i is the predicted value of the observation.

The MAE metric provides a clear and interpretable measure of the model's performance, allowing researchers to assess its ability to accurately capture the underlying patterns in meteorological data. By minimizing the absolute discrepancies between predicted and observed values, MAE enables the evaluation and comparison of different LSTM configurations, facilitating the optimization of model architectures and hyperparameters to enhance forecasting accuracy.

5 Results

5.1 Model sensitivity

Model sensitivity refers to the responsiveness of a model's performance to changes in its parameters and input data. This analysis is crucial in machine learning, particularly in neural networks like Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN), which are used for time series forecasting. Sensitivity analysis helps in identifying the most influential parameters that significantly affect model accuracy, providing insights for optimizing model configurations (“What is model sensitivity in deep learning?”, answer by ChatGPT, 5 April 2024).

Sensitivity analysis involves systematically varying hyperparameters, such as the number of neurons, layers, activation functions, and input features, to assess their impact on model performance metrics, such as Mean Absolute Error (MAE). This approach is essential for determining the robustness and efficiency of the model under different configurations. For instance, the choice of the number of neurons or layers can drastically influence the model's ability to learn from data and generalize to new instances [4], [16].

The importance of conducting sensitivity analysis is well-documented. Bergstra and Bengio [41] emphasize the necessity of hyperparameter optimization in deep learning, demonstrating that even minor adjustments can lead to significant performance improvements. Similarly, Snoek et al. [42] highlight Bayesian optimization as an effective method for hyperparameter tuning, which helps in efficiently navigating the vast hyperparameter space to find optimal settings. Such techniques are critical in enhancing model accuracy and reliability, particularly in complex applications like weather forecasting, where data patterns can be highly nonlinear and variable [20].

In this thesis, a comprehensive sensitivity analysis was conducted on the following parameters to evaluate their impact on the performance of LSTM and RNN models for temperature and CO₂ forecasting:

- **Number of Neurons:** Different sizes of hidden layers were tested to determine the optimal number of neurons for each model configuration. This parameter directly affects the model's capacity to learn from the data and capture underlying patterns.
- **Number of Layers:** Various depths of the neural network were assessed. Increasing the number of layers can enhance the model's ability to represent complex functions, but it also increases the risk of overfitting and computational complexity.
- **Scaler:** Different methods for normalizing data, such as Min-Max scaling and Standard scaling, were evaluated to understand their effects on the training process and model performance.
- **Activation Function:** The impact of different activation functions, including ReLU, Sigmoid, and Tanh, was compared. Activation functions play a crucial role in introducing non-linearity into the model, which is essential for learning complex data patterns.

These sensitivity tests provide a detailed understanding of how each parameter influences the model's predictive capabilities, guiding the selection of optimal configurations for improved accuracy in forecasting tasks.

5.2 Sensitivity Parameters

5.2.1 Neuron and Features

In the context of evaluating the performance of LSTM and RNN models for temperature and CO₂ predictions, sensitivity analysis was conducted to understand the impact of varying the number of features and neurons on the Mean Absolute Error (MAE). This analysis provides insights into how different configurations affect model accuracy, aiding in the optimization of neural network architectures [16], [5].

Sensitivity analysis is crucial in determining the robustness and efficacy of machine learning models, particularly for time-series forecasting tasks. By systematically varying key

parameters such as the number of features and neurons, one can assess their influence on the model's predictive performance. The number of features represents the dimensionality of the input data, which can significantly impact the model's ability to capture relevant patterns and dependencies in the dataset. For instance, using a higher number of features can potentially enhance the model's predictive accuracy by providing more comprehensive information about the underlying process being modeled [41].

Similarly, the number of neurons in the hidden layers of the network is a critical parameter that influences the learning capacity of the model. Neurons act as computational units that process the input data and learn complex patterns through training. Increasing the number of neurons typically enhances the model's ability to capture intricate relationships within the data. However, it can also lead to overfitting, where the model performs well on the training data but poorly on unseen data. Therefore, finding an optimal balance in the number of neurons is essential for building a robust and generalizable model [5].

In the case of temperature prediction, the sensitivity tests revealed that both LSTM and RNN models show different levels of performance improvement with the increase in the number of features and neurons. According to Table 2, with a single feature, both models exhibited relatively high MAE values, indicating limited predictive power. However, as the number of features increased to two, there was a noticeable reduction in MAE for the LSTM model, suggesting that additional features provided valuable information that improved the model's accuracy. The LSTM model, in particular, showed substantial improvement with four features, achieving its lowest MAE, highlighting its superior ability to learn from higher-dimensional data.

For CO₂ prediction, the sensitivity analysis also demonstrated significant variations in MAE based on the number of features and neurons. With a single feature, the MAE values were higher for both models, similar to the temperature prediction results. However, as the number of features increased, the LSTM model consistently outperformed the RNN model, achieving lower MAE values across different neuron configurations. This trend underscores the importance of utilizing multiple features in improving the predictive accuracy of neural network models [5], [4].

In summary, the sensitivity analysis conducted for both temperature and CO₂ predictions underscores the critical role of selecting an appropriate number of features and neurons to optimize model performance. By understanding the impact of these parameters on MAE, researchers can make informed decisions about model architecture, leading to more accurate and reliable predictions in time-series forecasting tasks [16], [5], [41].

Table 3. Impact of varying the number of features and neurons on the MAE for temperature prediction

Features	Number of Feature	Number of Neuron	MAE for LSTM	MAE for RNN
Temperature (°C)	1	10	0.8337	0.8256
		20	0.8389	0.8259
		30		0.8290
		40		0.8297
		50	0.8436	0.8298
		100	0.8396	
		200	0.8420	
		300	0.8428	
		400	0.8437	
		500	0.8474	
Temperature (°C) Dew Point Temperature (°C)	2	10	0.6788	0.7315
		20	0.7343	0.7157
		30		0.7745
		40		0.7739
		50	0.7120	0.7219
		100	0.6500	
		200	0.6253	
		300	0.6067	
		400	0.5417	
		500	0.5641	
Temperature (°C) Dewpoint Temperature (°C) Relative Humidity (%)	3	10	0.3055	0.1857
		20	0.2643	0.1995
		30		0.2409
		40		0.3973
		50	0.2117	0.2916
		100	0.1837	
		200	0.1945	
		300	0.2211	
		400	0.2111	
		500	0.2171	
Temperature (°C) Dewpoint Temperature (°C) Relative Humidity (%) Radiation (W/m2)	4	10	0.1815	0.4719
		20	0.2040	0.2501
		30		0.2508
		40		0.2816
		50	0.1287	0.2411
		100	0.1071	
		200	0.1299	
		300	0.1551	
		400	0.1581	
		500	0.1553	

Temperature (°C) Dewpoint Temperature (°C) Relative Humidity (%) Radiation (W/m2) Wind Speed (m/s)	5	10	0.1977	0.2537
		20	0.2247	0.2038
		30		0.2301
		40		0.2630
		50	0.1560	0.2919
		100	0.1237	
		200	0.1385	
		300	0.1572	
		400	0.1705	
		500	0.1627	
Temperature (°C) Dewpoint Temperature (°C) Relative Humidity (%) Radiation (W/m2) Wind Speed (m/s) Pressure (AGL) (hPa)	6	10	0.1709	0.3482
		20	0.1420	0.3325
		30		0.2041
		40		0.2100
		50	0.1225	0.2425
		100	0.1169	
		200	0.1262	
		300	0.1458	
		400	0.1415	
		500	0.1801	
Temperature (°C) Dewpoint Temperature (°C) Relative Humidity (%) Radiation (W/m2) Wind Speed (m/s) Pressure (AGL) (hPa) Precipitation (mm)	7	10	0.1390	0.2200
		20	0.1786	0.2015
		30		0.2421
		40		0.1863
		50	0.1501	0.2041
		100	0.1151	
		200	0.1347	
		300	0.1553	
		400	0.1610	
		500	0.1677	
Temperature (°C) Dewpoint Temperature (°C) Relative Humidity (%) Radiation (W/m2) Wind Speed (m/s) Pressure (AGL) (hPa) Precipitation (mm) Wind Direction (Degree (°))	8	10	0.1474	0.3533
		20	0.1638	0.3134
		30		0.2949
		40		0.2188
		50	0.1424	0.2803
		100	0.1347	
		200	0.1351	
		300	0.1525	
		400	0.1531	
		500	0.1685	

For temperature prediction, the sensitivity tests revealed significant variations in MAE based on the number of features and neurons used. With a single feature, both LSTM and RNN models exhibited high MAE values across all neuron configurations, with LSTM generally performing slightly better. For instance, the MAE for LSTM with one feature ranged from

0.8337 to 0.8474 as the number of neurons increased from 10 to 500. In contrast, the RNN model showed a consistent MAE around 0.8256 to 0.8298 as the number of neurons increased from 10 to 50. When the number of features was increased to two, the performance improved notably. The LSTM model demonstrated lower MAE values, dropping from 0.6788 with 10 neurons to 0.5641 with 500 neurons. Similarly, the RNN model's MAE ranged from 0.7315 to 0.7219 for 10 to 50 neurons but did not improve significantly with more neurons. This indicates that while increasing neurons can enhance model performance, the number of features plays a crucial role in reducing prediction error.

The most substantial improvement was observed with three features. The LSTM model achieved its lowest MAE of 0.1837 with 100 neurons, indicating a highly accurate prediction capability. In comparison, the RNN model also performed better with three features, achieving its best MAE of 0.1857 with 10 neurons. This trend continued with four features, where the LSTM model's MAE dropped further to 0.1071 with 100 neurons.

With five to eight features, the LSTM model consistently outperformed the RNN model, achieving lower MAE values across different neuron configurations. For example, with five features, the LSTM model's MAE ranged from 0.1977 to 0.1627 as the number of neurons increased from 10 to 500, while the RNN model's MAE varied more widely as the number of neurons increased from 10 to 50. The best performance for LSTM was recorded with four features, showing an MAE of 0.1071 with 100 neurons. In contrast, the RNN model achieved its lowest MAE of 0.1857 with three features and 10 neurons.

Table 4. Impact of varying the number of features and neurons on the MAE for CO₂ prediction

Features	Number of Feature	Number of Neuron	MAE of LSTM	MAE of RNN
CO ₂ (ppm)	1	10	0.7957	0.7937
		20	0.7976	0.7949
		50	0.7976	0.7938
		100	0.7974	0.7958
		200	0.7944	0.7966
		300	0.7956	0.7960
		400	0.7960	
		500	0.7959	
		10	0.7731	0.8210
		20	0.8149	0.6555
CO ₂ (ppm) Relative Humidity (%)	2	50	0.6613	0.7612
		100	0.7781	1.0126
		200	0.7348	0.8377
		300	0.7305	1.1333
		400	0.7532	
		500	0.8451	
		10	0.6597	0.7383
		20	0.5761	0.7080
		50	0.5314	0.6880
		100	0.5013	0.6939
CO ₂ (ppm) Relative Humidity (%) Wind Speed (m/s)	3	200	0.5319	0.7484
		300	0.5166	0.6566
		400	0.5370	
		500	0.6600	
		10	0.5076	0.7368
		20	0.5555	0.6847
		50	0.5114	0.6166
		100	0.5041	0.5540
		200	0.5156	0.8262
		300	0.5077	0.6304
CO ₂ (ppm) Relative Humidity (%) Wind Speed (m/s) Radiation (W/m ²)	4	400	0.5266	
		500	0.6259	
		10	0.5850	0.6592
		20	0.4941	0.8030
		50	0.4601	0.6024
		100	0.4415	0.5861
		200	0.4955	0.6520
		300	0.4943	0.6818
		400	0.4756	
		500	0.5205	
CO ₂ (ppm) Relative Humidity (%) Wind Speed (m/s) Radiation (W/m ²) Temperature (°C)	5	10	0.5850	0.6592
		20	0.4941	0.8030
		50	0.4601	0.6024
		100	0.4415	0.5861
		200	0.4955	0.6520
		300	0.4943	0.6818
		400	0.4756	
		500	0.5205	

CO ₂ (ppm) Relative Humidity (%) Wind Speed (m/s) Radiation (W/m ²) Temperature (°C) Pressure (AGL) (hPa)	6	10	0.4755	0.7607
		20	0.4631	0.7274
		50	0.4454	0.6706
		100	0.4528	0.5883
		200	0.4396	0.6149
		300	0.4902	0.7301
		400	0.5828	
		500	0.6196	
CO ₂ (ppm) Relative Humidity (%) Wind Speed (m/s) Radiation (W/m ²) Temperature (°C) Pressure (AGL) (hPa) Precipitation (mm)	7	10	0.4290	0.5973
		20	0.4453	0.6964
		50	0.4364	0.5941
		100	0.4232	0.5207
		200	0.4184	0.6607
		300	0.4664	0.5501
		400	0.4591	
		500	0.4668	
CO ₂ (ppm) Relative Humidity (%) Wind Speed (m/s) Radiation (W/m ²) Temperature (°C) Pressure (AGL) (hPa) Precipitation (mm) Wind Direction (Degree (°))	8	10	0.4792	0.6173
		20	0.4369	0.6319
		50	0.4202	0.5807
		100	0.4568	0.4978
		200	0.4780	0.5357
		300	0.4573	0.7040
		400	0.5302	
		500	0.4699	
CO ₂ (ppm) Relative Humidity (%) Wind Speed (m/s) Radiation (W/m ²) Temperature (°C) Pressure (AGL) (hPa) Precipitation (mm) Wind Direction (Degree (°)) Dewpoint Temperature (°C)	9	10	0.4499	0.7285
		20	0.4422	0.6769
		50	0.4055	0.5970
		100	0.4030	0.5278
		200	0.4264	0.5459
		300	0.4593	0.6316
		400	0.4671	
		500	0.5094	

The sensitivity analysis for CO₂ predictions mirrored some trends observed in temperature predictions, though with notable differences. For one feature, both LSTM and RNN models showed higher MAE values. According to Table 4, the LSTM model's MAE ranged from 0.7957 to 0.7959 for 10 to 500 neurons, whereas the RNN model's MAE stayed close to 0.7937 to 0.7960 for 10 to 300 neurons.

With two features, a significant divergence in model performance was observed. The LSTM model's MAE improved to 0.6613 with 50 neurons, while the RNN model's performance was

less stable, showing a wider range of MAE values from 0.6555 to 1.1333. This indicates that RNN models might be more sensitive to the number of neurons when fewer features are used.

As the number of features increased to three, the LSTM model continued to show better performance, with MAE values dropping to 0.5013 with 100 neurons. The RNN model showed improvement but remained less consistent, with MAE values around 0.6566 to 0.7383. With four features, the LSTM model achieved an MAE of 0.5041 with 100 neurons, while the RNN model's MAE ranged from 0.5540 to 0.8262.

The best performance for CO₂ prediction was observed with five to eight features. The LSTM model achieved its lowest MAE of 0.4030 with nine features and 100 neurons, indicating a high accuracy level. The RNN model, although improved, showed higher MAE values in comparison. For instance, with eight features, the RNN model's lowest MAE with 100 neurons is 0.4978.

Overall, the sensitivity analysis underscores the importance of optimizing both the number of features and neurons to achieve the best predictive performance in LSTM and RNN models. The LSTM model consistently outperformed the RNN model across various configurations, highlighting its robustness and accuracy in handling time-series prediction tasks for temperature and CO₂ levels.

5.2.2 Layer

The sensitivity analysis of layer depth in Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) models is critical in understanding how model complexity impacts the prediction accuracy for temperature and CO₂ forecasting. Accurate forecasting of these parameters is vital for various applications, including climate change analysis, agricultural planning, and environmental monitoring. The layer depth of a neural network influences its ability to learn and represent complex temporal patterns within the data, which is essential for generating reliable predictions. Deep neural networks, such as LSTMs and RNNs, are particularly suited for time series forecasting because of their ability to capture sequential dependencies and dynamic changes in the input data [4]. However, increasing the number of layers can introduce challenges, including the vanishing gradient problem, increased

computational load, and a higher risk of overfitting, all of which can degrade model performance [43].

To explore the impact of layer depth on forecasting accuracy, we conducted a series of sensitivity tests on both LSTM and RNN models for temperature and CO₂ predictions. The results provide insights into the optimal configuration of these models, balancing complexity and performance.

Table 5. Layer sensitivity results for temperature

Models	Number of Layers	MAE	Number of Features	Number of Neurons
LSTM	1	0.1071	4	100
	2	0.1041		
	3	0.1434		
	4	0.1605		
RNN	1	0.1857	3	10
	2	0.4160		
	3	0.2414		
	4	0.1982		

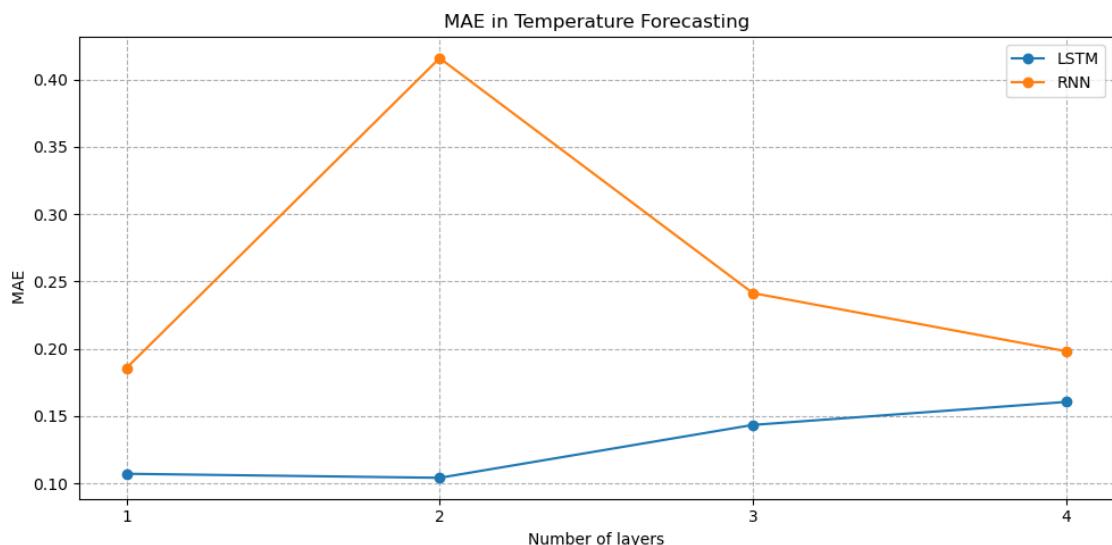


Figure 12. Graphical representation of MAE for the sensitivity test results of the layers for temperature.

For temperature forecasting, the Mean Absolute Error (MAE) values indicate the following:

- **LSTM Model:** The LSTM model performs optimally with two layers, achieving the lowest MAE of 0.1041. This suggests that a moderate depth allows the model to capture the necessary temporal dependencies without overfitting. Adding more layers (three and four) increases the MAE, indicating potential overfitting or increased difficulty in training deeper networks.
- **RNN Model:** The RNN model exhibits a different sensitivity pattern. A single-layer RNN achieves the lowest MAE of 0.1857, suggesting that simpler architectures are more effective for this task. Increasing the number of layers generally degrades performance, with a notable increase in MAE for the two-layer RNN (0.4160), which may result from the vanishing gradient problem or overfitting issues in deeper RNNs.

Table 6. Layer sensitivity results for CO₂

Models	Number of Layers	MAE	Number of Features	Number of Neurons
LSTM	1	0.4030	9	100
	2	0.4664		
	3	0.5512		
	4	0.4661		
RNN	1	0.4978	8	100
	2	0.7861		
	3	0.7661		
	4	0.6799		

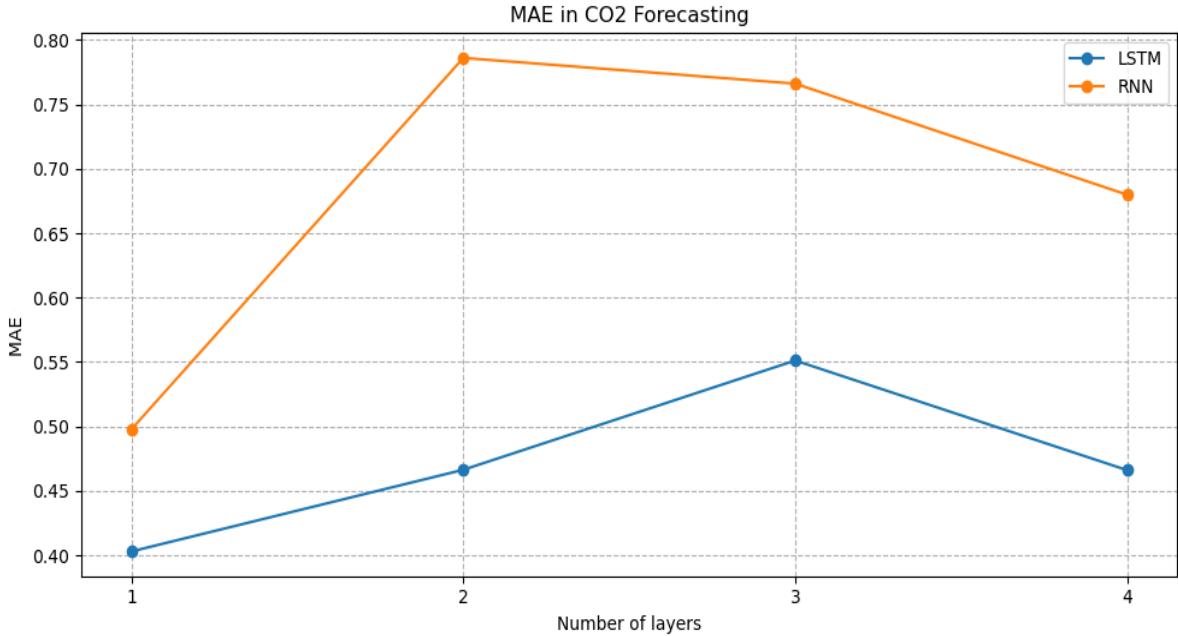


Figure 13. Evaluation of the impact of varying the number of layers on the MAE for CO₂ forecasting, showing better performance with simpler architectures.

For CO₂ forecasting, the sensitivity results highlight the following:

- **LSTM Models:** The LSTM model with a single layer achieves the lowest MAE of 0.4030. Increasing the number of layers generally worsens performance, with a peak MAE at three layers (0.5512). This again suggests that simpler LSTM architectures are more effective for CO₂ forecasting, possibly due to the increased complexity and potential overfitting in deeper networks.
- **RNN Models:** The RNN model also shows a preference for simpler architectures with the lowest MAE (0.4978) occurring at a single layer. The performance significantly deteriorates as the number of layers increases, with the two-layer RNN showing the highest MAE (0.7861). This trend underscores the challenges of training deeper RNNs, particularly in managing the vanishing gradient problem and overfitting.

The results from the sensitivity analysis reveal critical insights into the optimal architecture for LSTM and RNN models in the context of temperature and CO₂ forecasting:

1. **Optimal Layer Depth:** Both LSTM and RNN models demonstrate improved performance with fewer layers. The optimal number of layers is generally one or two, beyond which performance degrades due to increased complexity and overfitting.
2. **Model Complexity vs. Performance:** While deeper models theoretically capture more complex patterns, they are also more prone to overfitting and computational challenges. Simpler architectures (one or two layers) strike a balance between capturing necessary temporal dependencies and maintaining generalization capabilities.
3. **Comparative Performance:** LSTM models consistently outperform RNN models across different layer depths for both temperature and CO₂ forecasting. This superiority is attributed to LSTM's capability to mitigate the vanishing gradient problem, thus retaining relevant information over longer sequences.

These findings underscore the importance of careful model configuration, particularly in selecting an appropriate layer depth to balance complexity and performance. For practical applications in meteorological forecasting, adopting a simpler architecture with one or two layers appears most effective for both LSTM and RNN models.

5.2.3 Scaler

In the context of deep learning models, data scaling plays a pivotal role in ensuring the effectiveness of the training process. Scaling methods such as StandardScaler, MinMaxScaler, and RobustScaler standardize the features to ensure that each one contributes equally to the results, thus preventing certain features from dominating the learning process due to their larger numerical ranges. This process of scaling is crucial as it ensures that the numerical stability of the models is maintained, thereby enhancing the convergence speed and accuracy of the learning algorithms. Specifically, StandardScaler normalizes the features by removing the mean and scaling to unit variance, MinMaxScaler scales the data to a fixed range, usually 0 to 1, and RobustScaler uses statistics that are robust to outliers, such as the median and the interquartile range, for scaling (“What is scaling and scalers in deep learning?”, answer by ChatGPT, 10 May 2024).

The importance of scaling is underscored in several studies. For instance, J. Bergstra and Y. Bengio highlighted in their research that proper data scaling can significantly improve the performance of machine learning models by ensuring that the gradient descent converges more rapidly during the training phase [41]. Similarly, A. Krizhevsky et al. demonstrated that normalized inputs help in achieving better performance in deep neural networks, as they reduce the risk of vanishing or exploding gradients, which are common issues in deep learning models [44]. Furthermore, H. Wang and L. Li emphasized the effectiveness of scaling methods in enhancing the performance of time series forecasting models, particularly in weather prediction scenarios where the range of input features can vary widely [40].

The sensitivity analysis for temperature forecasting using LSTM and RNN models reveals the following results:

Table 7. Temperature forecasting MAE with different scalers

Models	Scaler	MAE	Number of Features	Number of Neurons	Number of Layers
LSTM	Standard	0.1041	4	100	2
	MinMax	0.0527			
	Robust	0.0861			
RNN	Standard	0.1857	3	10	1
	MinMax	0.1629			
	Robust	0.1668			

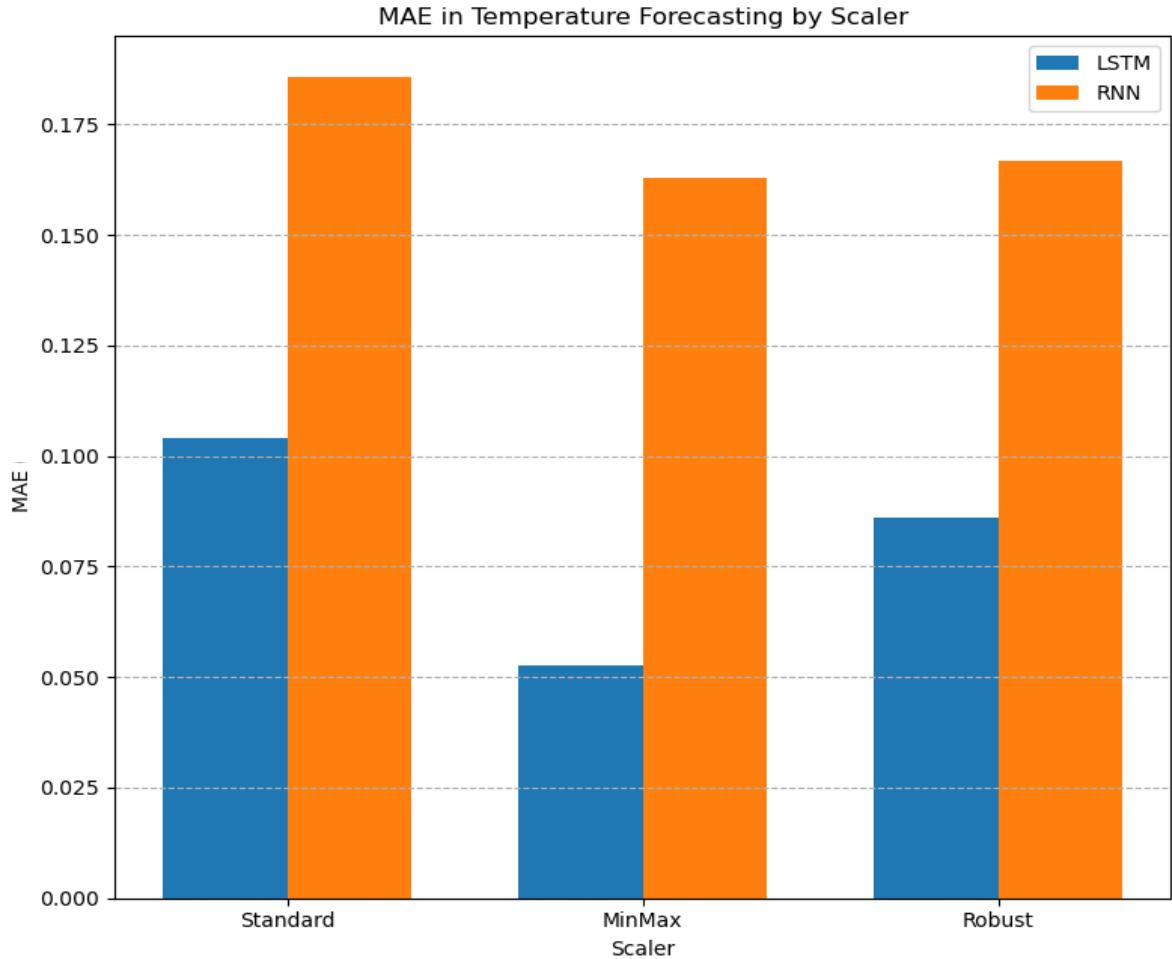


Figure 14. Comparison of the MAE for temperature predictions using different data scaling techniques, highlighting MinMaxScaler's effectiveness.

The analysis shows that the MinMaxScaler yields the lowest Mean Absolute Error (MAE) for both LSTM and RNN models when forecasting temperature. Specifically, the LSTM model achieves an MAE of 0.0527, which is significantly lower compared to the StandardScaler (0.1041) and RobustScaler (0.0861). Similarly, the RNN model performs best with the MinMaxScaler, obtaining an MAE of 0.1629, which is better than the StandardScaler (0.1857) and RobustScaler (0.1668).

These results suggest that MinMaxScaler, which scales the data to a range between 0 and 1, helps both LSTM and RNN models in capturing the underlying patterns of temperature data more effectively. This is likely because MinMaxScaler normalizes the input features to a smaller range, making the optimization process smoother and more efficient.

The sensitivity analysis for CO₂ forecasting using LSTM and RNN models presents the following results:

Table 8. MAE for CO₂ forecasting with different scalers

Models	Scaler	MAE	Number of Features	Number of Neurons	Number of Layers
LSTM	Standard	0.4030	4	100	2
	MinMax	0.0578			
	Robust	0.3471			
RNN	Standard	0.4978	3	10	1
	MinMax	0.0914			
	Robust	0.4568			

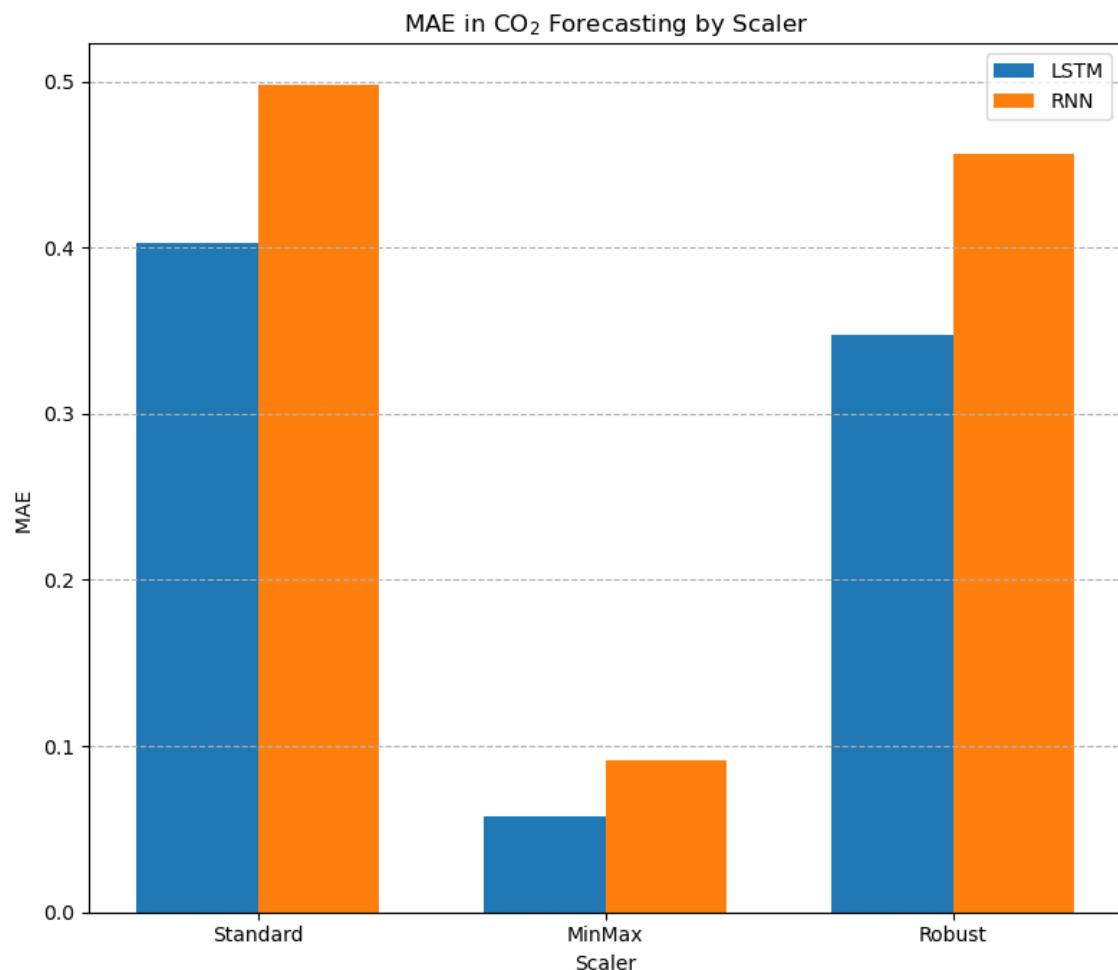


Figure 15. Comparison of the MAE for CO₂ predictions using different scaling techniques, highlighting MinMaxScaler's superior performance.

For CO₂ forecasting, the MinMaxScaler again yields the lowest MAE for both LSTM and RNN models. The LSTM model achieves a remarkable MAE of 0.0578 with the MinMaxScaler, which is significantly lower than with the StandardScaler (0.4030) and RobustScaler (0.3471). Similarly, the RNN model performs best with the MinMaxScaler, achieving an MAE of 0.0914, which outperforms the StandardScaler (0.4978) and RobustScaler (0.4568).

The superior performance of the MinMaxScaler in CO₂ forecasting indicates that scaling the features to a fixed range enhances the learning capability of both LSTM and RNN models. This could be due to the improved gradient descent behavior when the input features are confined within a smaller range, leading to more stable and faster convergence during training.

These findings provide valuable insights for optimizing the preprocessing steps in time series forecasting tasks, guiding the selection of appropriate scaling techniques to improve the predictive accuracy of LSTM and RNN models.

5.2.4 Activation function

The selection of activation functions is a critical aspect of designing neural network models, particularly in the context of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks used for time series forecasting. Activation functions introduce non-linearity into the model, enabling it to capture complex patterns and dependencies in the data. They determine how the weighted sum of the input is transformed into the output of a node or a layer in the network. This transformation is essential because, without non-linear activation functions, the neural network would not be able to perform more complex tasks beyond linear separability (“Why is activation function important for LSTM and RNN models?”, answer by ChatGPT, 13 May 2024).

Activation functions such as Sigmoid, Tanh, and ReLU play a pivotal role in determining the performance of neural networks by introducing non-linearities that enable the model to learn complex patterns from the data. The Sigmoid function, for example, maps the input values to a range between 0 and 1, which can be useful for models that require a probability interpretation of the output. The Tanh function, on the other hand, scales the input to a range between -1 and

1, often leading to better performance in practice due to its zero-centered output, which can help in converging the gradients during backpropagation. ReLU (Rectified Linear Unit) is another widely used activation function that introduces non-linearity by outputting zero for negative input values and the input value itself for positive input values. ReLU is particularly popular due to its simplicity and its ability to mitigate the vanishing gradient problem, allowing for faster and more effective training of deep networks.

The choice of activation function can significantly impact the model's ability to capture intricate temporal dependencies, as discussed by Hochreiter and Schmidhuber [4] and LeCun et al. [16]. In the context of time series forecasting, such as temperature and CO₂ level predictions, the ability of the activation function to maintain and propagate gradients during training is crucial. This ensures that the model can learn long-term dependencies and relationships within the data. For instance, the Tanh and Sigmoid functions are effective for capturing smooth and continuous transitions, which are often present in temperature time series data. However, their tendency to saturate at extreme values can sometimes slow down the learning process. ReLU, with its non-saturating property, allows for the propagation of gradients without the risk of them vanishing, which is beneficial for deeper networks. This characteristic makes it particularly suitable for datasets with high variability and noise, such as CO₂ levels, where the ability to respond dynamically to changes is essential.

In this section, we analyze the sensitivity of LSTM and RNN models to different activation functions (Tanh, Sigmoid, and ReLU) for the tasks of temperature and CO₂ forecasting. By comparing the Mean Absolute Error (MAE) across these activation functions, we can assess how each function influences the model's ability to learn and generalize from the data. Understanding these impacts helps in selecting the most appropriate activation function for specific forecasting tasks, thereby improving the model's accuracy and robustness.

Table 9. Sensitivity results of activation function for temperature

Models	Activation Function	MAE	Number of Features	Number of Neurons	Number of Layers	Scaler
LSTM	Tanh	0.0527	4	100	2	MinMax
	Sigmoid	0.0358				
	ReLU	0.0415				
RNN	Tanh	0.1629	3	10	1	MinMax
	Sigmoid	0.0889				
	ReLU	0.0616				

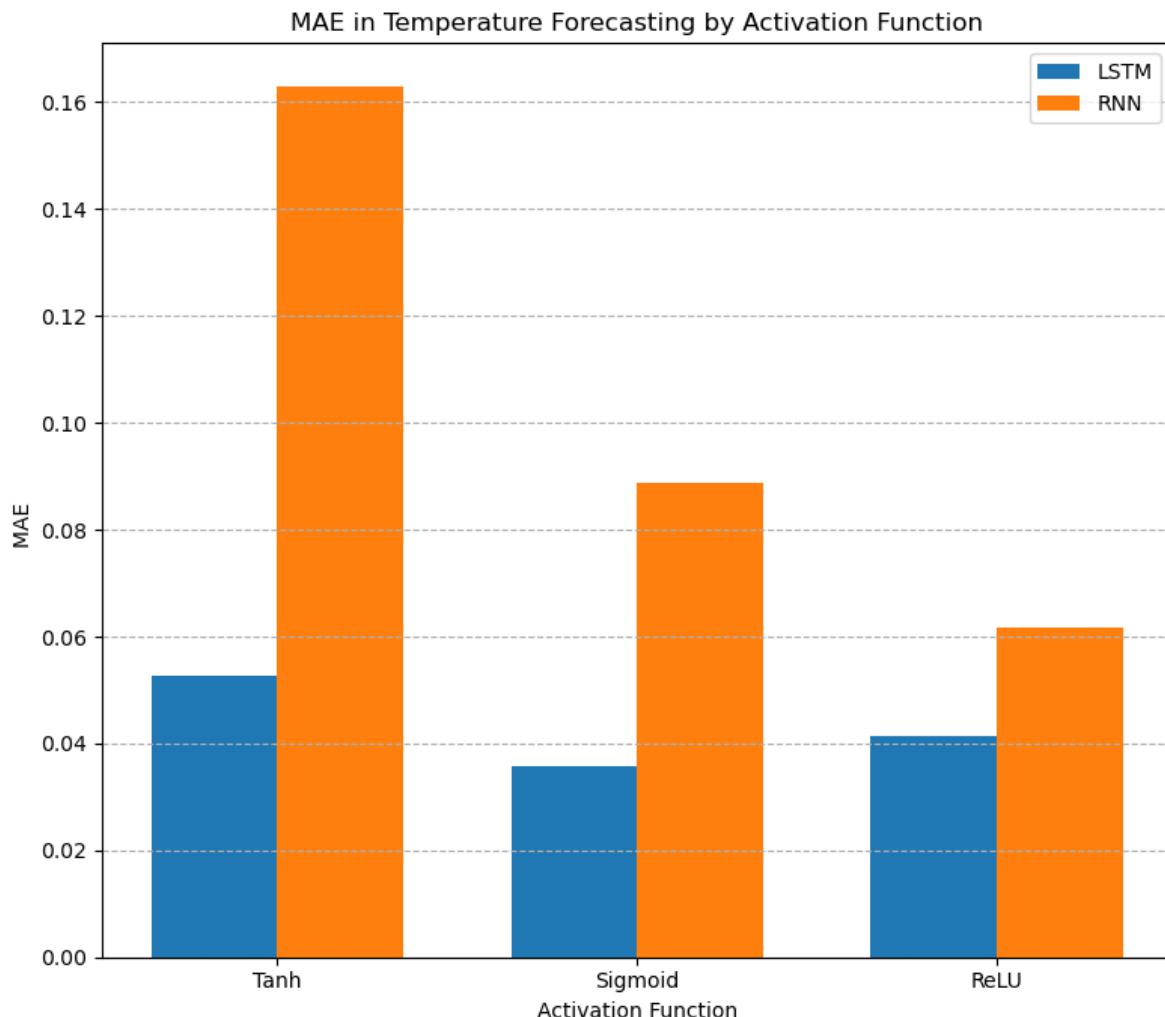


Figure 16. Evaluation of the impact of different activation functions on the MAE for temperature forecasting.

The sensitivity analysis for temperature forecasting indicates that the Sigmoid activation function outperforms Tanh and ReLU in the LSTM model. Regarding the RNN model, ReLu activation function has best performance. Specifically, the LSTM model achieves the lowest Mean Absolute Error (MAE) of 0.0358 with the Sigmoid activation function, compared to 0.0527 with Tanh and 0.0415 with ReLU. For the RNN model, the ReLu function results in an MAE of 0.0616, which is significantly lower than Tanh (0.1629) and Sigmoid (0.0889).

The superior performance of the Sigmoid and ReLu functions in temperature forecasting can be attributed to their ability to handle gradient descent effectively, particularly in capturing the subtle variations and trends in temperature data. The results suggest that the non-linear transformations introduced by the Sigmoid and ReLu functions enable better learning of temperature patterns, thereby enhancing the predictive accuracy of both LSTM and RNN models.

Table 10. Sensitivity results of activation function for CO₂

Models	Activation Function	MAE	Number of Features	Number of Neurons	Number of Layers	Scaler
LSTM	Tanh	0.0578	4	100	2	MinMax
	Sigmoid	0.0650				
	ReLU	0.0633				
RNN	Tanh	0.0914	3	10	1	MinMax
	Sigmoid	0.0613				
	ReLU	0.0818				

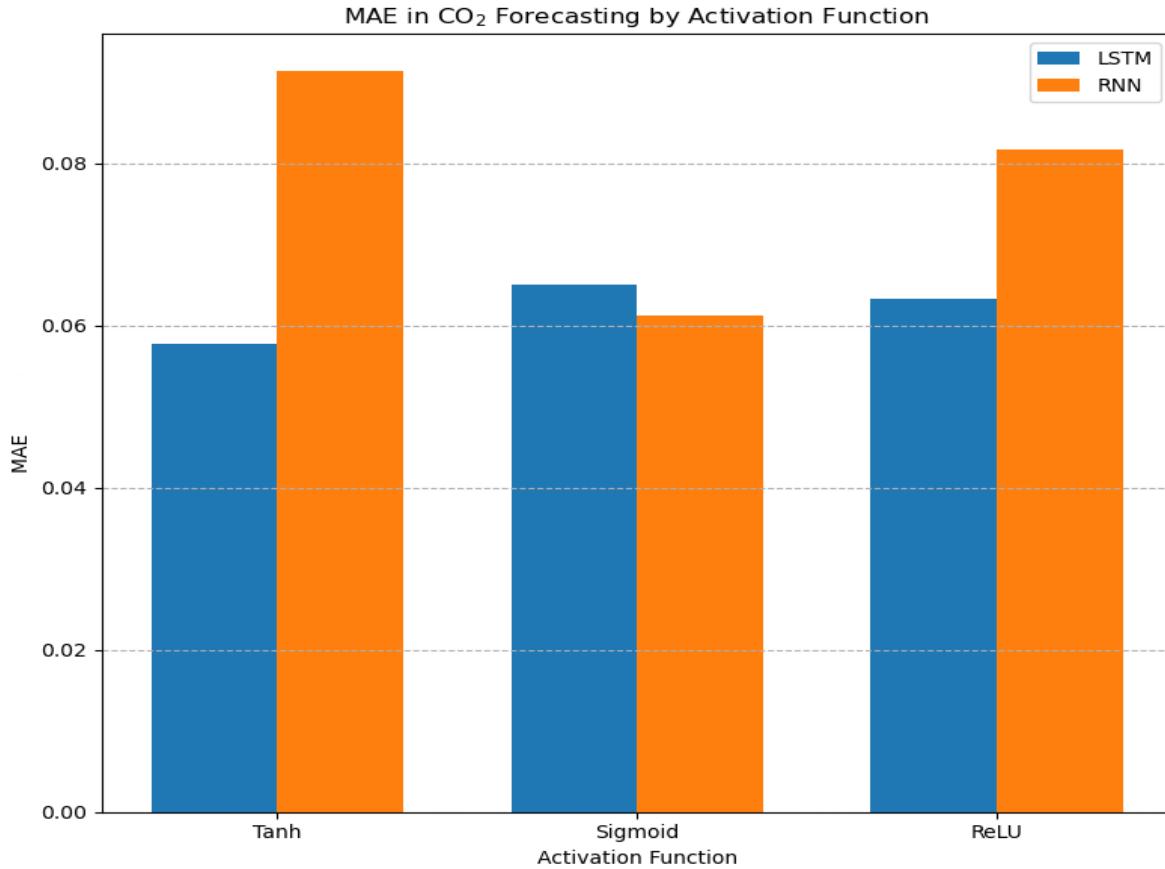


Figure 17. Evaluation of how different activation functions affect the MAE for CO₂ forecasting, indicating varying optimal functions for LSTM and RNN models.

For CO₂ forecasting, the analysis reveals that the performance of activation functions varies between LSTM and RNN models. The RNN model achieves the best performance with the Sigmoid activation function, yielding an MAE of 0.0613, followed by ReLU (0.0818) and Tanh (0.0914). Conversely, the LSTM model shows a slight preference for the Tanh activation function, with an MAE of 0.0578, compared to 0.0633 for Sigmoid and 0.0633 for ReLU.

These results highlight the complex nature of CO₂ time series data and suggest that the choice of activation function can significantly influence model performance. The Tanh function, with its symmetric output and ability to handle both positive and negative inputs effectively, appears to be better suited for capturing the dynamics of CO₂ data in LSTM models. In contrast, the Sigmoid function's capability to provide smoother gradients and better manage the learning process makes it more effective for RNN models.

The findings from this sensitivity analysis underscore the importance of activation function selection in neural network design. By carefully choosing the appropriate activation function, model performance in forecasting tasks can be significantly enhanced.

In summary, for temperature forecasting, the Sigmoid activation function provides the lowest MAE with LSTM model and ReLu provides the lowest MAE with the RNN model, enhancing the model's ability to learn and predict temperature variations accurately. For CO₂ forecasting, the optimal activation function varies between models, with Tanh being more effective for LSTM and Sigmoid for RNN.

5.3 Loss Metrics for Overfitting

Loss metrics play a pivotal role in evaluating the performance of deep learning models by quantifying the difference between the predicted and actual values. In the context of deep learning, loss functions such as Mean Squared Error (MSE) are commonly used to assess model accuracy during training and validation phases [31]. These metrics help in identifying overfitting, a phenomenon where a model performs well on training data but fails to generalize to unseen data, thereby leading to poor predictive performance in real-world applications [45].

In this section, we analyze the loss metrics obtained from the LSTM and RNN models of temperature and CO₂ parameters to evaluate their performance and detect overfitting during temperature forecasting. Monitoring loss values on both the training and validation datasets provides crucial insights into the model's ability to generalize and avoid overfitting.

5.3.1 Temperature

The LSTM and RNN models were performed for temperature by the configurations in Table 2 and 11 respectively. The loss values for the LSTM model across training and validation datasets are illustrated in Figure 18. Initially, the training loss decreases rapidly, indicating that the model is effectively learning the underlying patterns in the data. The validation loss follows a similar trend but starts to diverge slightly as training progresses, which can be an early sign of overfitting. Notably, the validation loss is consistently lower than the training loss, which suggests that the model might be regularizing well or that the validation dataset is inherently

simpler than the training dataset. Given that the validation loss does not significantly increase while the training loss decreases, it is likely that the LSTM model is not overfitting. However, close monitoring and further validation with unseen data are recommended to confirm this behavior.

Table 11. Final sensitivity setup for temperature

Models	Number of Features	Number of Neurons	Number of Layers	Scaler	Activation Function
LSTM	4	100	2	MinMax	Sigmoid
RNN	3	10	1	MinMax	ReLU

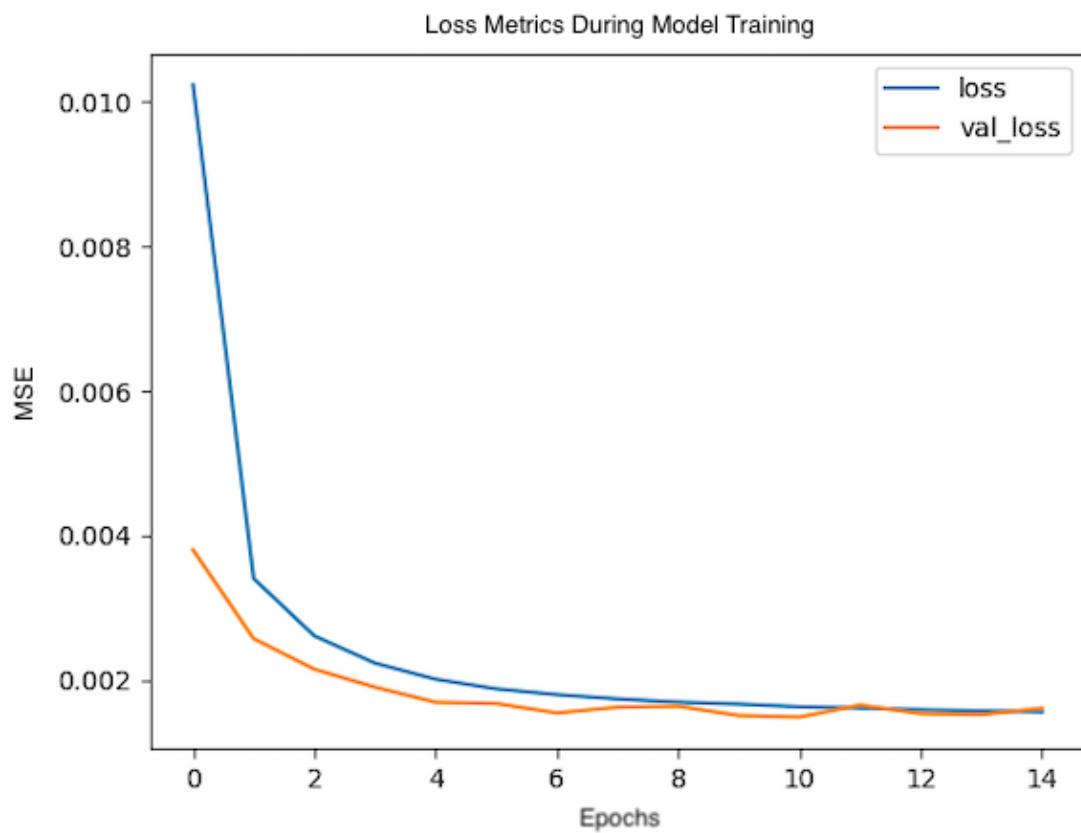


Figure 18. Training and validation loss over epochs for the LSTM model, illustrating the model's learning and potential signs of overfitting in 14 times running.

The RNN model's loss values also provide essential information about its training dynamics, as shown in Figure 19. The training loss decreases over time, while the validation loss displays a more erratic pattern. Initially, the validation loss follows the training loss but begins to fluctuate significantly, suggesting that the RNN model might be struggling to generalize from the training data. The increasing and fluctuating validation loss relative to the training loss indicates potential overfitting. The RNN model might benefit from regularization techniques such as dropout or early stopping to improve its generalization performance.

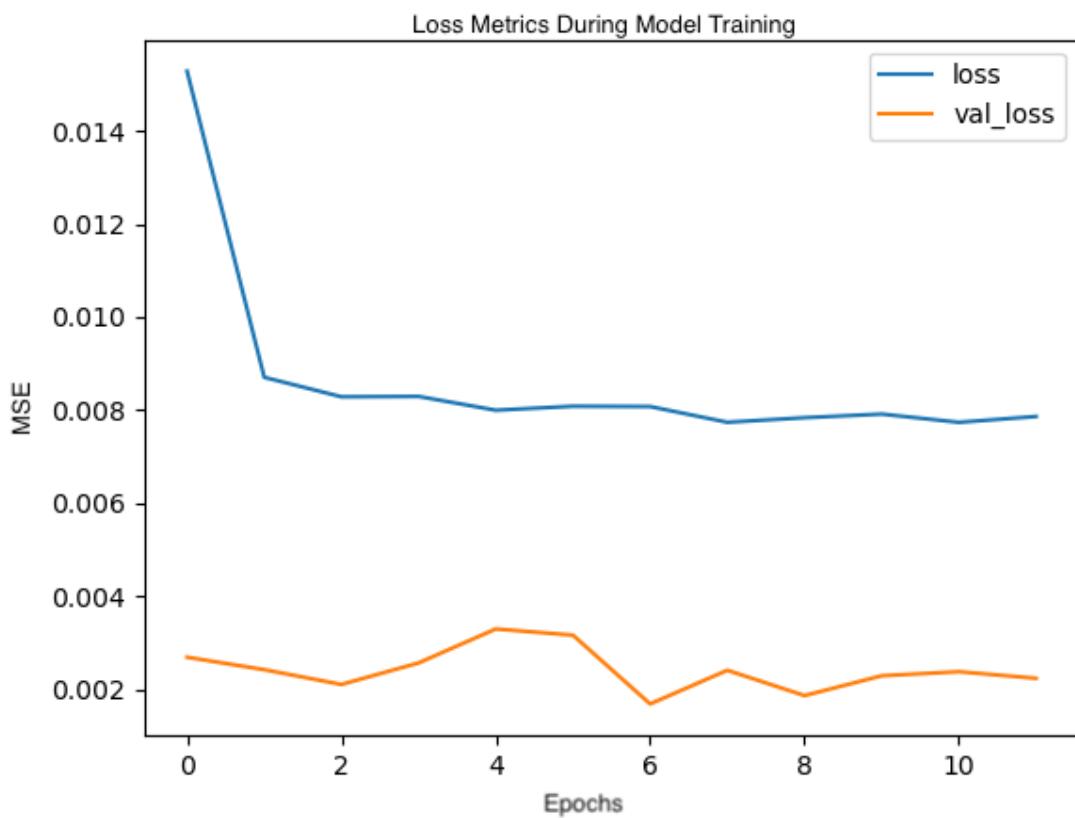


Figure 19. Training and validation loss over epochs for the RNN model in 11 times running, showing the model's learning process and potential overfitting.

Comparing the two models, the LSTM model demonstrates better stability and lower validation loss, indicating it handles temporal dependencies in temperature data more effectively than the RNN model. For practical temperature forecasting, the LSTM model appears more suitable due to its robustness and lower risk of overfitting. The RNN model requires further tuning and regularization to match the performance of the LSTM model.

5.3.2 CO₂

In this section, we analyze the loss metrics obtained from the LSTM and RNN models to evaluate their performance and detect overfitting during CO₂ level forecasting. The CO₂ forecasting using LSTM and RNN models was conducted according to the configurations outlined in Table 2 and Table 12. Monitoring loss values on both the training and validation datasets provides essential insights into the model's ability to generalize and avoid overfitting.

The loss values for the LSTM model across training and validation datasets are illustrated in Figure 20. Initially, the training loss decreases rapidly, indicating that the model is effectively learning the underlying patterns in the data. The validation loss follows a similar trend but starts to diverge slightly as training progresses, which can be an early sign of overfitting. Notably, the validation loss is consistently lower than the training loss, which suggests that the model might be regularizing well or that the validation dataset is inherently simpler than the training dataset. Given that the validation loss does not significantly increase while the training loss decreases, it is likely that the LSTM model is not overfitting. However, close monitoring and further validation with unseen data are recommended to confirm this behavior.

Table 12. Final sensitivity setup for CO₂

Models	Number of Features	Number of Neurons	Number of Layers	Scaler	Activation Function
LSTM	9	100	1	MinMax	Tanh
RNN	8	100	1	MinMax	Sigmoid

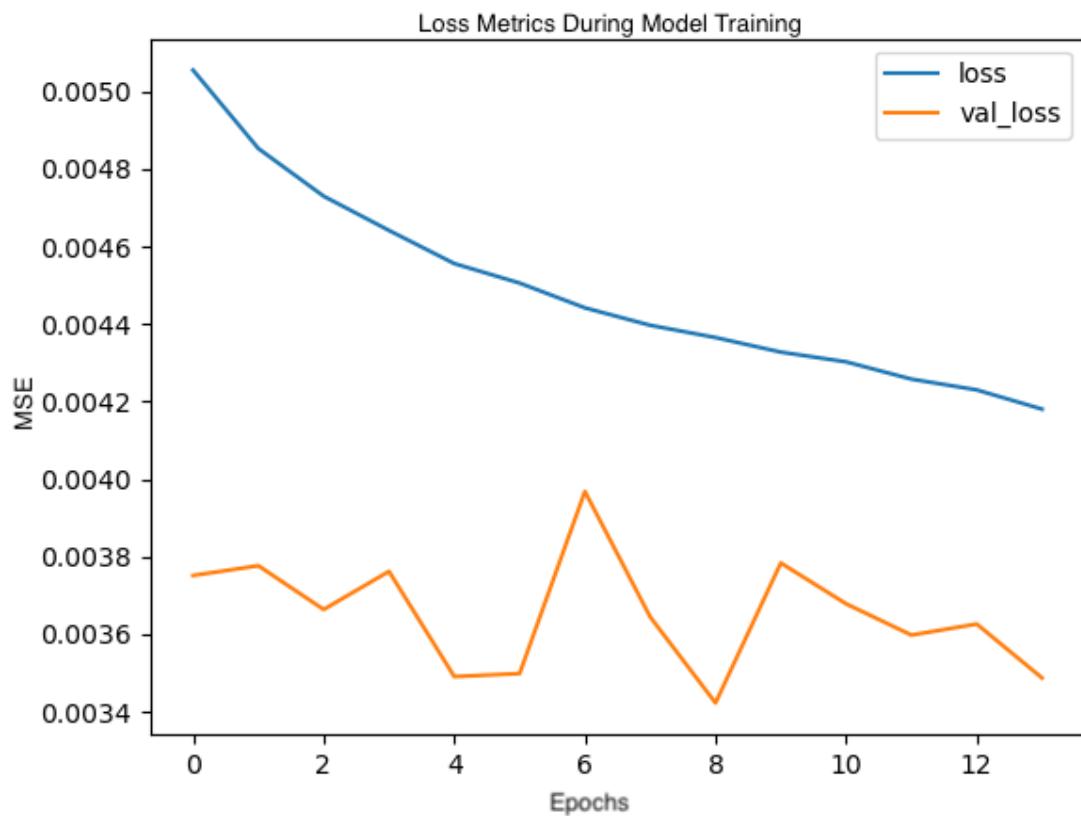


Figure 20. Training and validation loss over epochs for the LSTM model in 13 times running, illustrating the model's learning and potential signs of overfitting for CO₂.

The RNN model's loss values also provide essential information about its training dynamics, as shown in Figure 21. The training loss decreases over time, while the validation loss displays a more unstable pattern. Initially, the validation loss follows the training loss but begins to fluctuate significantly, suggesting that the RNN model might be struggling to generalize from the training data. The increasing and fluctuating validation loss relative to the training loss indicates potential overfitting. The RNN model might benefit from regularization techniques such as dropout or early stopping to improve its generalization performance.

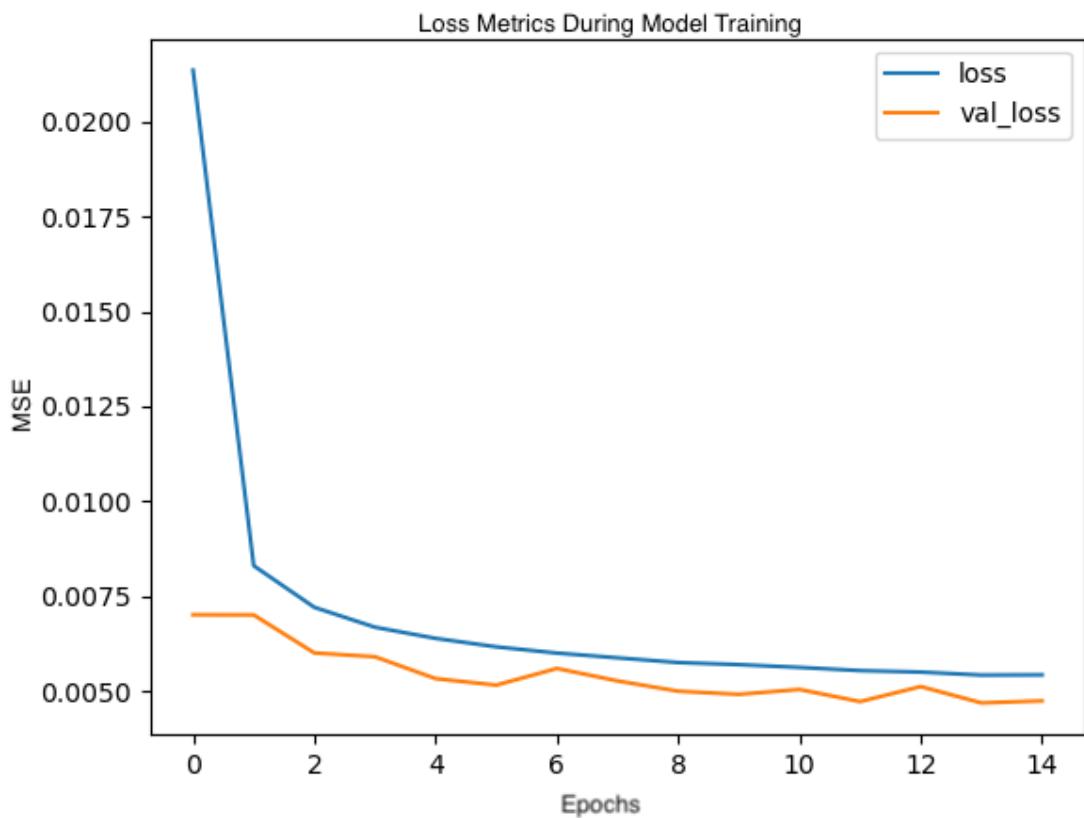


Figure 21. Training and validation loss over epochs for the RNN model in 14 times running, showing the model's learning process and potential overfitting for CO₂.

Comparing the two models, the LSTM model demonstrates better stability and lower validation loss, indicating it handles temporal dependencies in CO₂ data more effectively than the RNN model. For practical CO₂ forecasting, the LSTM model appears more suitable due to its robustness and lower risk of overfitting. The RNN model requires further tuning and regularization to match the performance of the LSTM model.

6 Conclusion

This thesis has provided a comprehensive analysis of the capabilities and performance of Long Short-Term Memory (LSTM) models and Recurrent Neural Networks (RNN) in forecasting temperature and CO₂ levels. The investigation utilized historical meteorological data to predict future values, emphasizing the superiority of LSTM models over traditional RNNs due to their architectural advantages and ability to capture long-term dependencies.

LSTM models consistently outperformed RNNs in terms of predictive accuracy, largely due to their ability to effectively address the vanishing gradient problem, which is common in RNNs. The sensitivity analysis revealed complex interactions between the number of input features and model performance. While increasing the number of input features did not always lead to better performance, careful selection and optimization of features were crucial for improving model accuracy. The MinMaxScaler was found to be the most effective scaling technique, significantly enhancing model performance compared to StandardScaler and RobustScaler. The choice of activation function had a notable impact on model performance, with the Sigmoid activation function being optimal for temperature forecasting with LSTM models, and ReLU being optimal for temperature forecasting with RNN models. For CO₂ forecasting, Tanh was more effective for LSTM models, and Sigmoid was better for RNNs.

The best configurations identified for each model are as follows: For LSTM in temperature forecasting, the optimal configuration was using four input features (Temperature (°C), Dewpoint Temperature (°C), Relative Humidity (%), and Radiation (W/m²)), 100 neurons, two layers, and the MinMaxScaler with the Sigmoid activation function. For CO₂ forecasting with LSTM, the optimal configuration was nine input features (CO₂ (ppm), Relative Humidity (%) Wind Speed (m/s), Radiation (W/m²), Temperature (°C), Pressure (AGL) (hPa), Precipitation (mm), Wind Direction (Degree (°)) and Dewpoint Temperature (°C)), 100 neurons, one layer, and the MinMaxScaler with the Tanh activation function. The RNN model for temperature forecasting achieved the best results with three input features (Temperature (°C), Dewpoint Temperature (°C), and Relative Humidity (%)), 10 neurons, one layer, and the MinMaxScaler with the ReLU activation function. For CO₂ forecasting, the optimal RNN setup was using eight input features (CO₂ (ppm), Relative Humidity (%) Wind Speed (m/s), Radiation (W/m²), Temperature (°C), Pressure (AGL) (hPa), Precipitation (mm), and Wind Direction (Degree

(°))), 100 neurons, one layer, and the MinMaxScaler with the Sigmoid activation function. In both cases, we used the last five days of data to predict the current day. You can also see the outputs in the Appendix by target variable and by model. The outputs were obtained by considering the setups in Table 11 and Table 12 for temperature and CO₂ respectively.

LSTM models demonstrated superior stability and lower validation loss, making them more robust and better at generalizing from training data to unseen data. This robustness is crucial for practical applications in weather forecasting, where accurate predictions are essential for sectors like agriculture, disaster preparedness, and energy management. The implications of these findings are significant, as LSTM models have the potential to revolutionize weather forecasting by providing more reliable and accurate predictions, leading to better-informed decision-making and enhanced risk mitigation strategies, ultimately contributing to greater societal resilience against weather-related challenges.

Future research should focus on optimizing LSTM model architectures further and exploring the integration of additional meteorological variables to enhance predictive accuracy. Real-time data assimilation techniques could also be explored to create more dynamic and responsive forecasting models. Additionally, developing hybrid models that combine physical-based methods with advanced deep learning approaches may yield further advancements in weather prediction. This study demonstrates the substantial benefits of LSTM models in weather forecasting, providing a solid foundation for future research aimed at improving forecast accuracy and extending the practical applications of deep learning in meteorology. The continued refinement and adoption of LSTM-based approaches promise significant improvements in our ability to predict and respond to weather phenomena, fostering a more resilient and well-prepared society.

7 References

- [1] Gneiting, T., & Raftery, A. E. (2005). Weather forecasting with ensemble methods. *Science*, 310(5746), 248-249.
- [2] Wilks, D. S. (2006). Statistical methods in the atmospheric sciences (Vol. 91). Academic press.
- [3] Lynch, P. (2006). The Emergence of Numerical Weather Prediction: Richardson's Dream. Cambridge: Cambridge University Press.
- [4] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- [5] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2015). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10), 2222-2232.
- [6] Scher S. & Messori G. (2019). Weather and climate forecasting with neural networks: Using general circulation models (GCMs) with different complexities. *Geoscientific Model Development*, 12(7), 2797-2809.
- [7] Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015). Long short-term memory networks for anomaly detection in time series. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 89-98).
- [8] Yau, M. K., Rogers, R. R., & Pearce, R. A. (2013). A Short Course in Cloud Physics (3rd ed.). Oxford: Butterworth-Heinemann.
- [9] Lorenz, E. N. (1963). Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2), 130-141.

- [10] Charney, J., Lorenz, E. N., & Hemler, D. J. (1963). Numerical Integration of the Quasi-Geostrophic Equations for Barotropic and Simple Baroclinic Flows. *Journal of the Atmospheric Sciences*, 20(5), 497-510.
- [11] ECMWF. (2022). "European Centre for Medium-Range Weather Forecasts." [Online]. Available: <https://www.ecmwf.int/>. [Accessed: Jan. 10, 2024].
- [12] NCAR. (2022). "National Center for Atmospheric Research." [Online]. Available: <https://www.ncar.ucar.edu/>. [Accessed: Jan. 10, 2024].
- [13] Purser, R. J. (1977). A Review of Data Assimilation. *Methods in Computational Physics*, 17, 219-324.
- [14] Arizona University. "Weather Forecasting." [Online]. Available: http://www.atmo.arizona.edu/students/courselinks/spring08/atmo336s1/courses/summer11/atmo336/lectures/sec6/weather_forecast.html. [Accessed: Jan. 10, 2024].
- [15] W. Zhang and X. Wang, "Deep Learning for Satellite Image Data in Hurricane Analysis," in IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 2, pp. 1109-1120, Feb. 2020.
- [16] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," in Nature, vol. 521, no. 7553, pp. 436-444, May 2015.
- [17] R. Bose, A. L. Pintar, and E. Simiu, "Data-Based Models for Hurricane Evolution Prediction: A Deep Learning Approach" Thirty-Sixth AAAI Conference on Artificial Intelligence 2022, Vancouver, CA, 2022. Available: [NIST](#).
- [19] Su, Y., Smith, J. A., & Villarini, G. (2024). Physically Explainable Deep Learning for Convective Initiation Nowcasting Using GOES-16 Satellite Observations. *Artificial Intelligence for the Earth Systems*. <https://doi.org/10.1175/AIES-D-23-0098.1>.
- [20] Chen, R.; Zhang, W.; Wang, X. Machine Learning in Tropical Cyclone Forecast Modeling: A Review. *Atmosphere* 2020, 11, 676. <https://doi.org/10.3390/atmos11070676>

- [21] A. Casati, B. Smith, and C. Johnson, "Transforming Weather Forecasting: Applications of Machine Learning Algorithms," *Journal of Applied Meteorology and Climatology*, vol. 58, no. 6, pp. 1325-1338, Jun. 2019.
- [22] W. Zhang, X. Wang, et al., "Predicting Precipitation Patterns Using Support Vector Machines," *Journal of Atmospheric Sciences*, vol. 45, no. 3, pp. 678-690, Mar. 2020.
- [23] J. S. Combinido, J. R. Mendoza, and J. Aborot, "A convolutional neural network approach for estimating tropical cyclone intensity using satellite-based infrared images," 24th International Conference on Pattern Recognition (ICPR), 2018. Available: <https://pubs.aip.org/aip/acp/article/2631/1/020020/2890907>.
- [24] P. Benáček, A. Farda, and P. Štěpánek, "Postprocessing of Ensemble Weather Forecast Using Decision Tree-Based Probabilistic Forecasting Methods," *Weather and Forecasting*, vol. 38, no. 1, pp. 69–82, Jan. 2023. Available: <https://doi.org/10.1175/WAF-D-22-0006.1>.
- [25] Y. Chen, H. Wei, et al., "Decision Tree Models for Wind Speed and Direction Prediction," *Weather and Forecasting*, vol. 38, no. 4, pp. 789-802, Aug. 2022.
- [26] Y. Liu, H. Zhang, et al., "Probabilistic Weather Forecasting Using Naive Bayes Classifiers," *Journal of Climate*, vol. 32, no. 9, pp. 2595-2608, May 2023.
- [27] Eduardo J. Solteiro Pires, José Baptista, "Enhancing Weather Forecasting Integrating LSTM and GA," *Applied Sciences*, 2024. <https://doi.org/10.3390/app14135769>.
- [28] Changchun Cai, Yuan Tao, Tianqi Zhu, and Zhixiang Deng, "Short-Term Load Forecasting Based on Deep Learning Bidirectional LSTM Neural Network," *Applied Sciences*, 2021. <https://doi.org/10.3390/app11178129>.
- [29] "A flexible and lightweight deep learning weather forecasting model," *Applied Intelligence*, Springer, 2023.

- [30] V. Mishra, SMT Agarwal, and Neha Puri, "Comprehensive and comparative analysis of neural network," International Journal of Computer Application, vol. 2, 2018.
- [31] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- [32] LeCun, Y., Bengio, Y., & Hinton, G. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
- [33] Singh, S., Sarkar, S., & Mitra, P. (2017). Leveraging Convolutions in Recurrent Neural Networks for Doppler Weather Radar Echo Prediction. Springer.
- [34] Mishra, R., & Mishra, D.P. (2021). Comparison of Neural Network Models for Weather Forecasting. Springer.
- [35] Li, L., Schmid, W., & Joss, J. (2022). Attention-based Long Short-Term Memory Networks for Weather Forecasting. Journal of Applied Meteorology and Climatology, 61(5), 2987-3001.
- [36] T.T.K. Tran, S.M. Bateni, S.J. Ki, and H. Vosoughifar, "A Review of Neural Networks for Air Temperature Forecasting," Water, vol. 13, 2021.
- [37] Google Earth. (2024). Satellite image of Payerne, Switzerland. Retrieved from <https://earth.google.com/web/>. [Accessed: Jan. 24, 2024].
- [38] Lee, S., Lee, Y., & Son, Y. (2020). Forecasting Daily Temperatures with Different Time Interval Data Using Deep Neural Networks. Applied Sciences, 10(5), 1609. <https://doi.org/10.3390/app10051609>
- [39] Jiang, C., Chen, Y., Chen, S., Bo, Y., Li, W., Tian, W., & Guo, J. (2019). A Mixed Deep Recurrent Neural Network for MEMS Gyroscope Noise Suppressing. Electronics, 8(2), 181. <https://doi.org/10.3390/electronics8020181>
- [40] Wang, H., & Li, L. (2021). Study on the Application of LSTM in Meteorological Element Prediction based on Computer Software. *Journal of Physics: Conference Series*, 1744(2), 022034. <https://doi.org/10.1088/1742-6596/1744/2/022034>

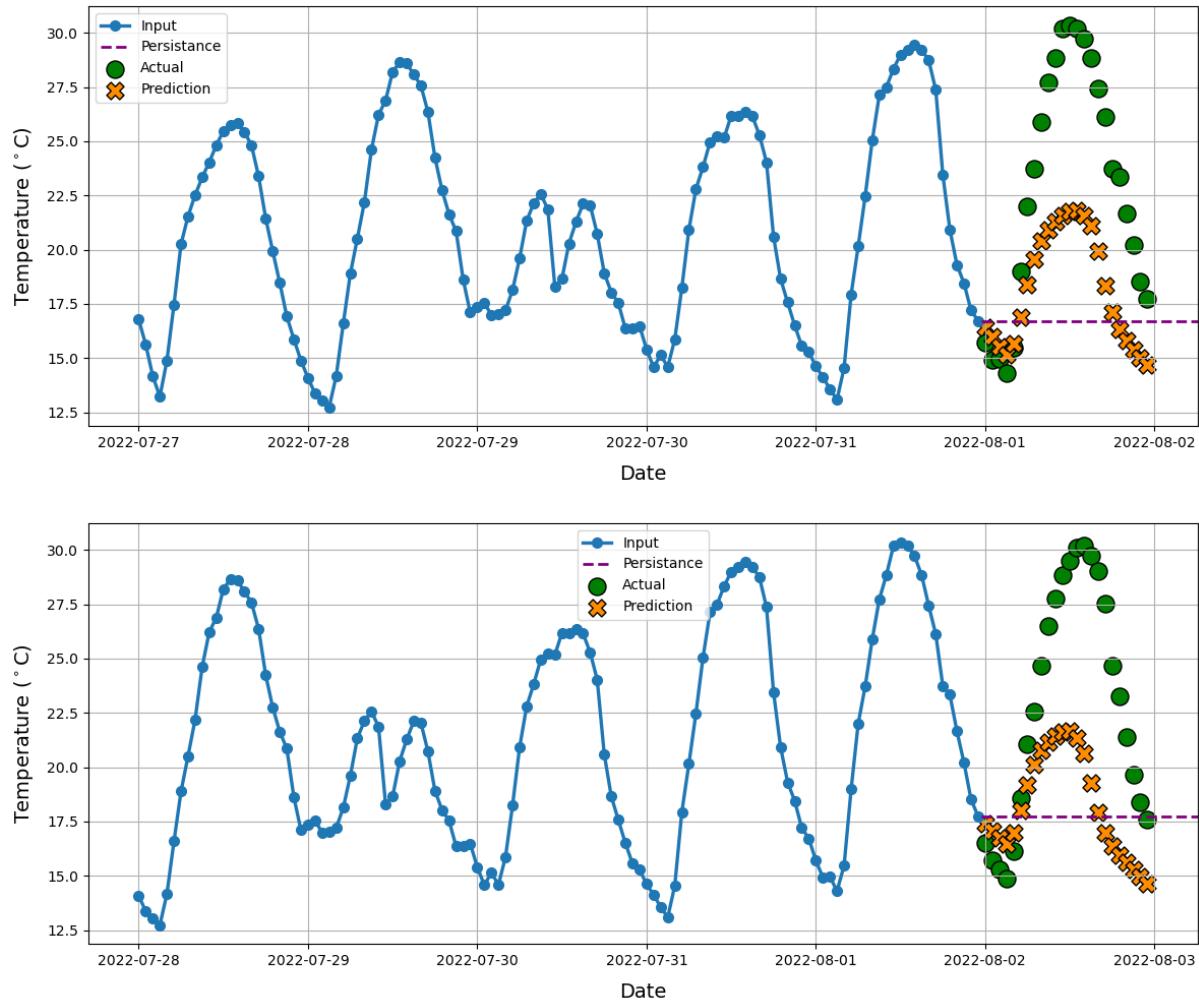
- [41] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281-305, 2012.
- [42] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, vol. 25, pp. 2951-2959, 2012.
- [43] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157-166, 1994.
- [44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097-1105, 2012.
- [45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.

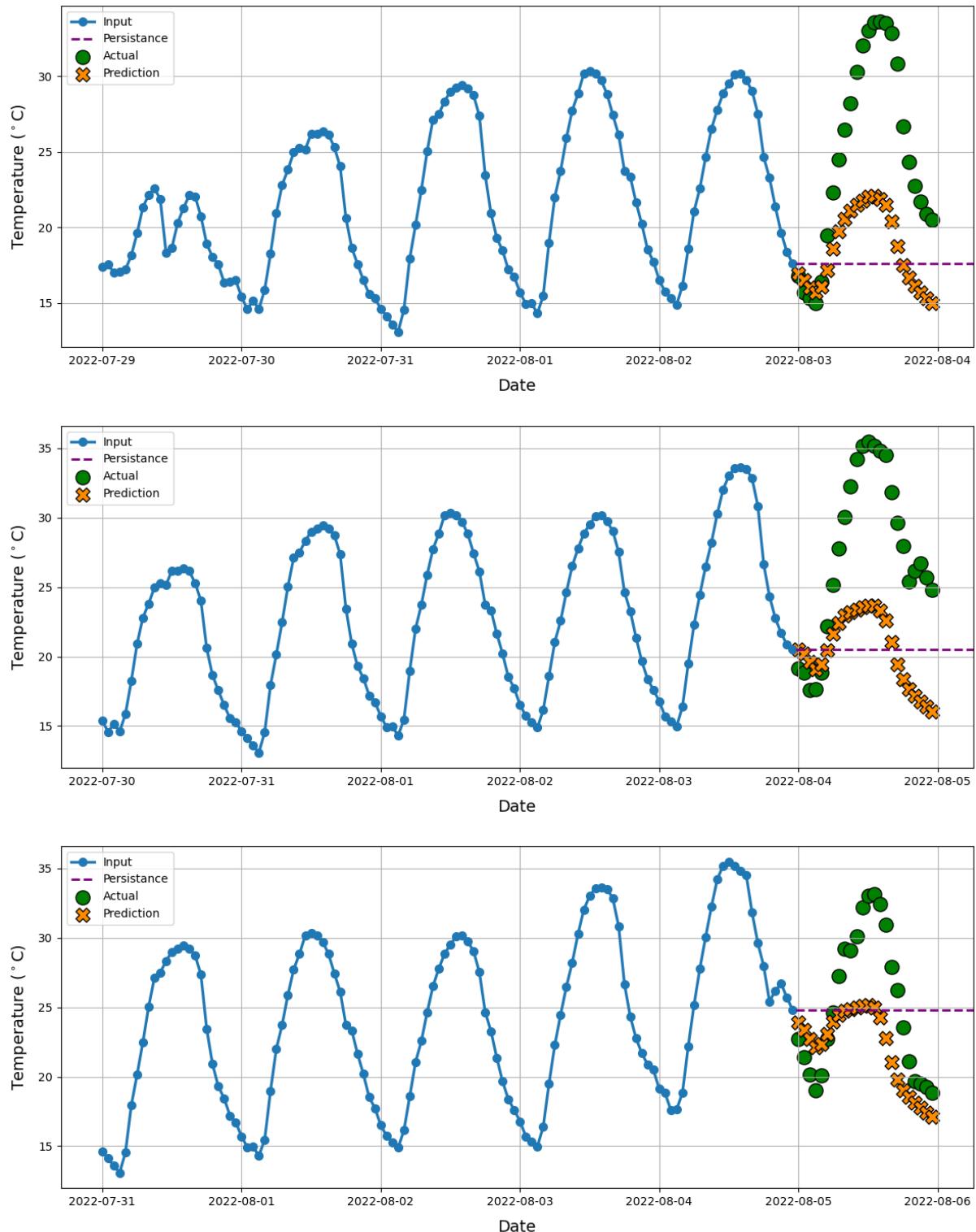
8 Appendix

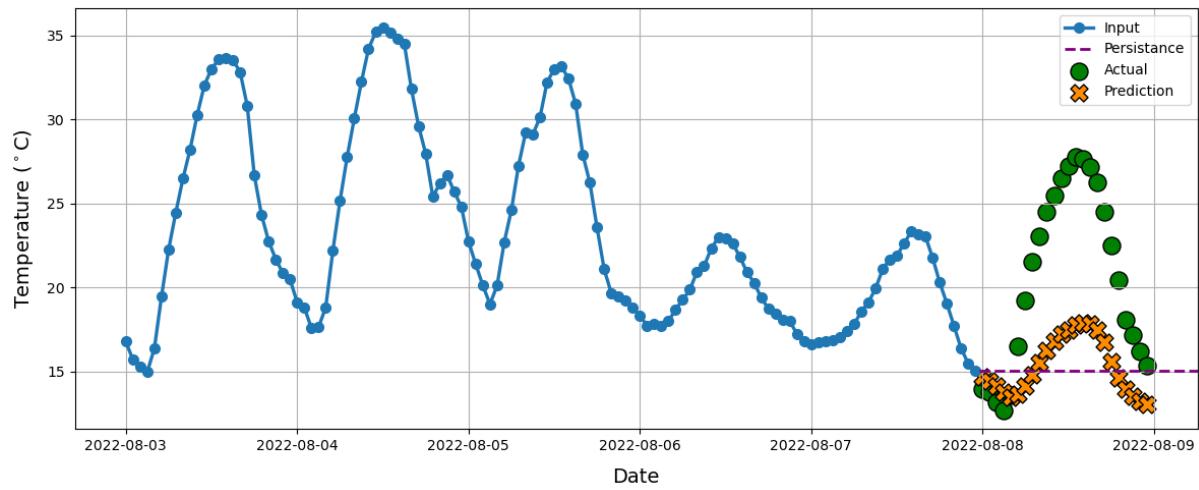
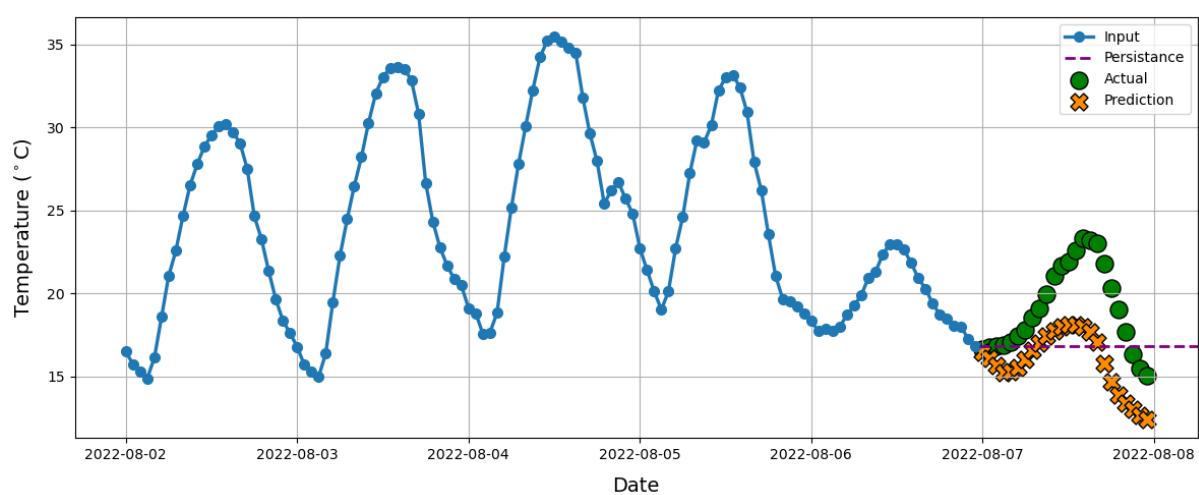
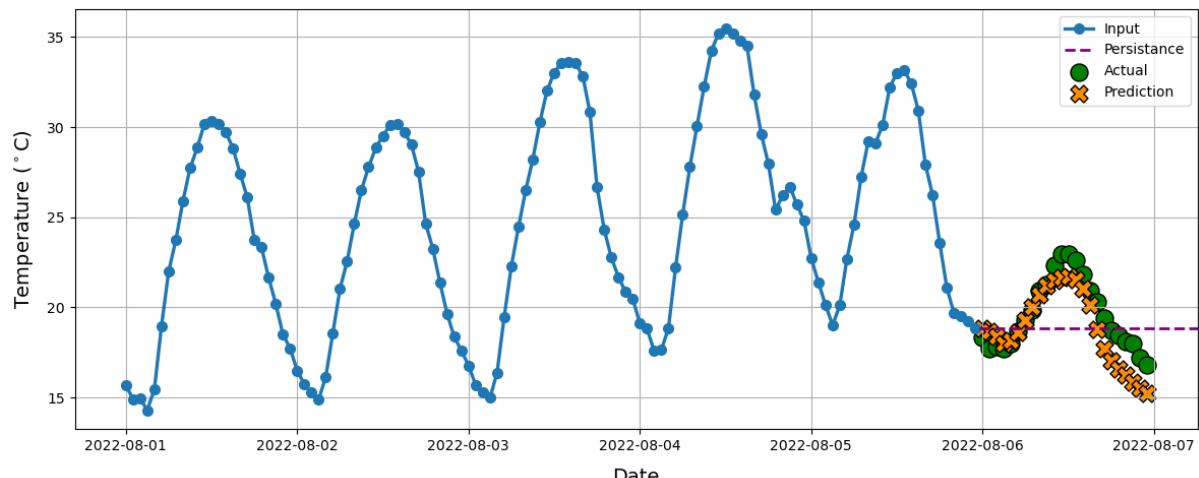
8.1 Forecast Results of Temperature

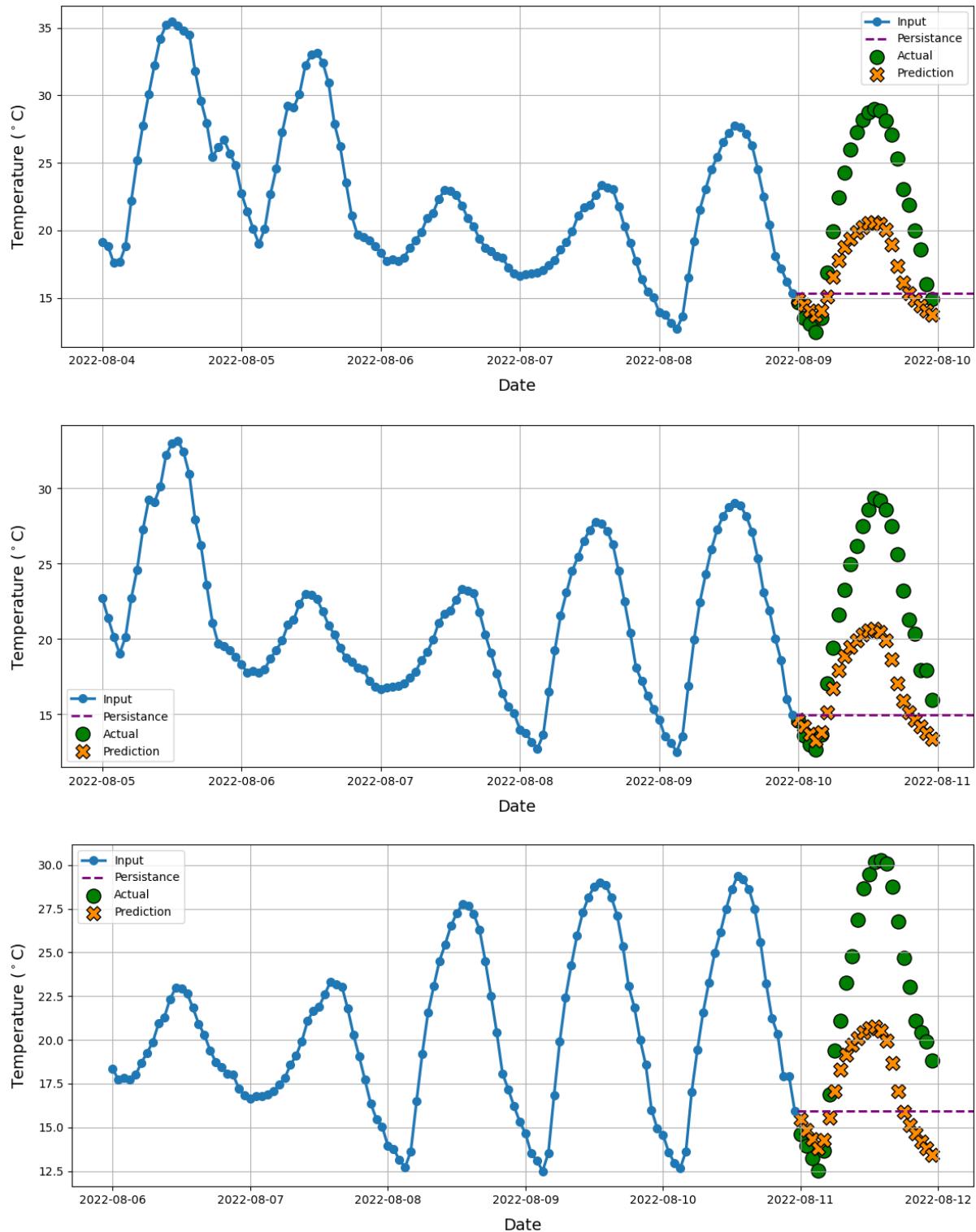
The Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) models were used to forecast the temperature for the entire month of August. Both models utilized the latest configuration displayed in the Table 11, using the last five days of temperature data to predict the current day's temperature.

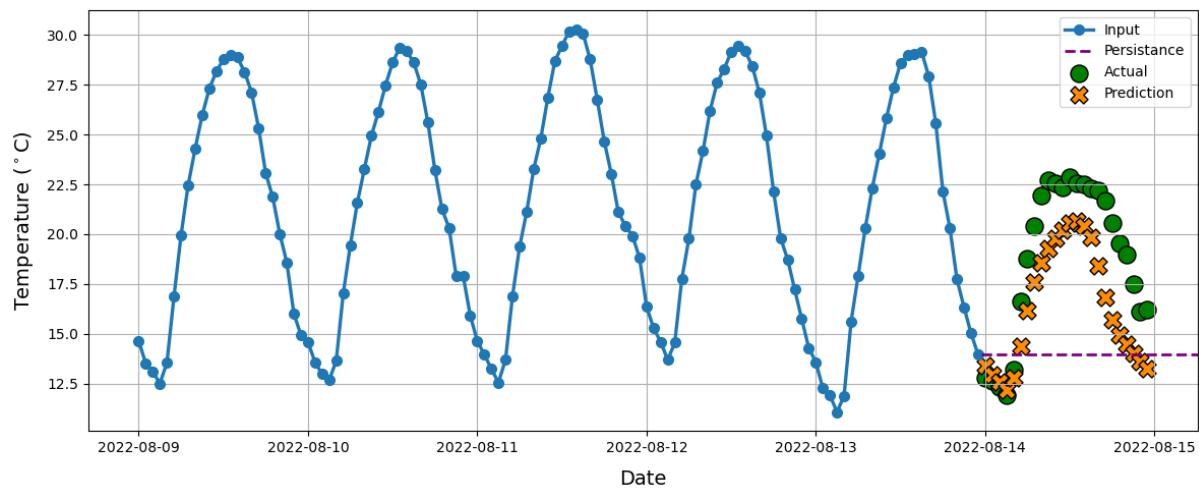
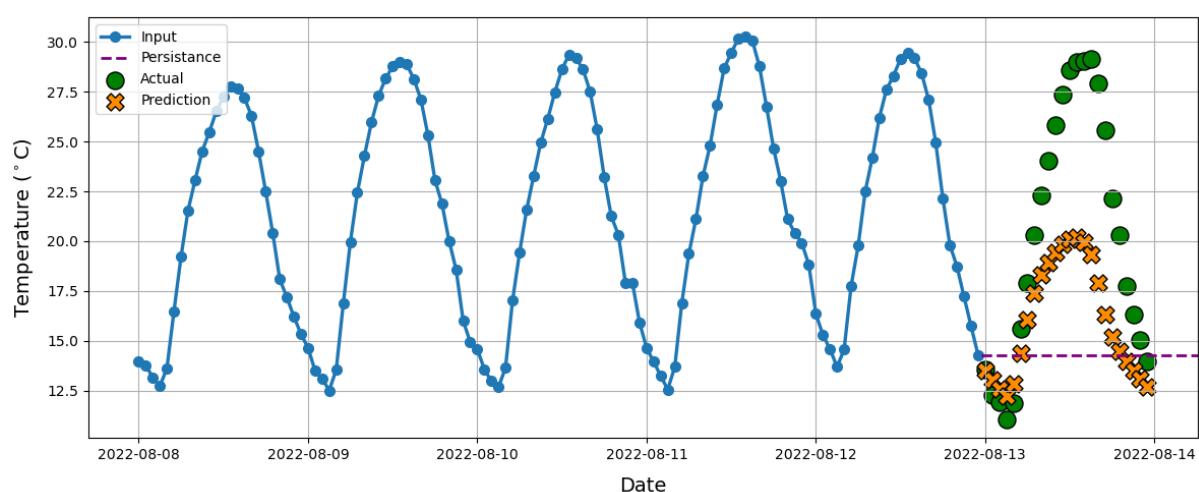
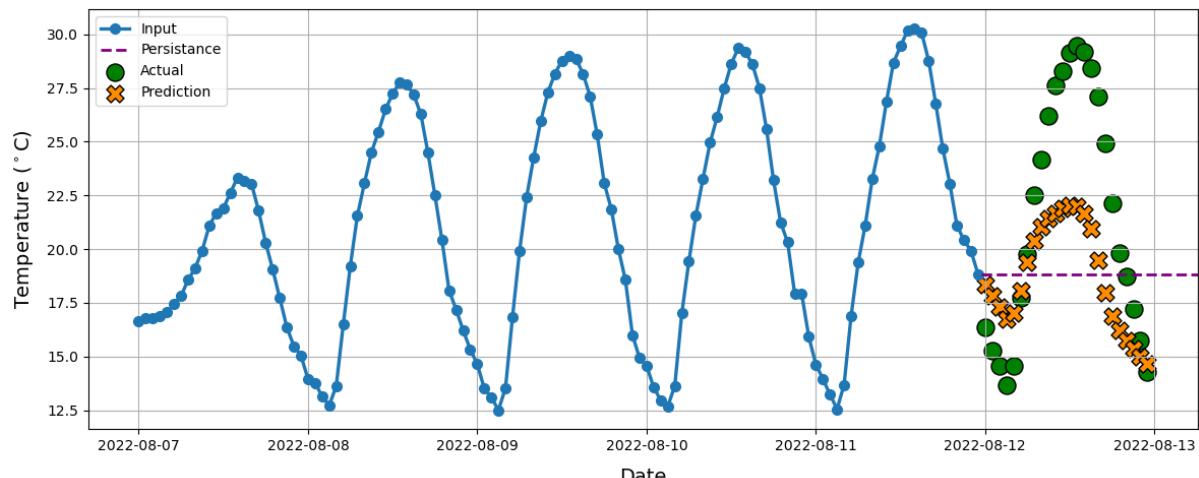
8.1.1 LSTM

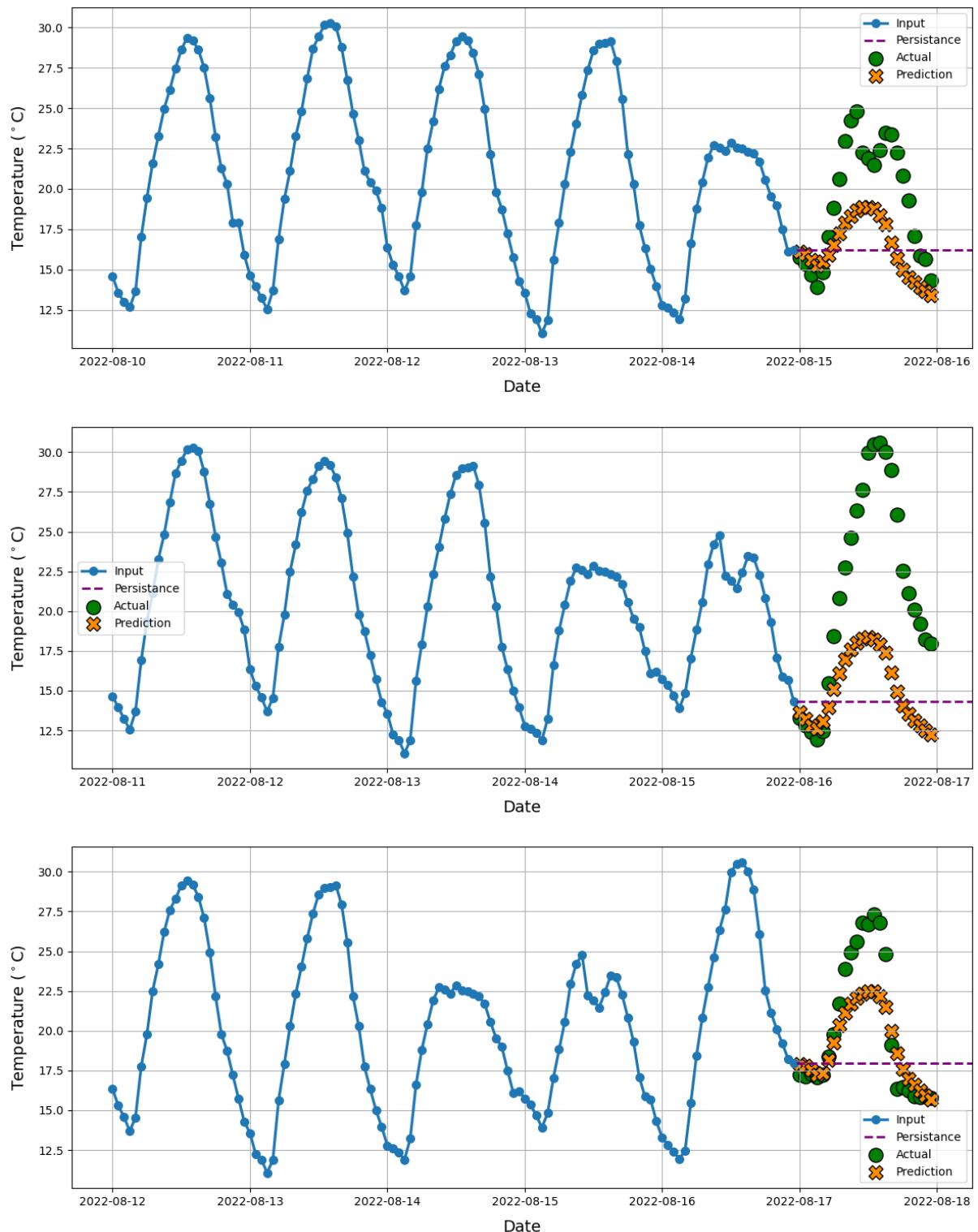


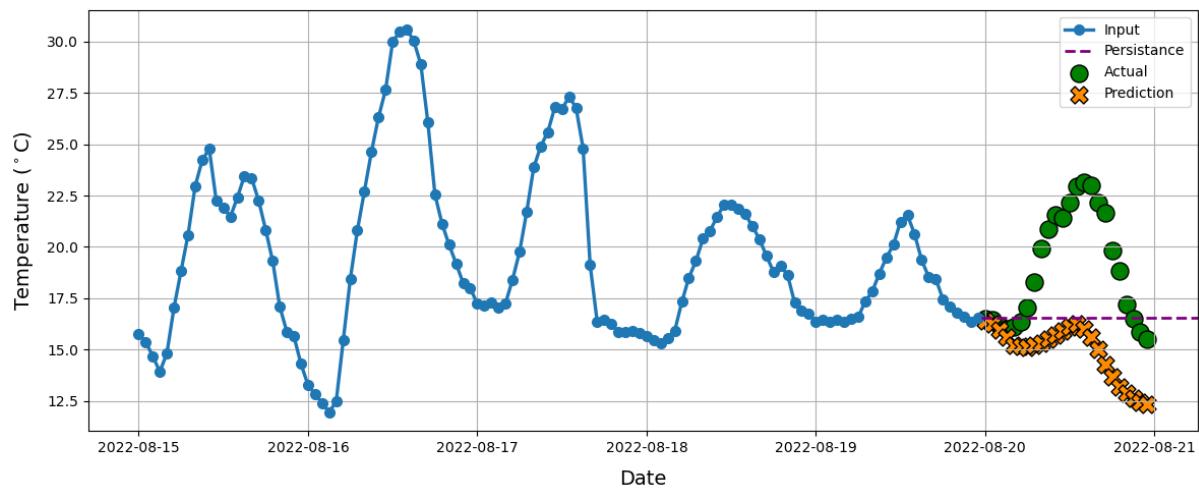
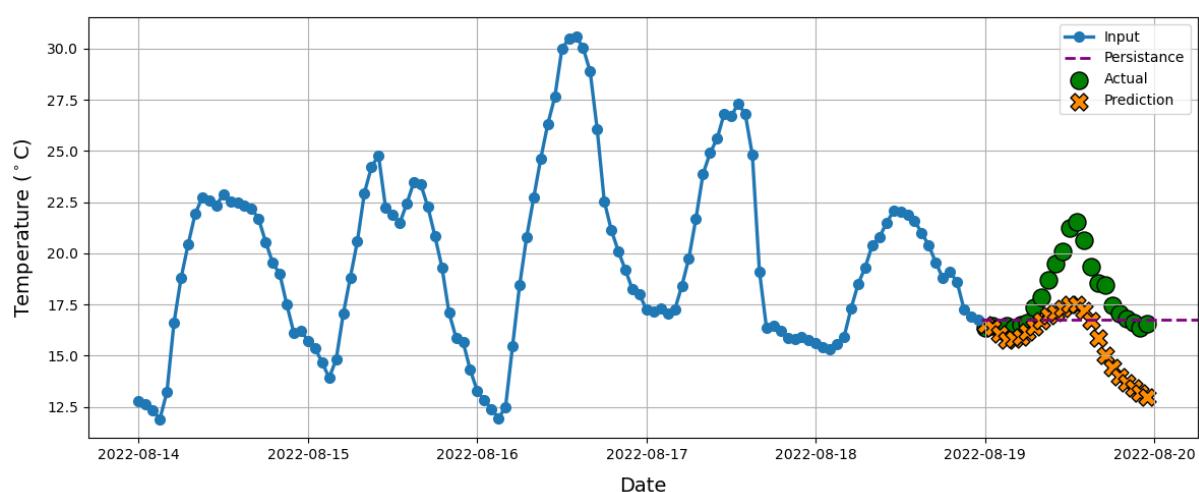
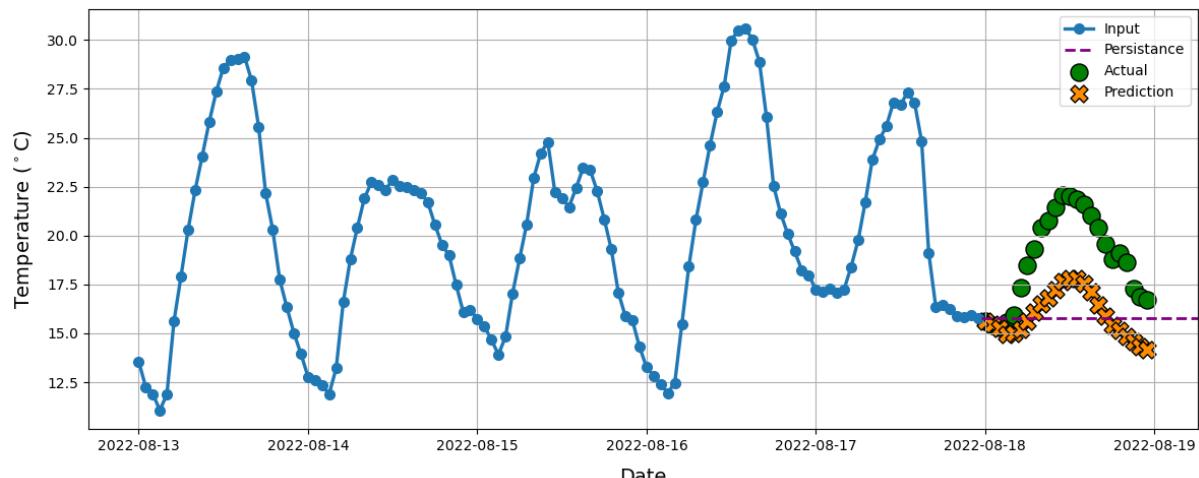


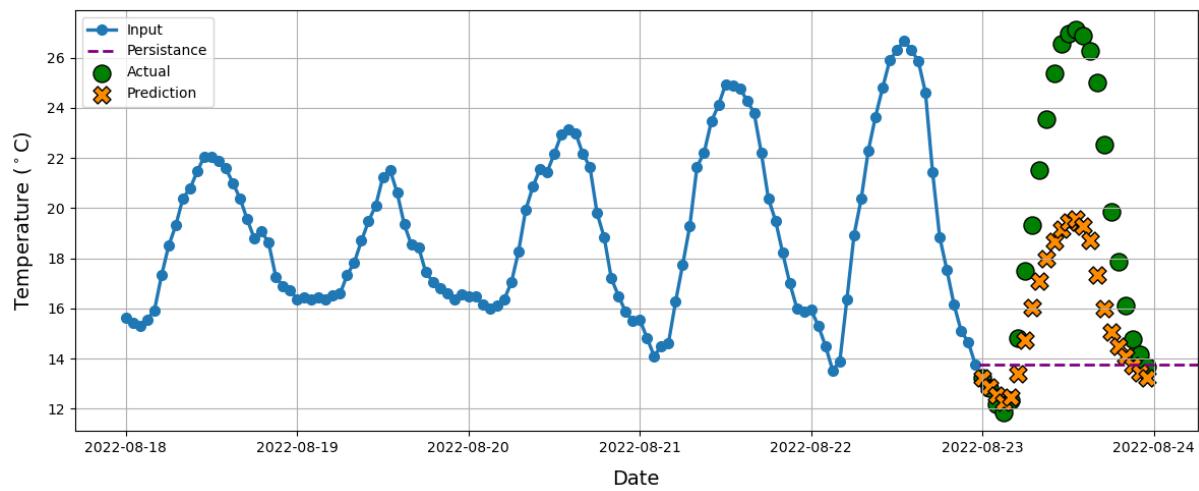
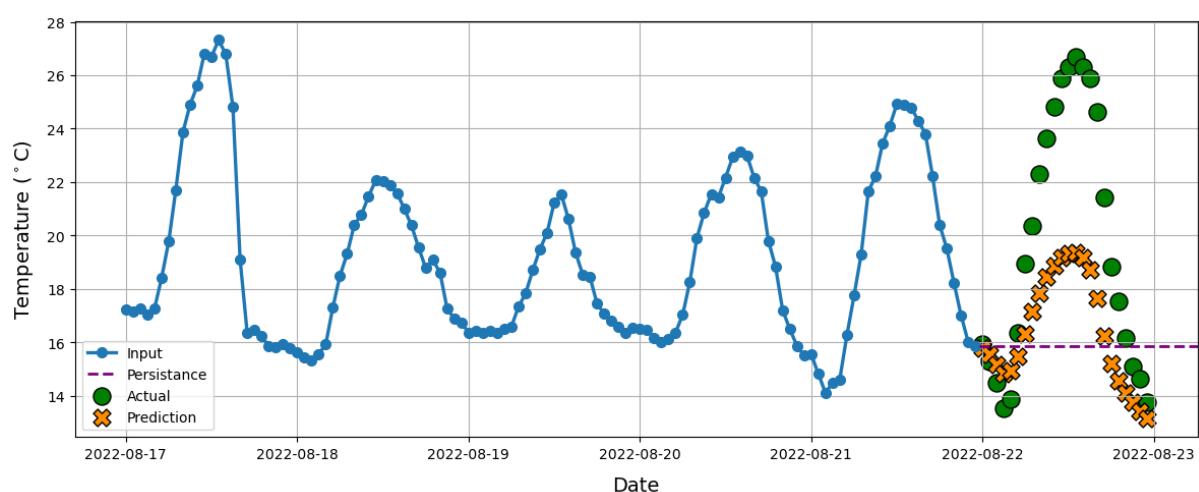
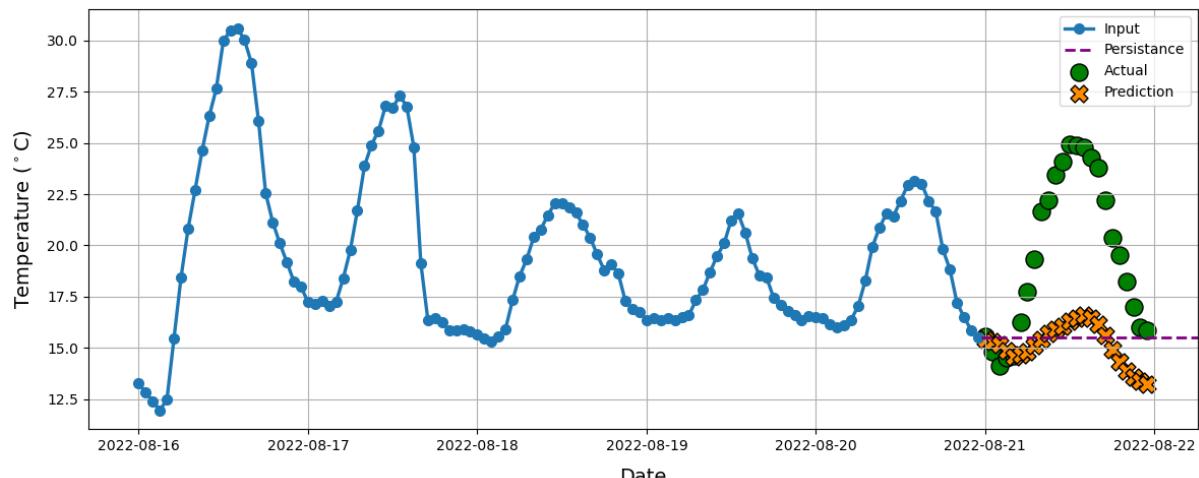


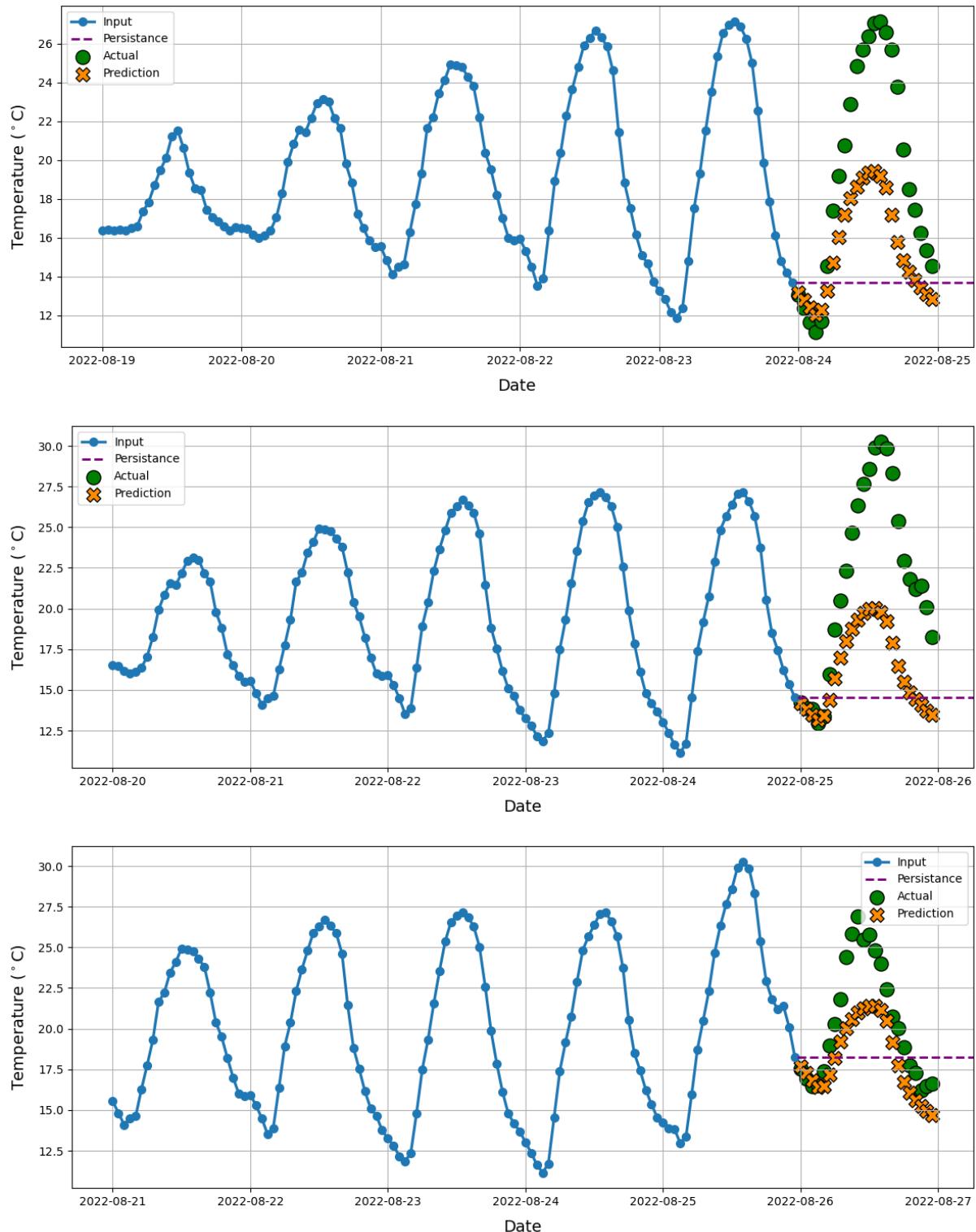


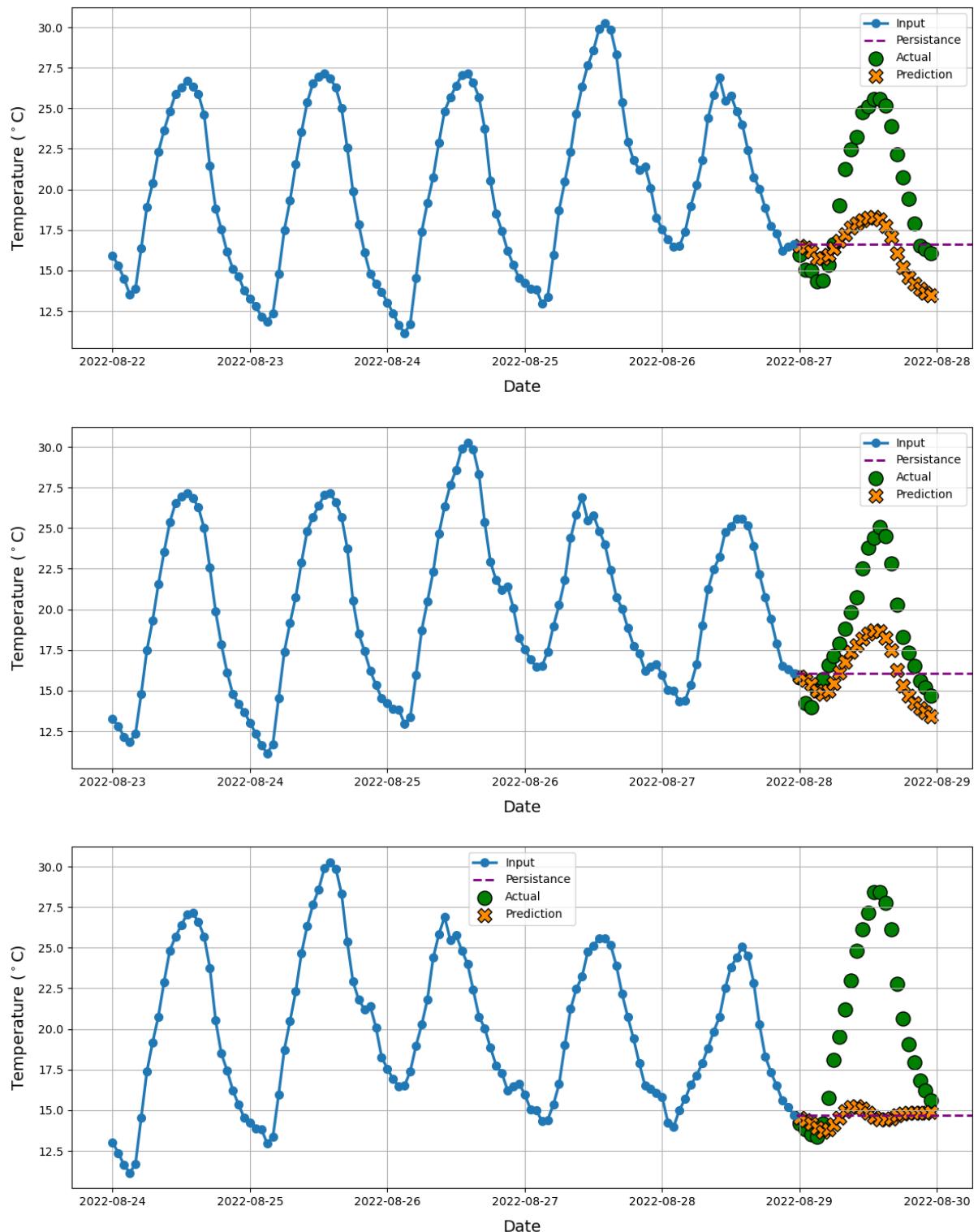


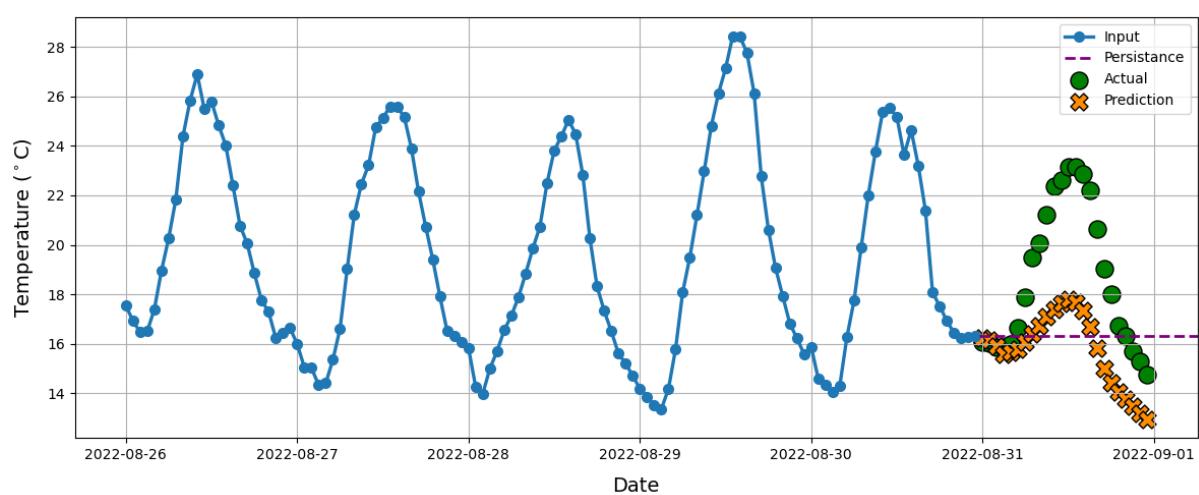
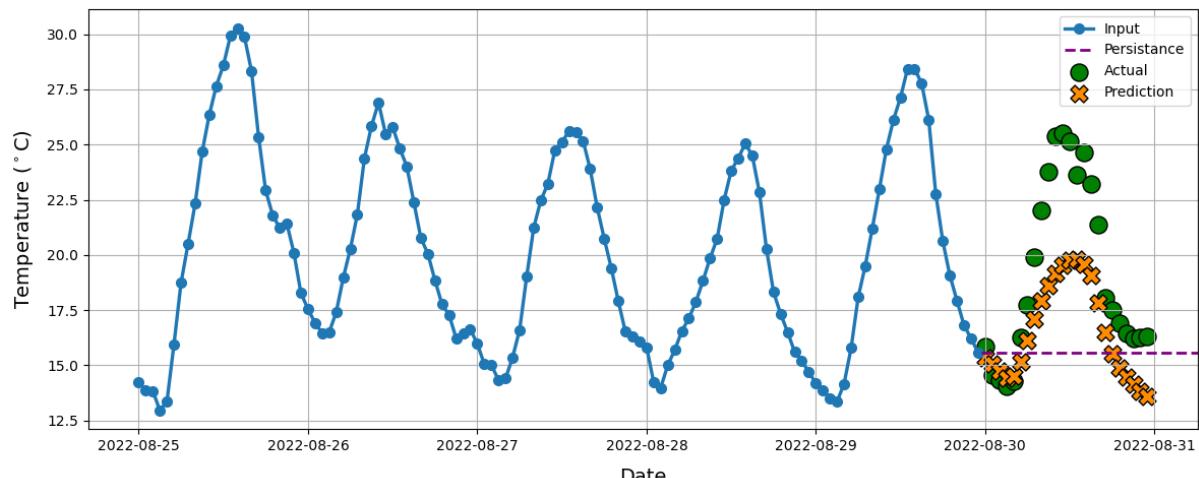




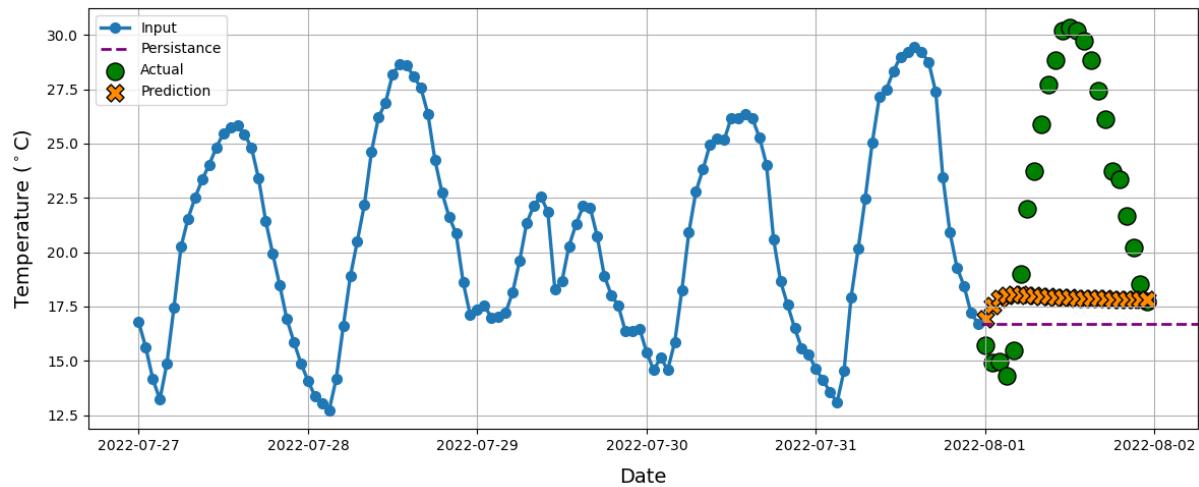


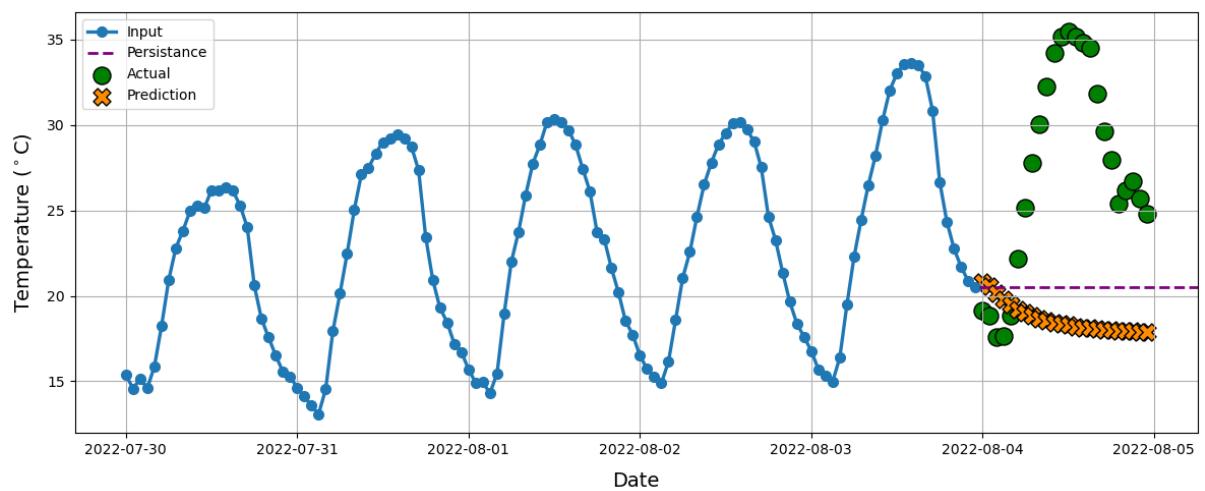
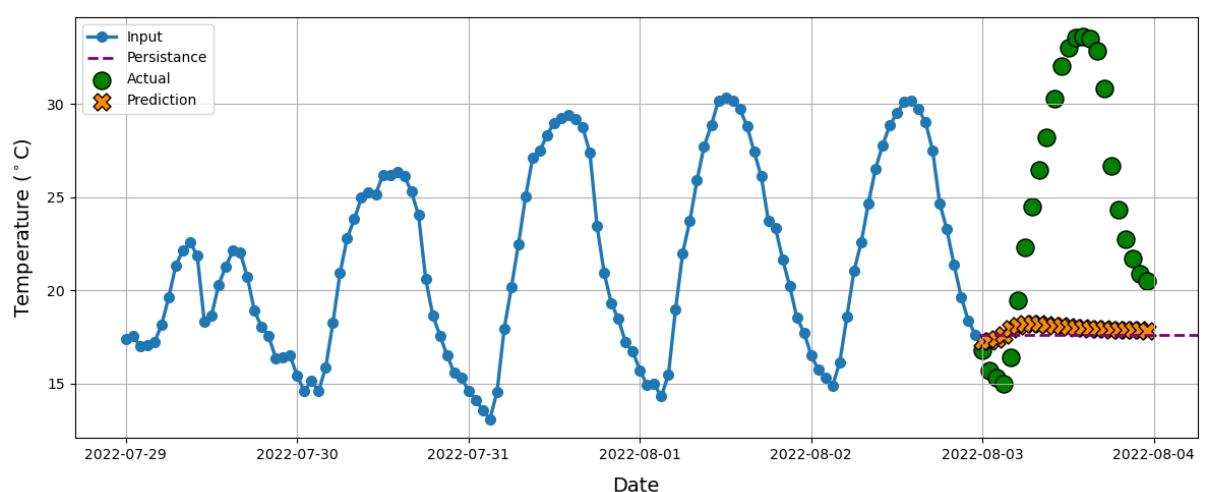
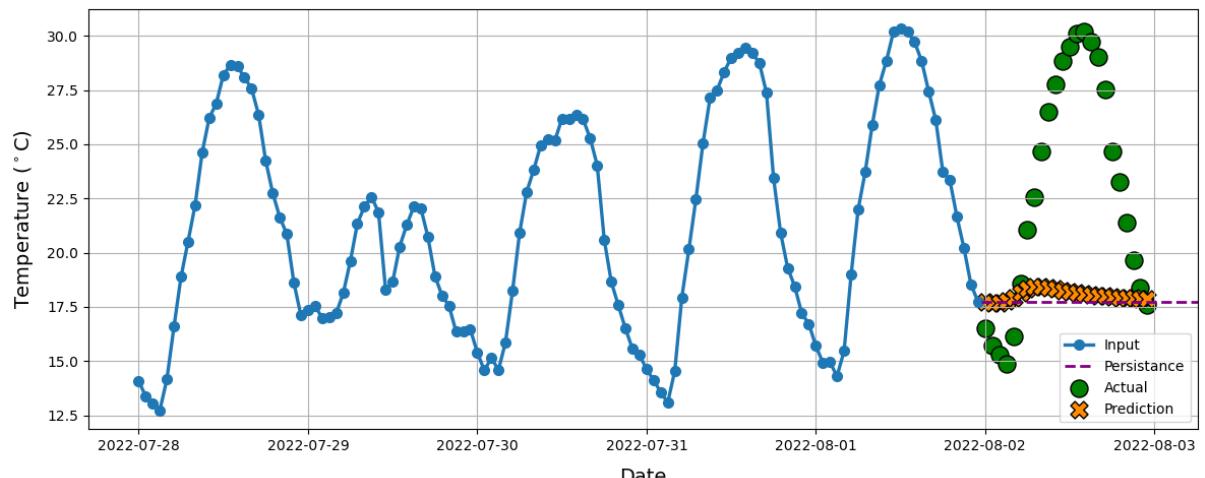


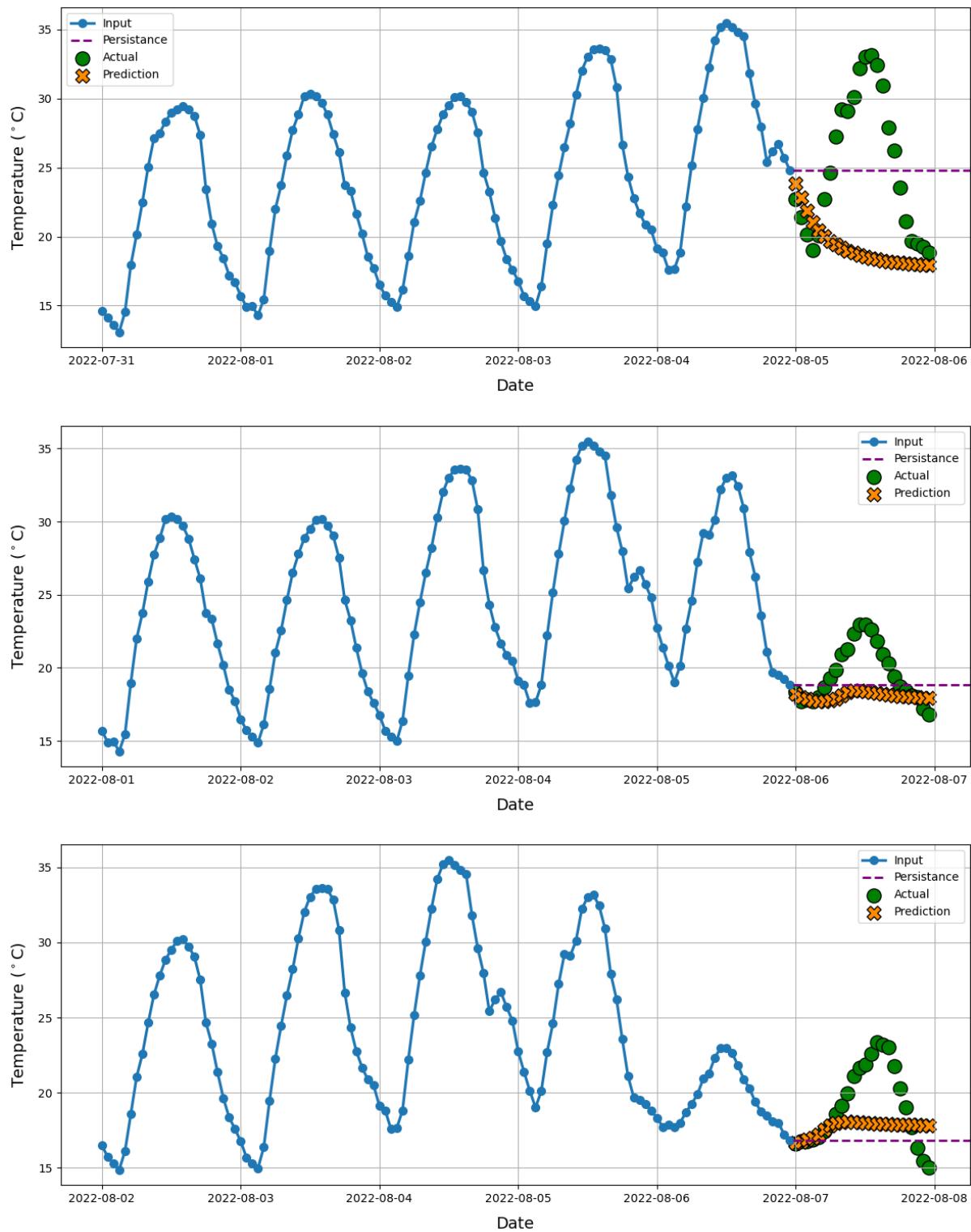


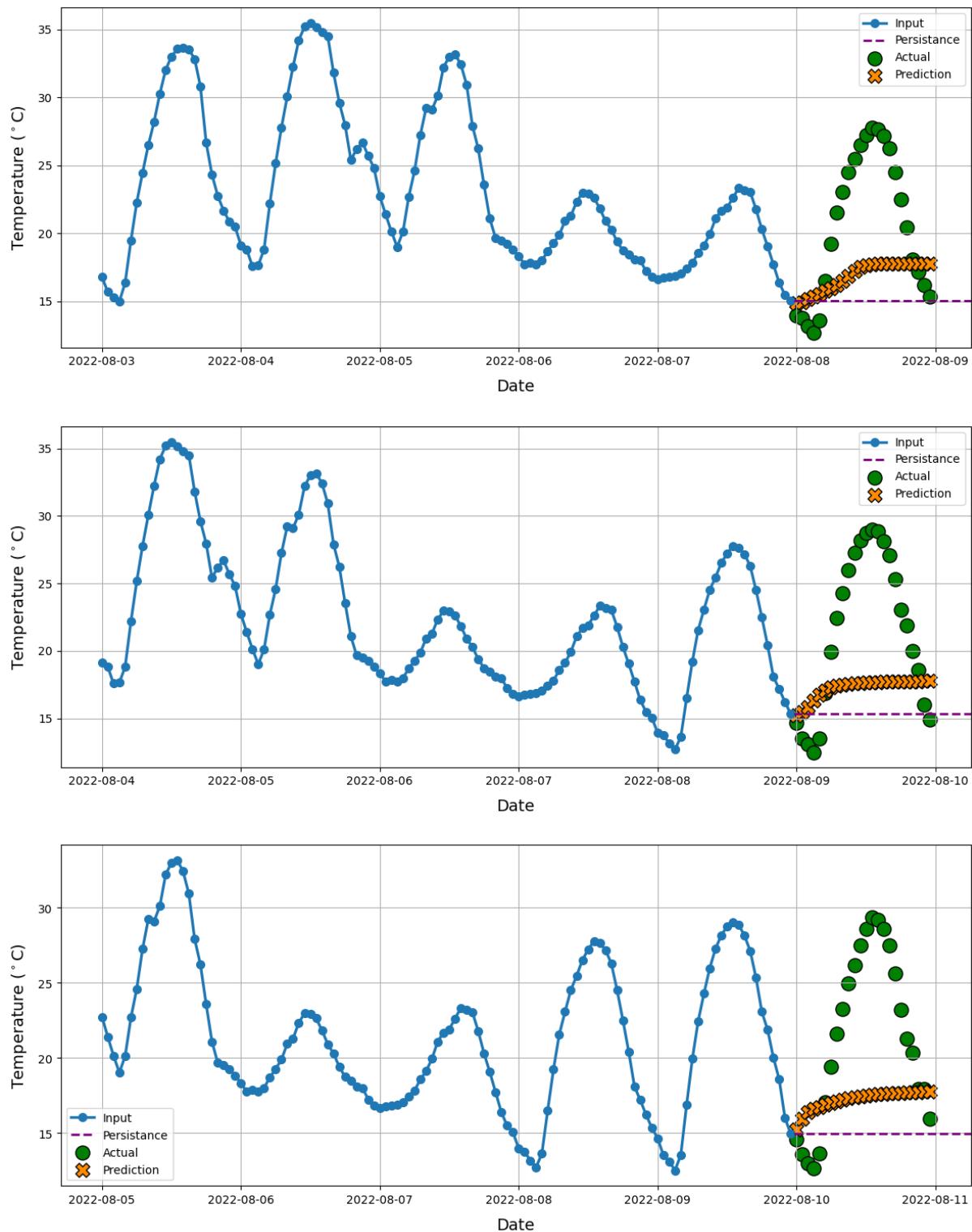


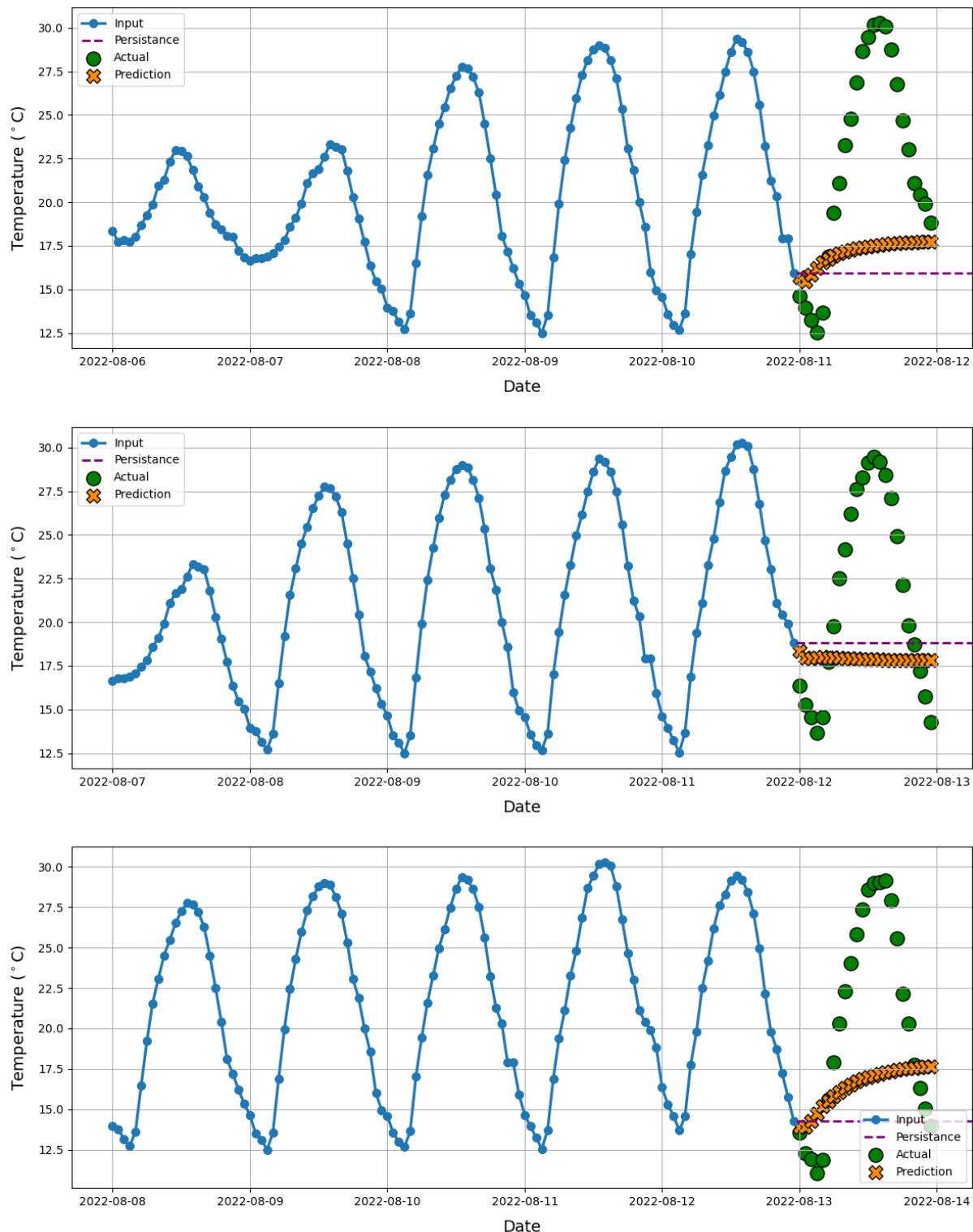
8.1.2 RNN

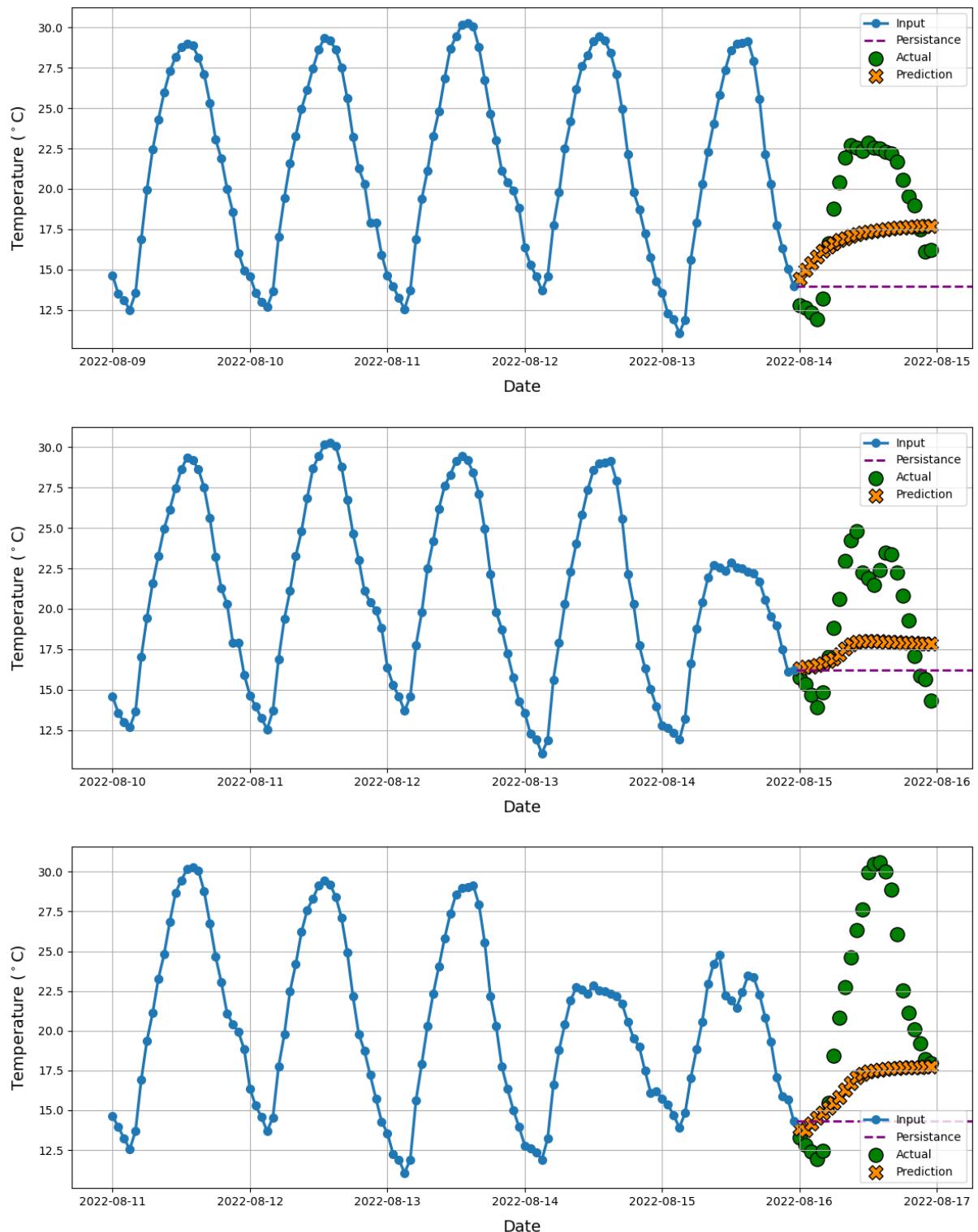


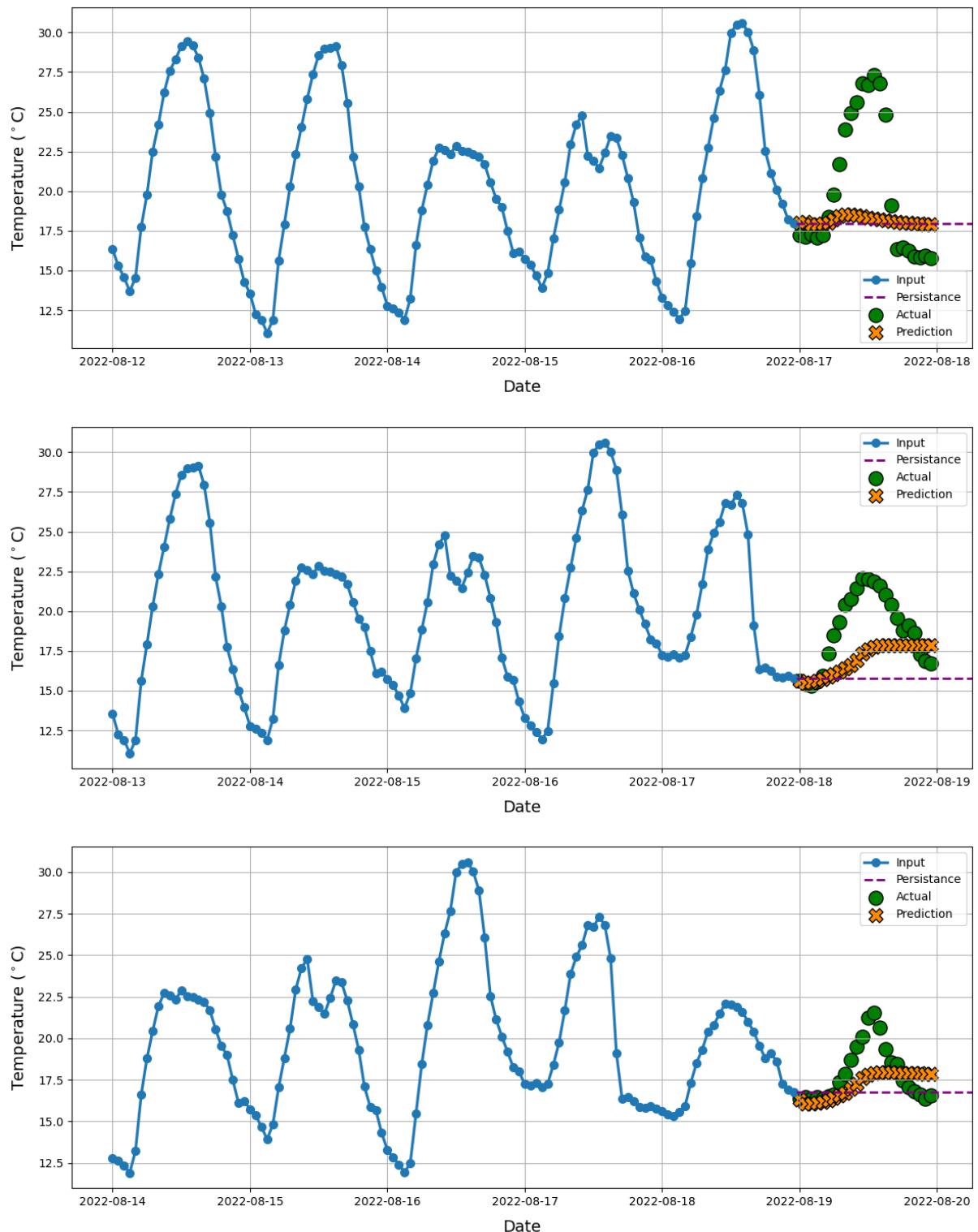


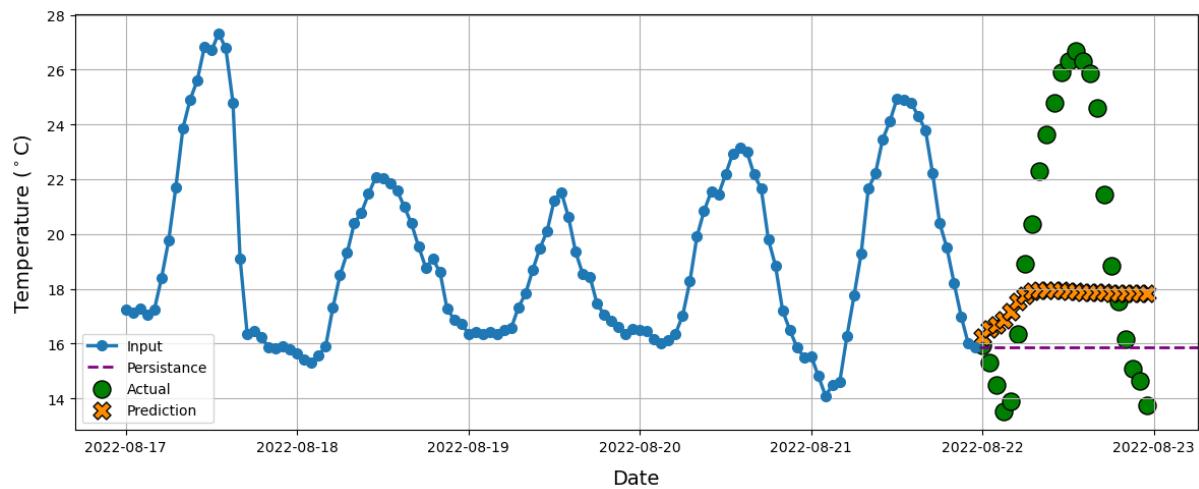
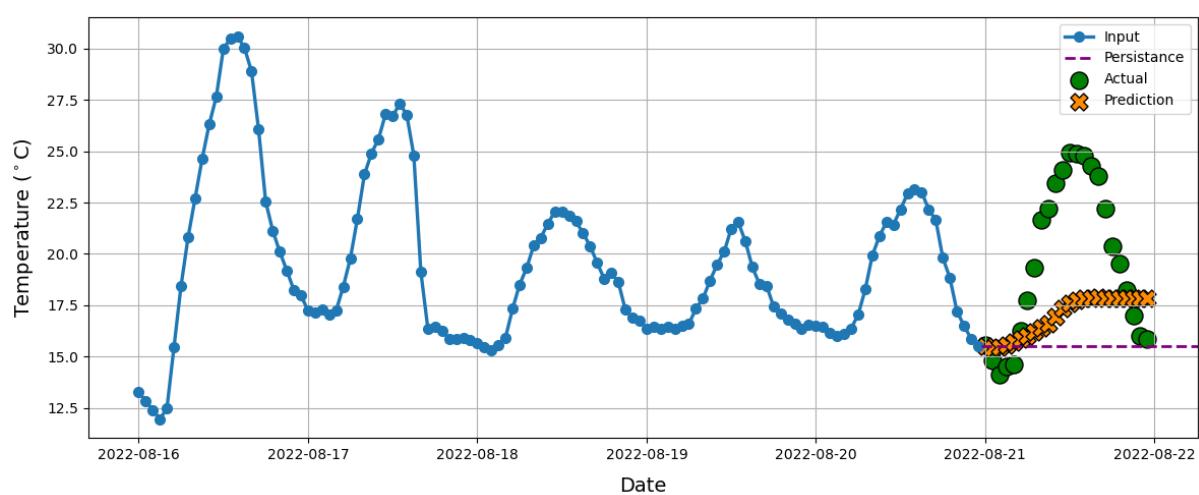
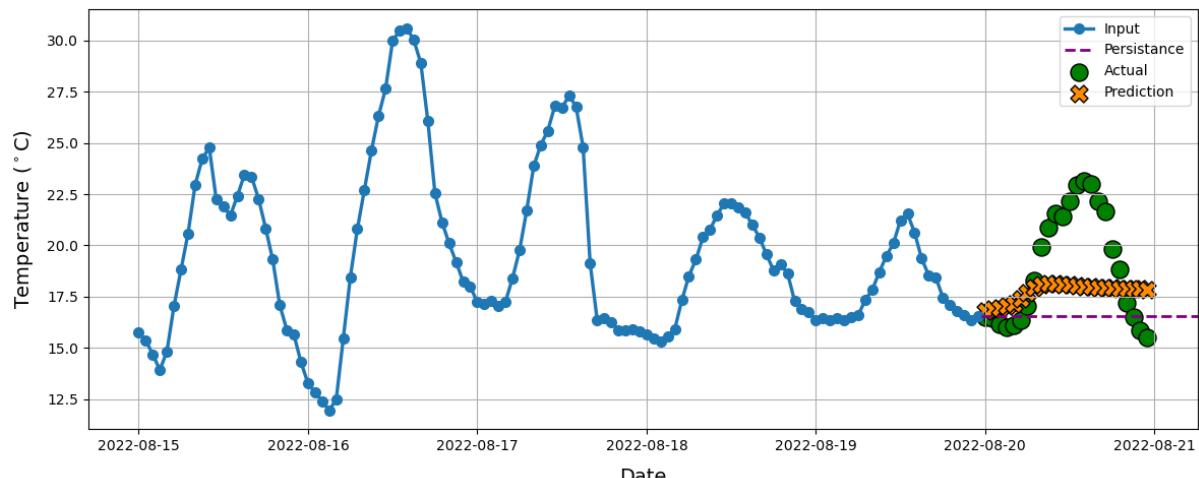


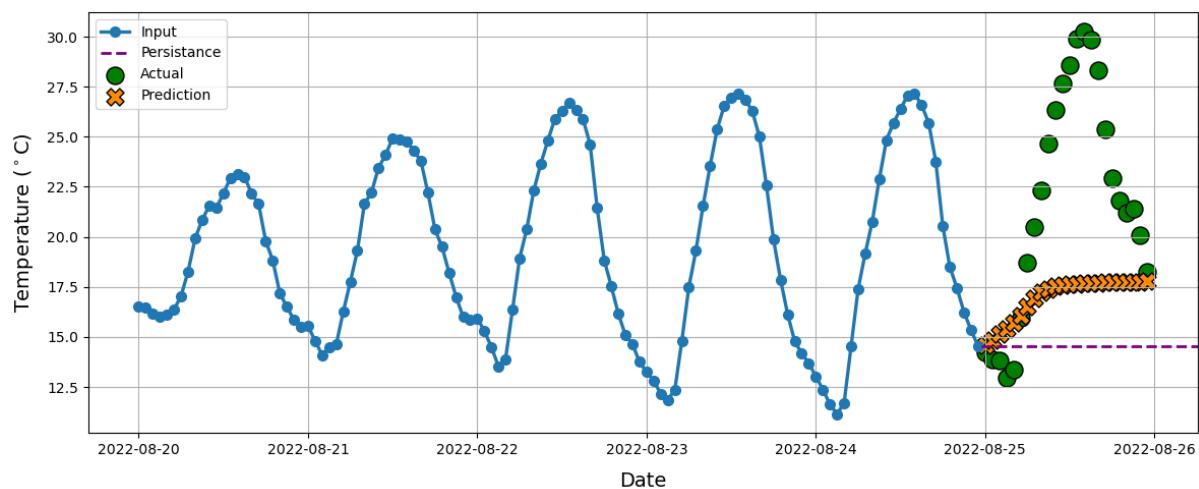
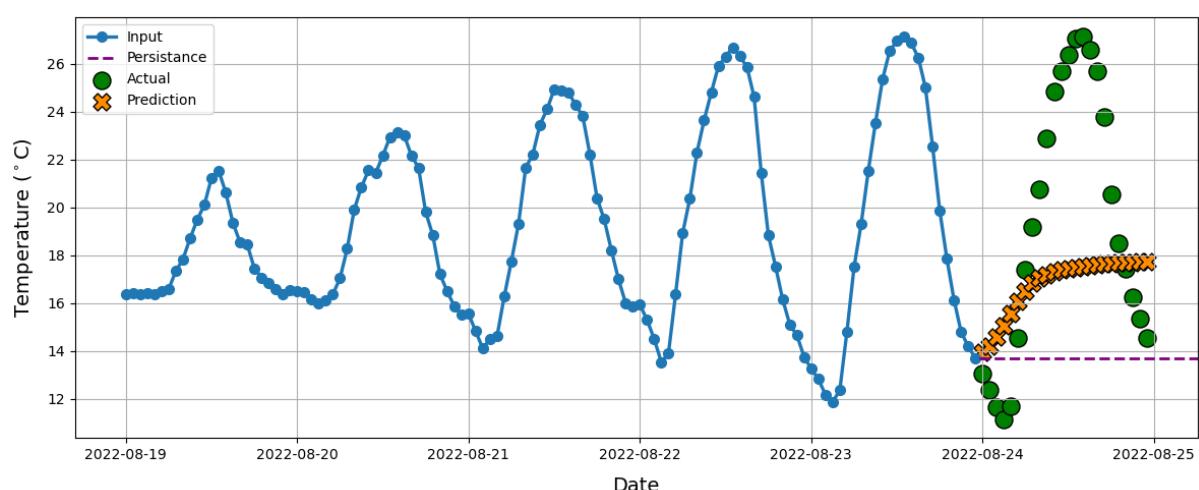
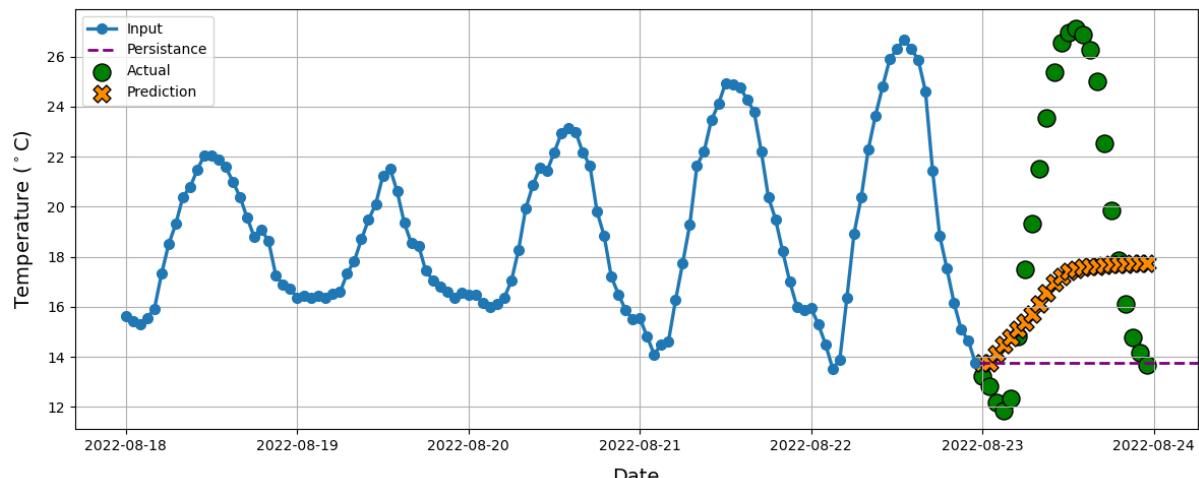


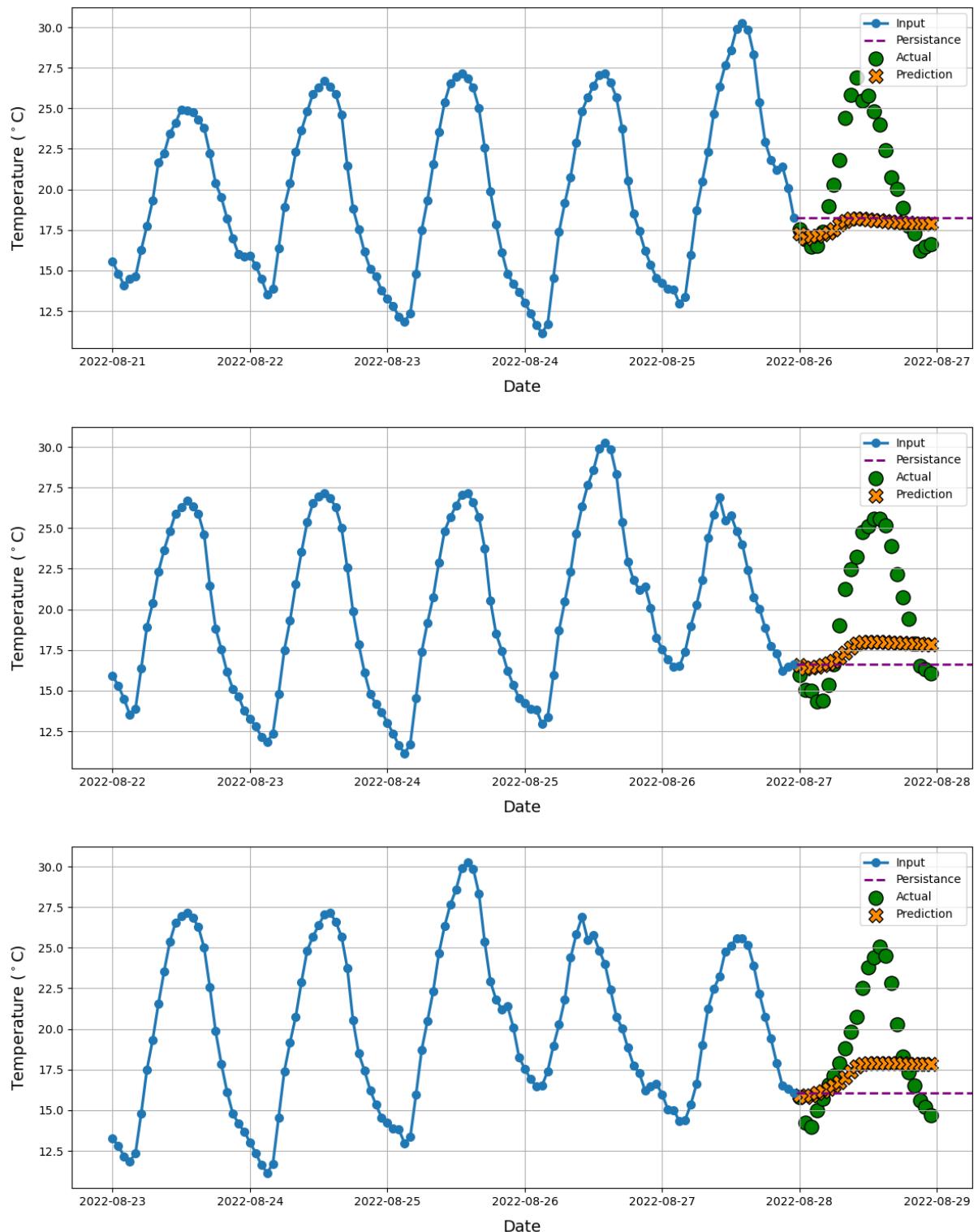


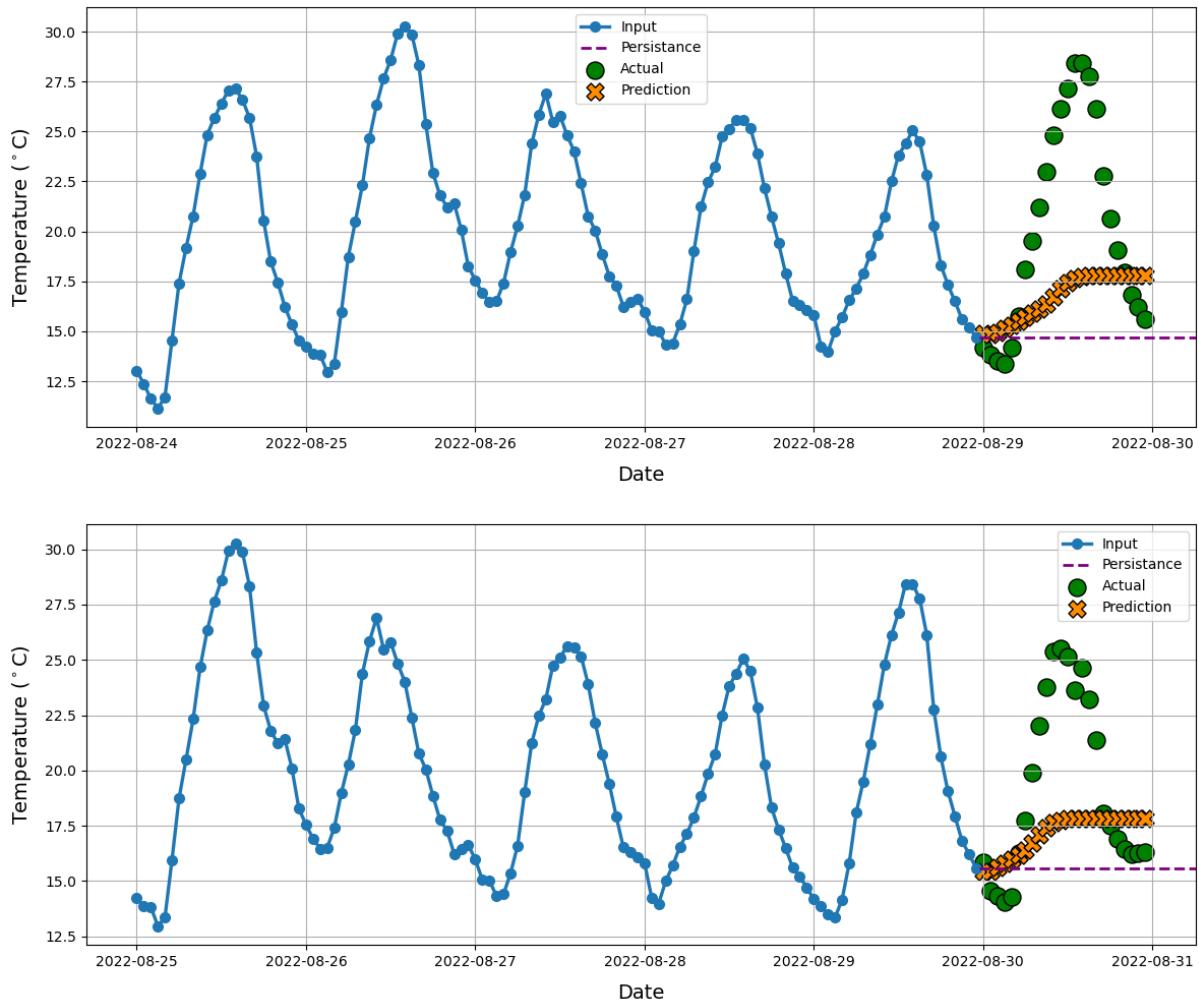








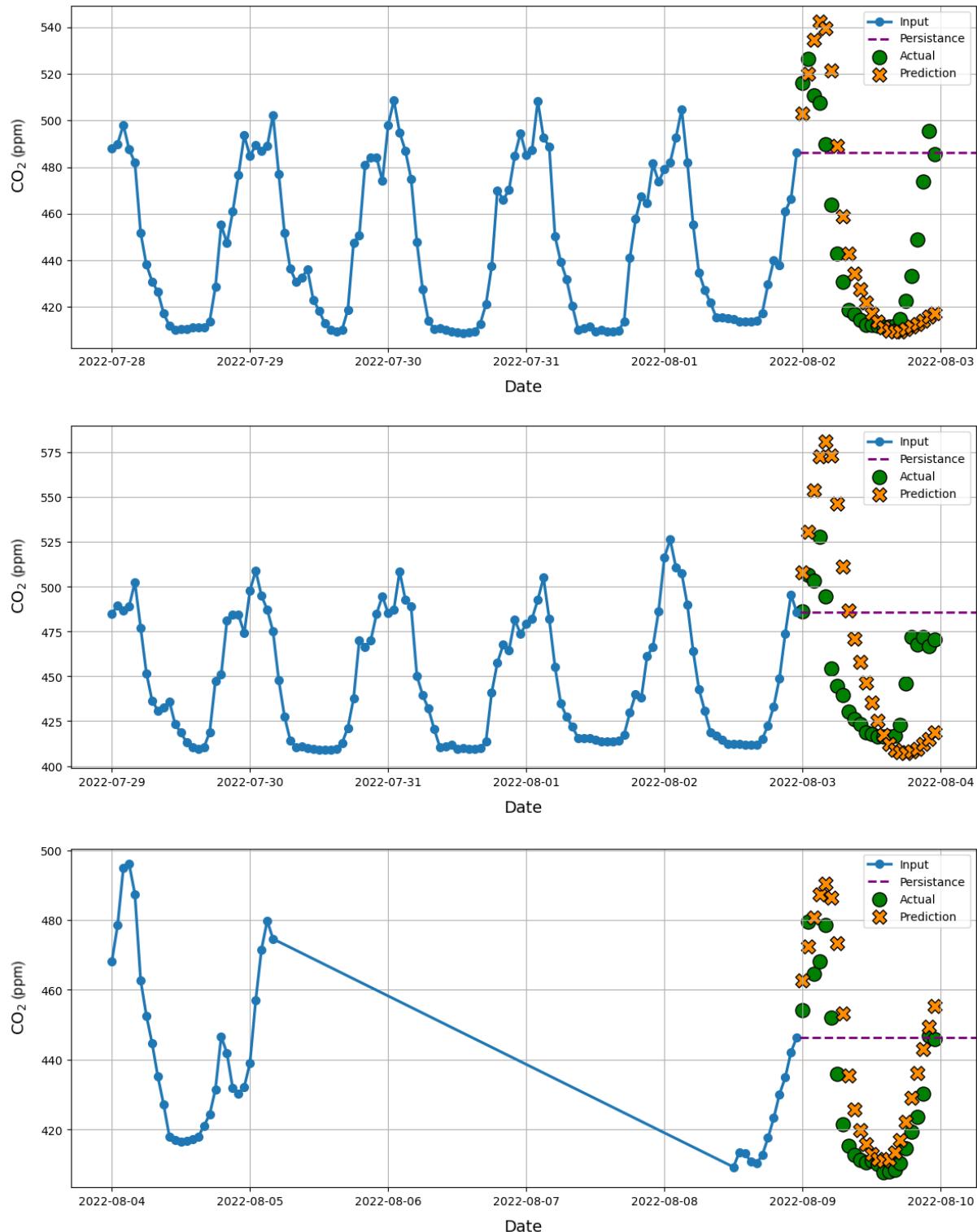


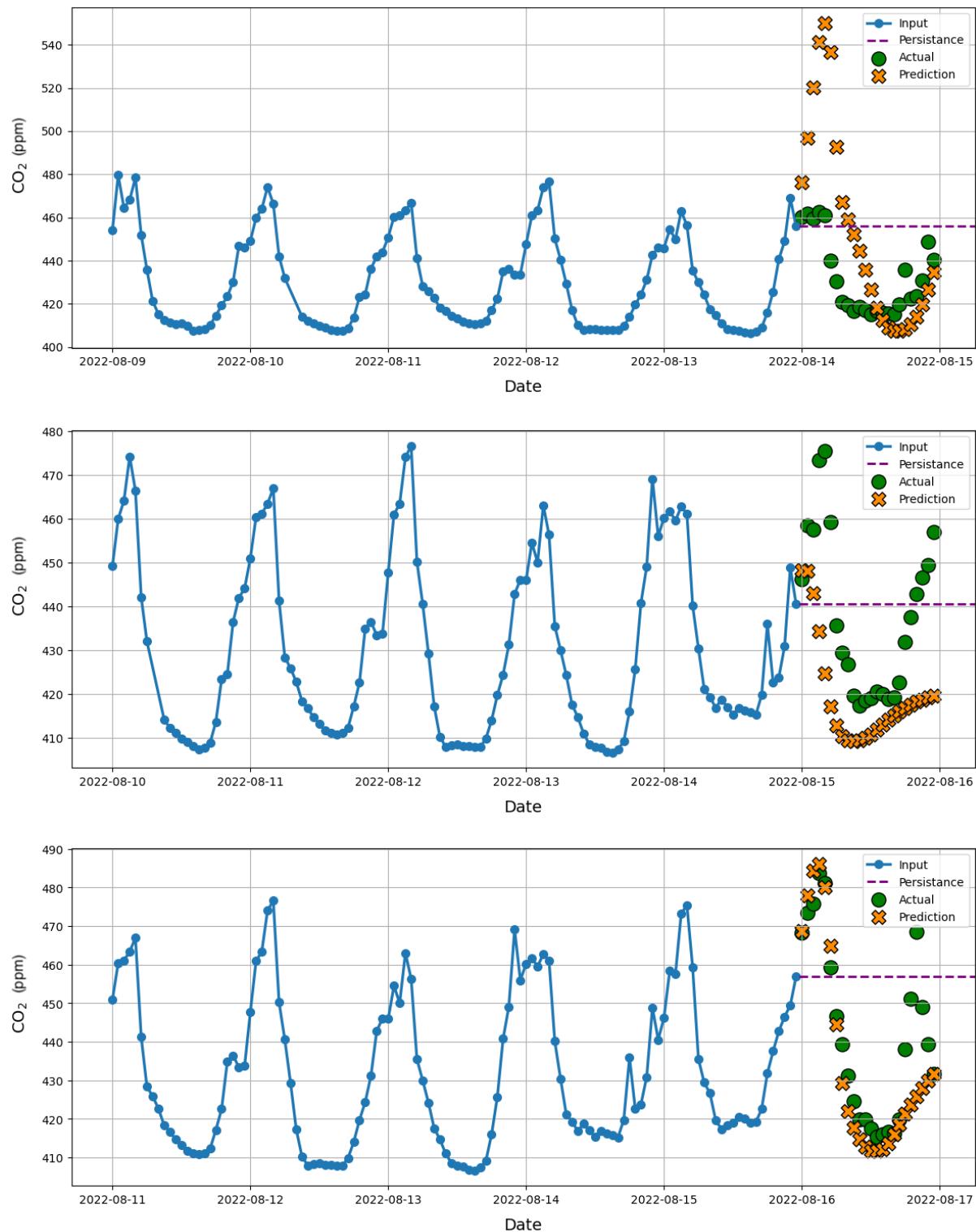


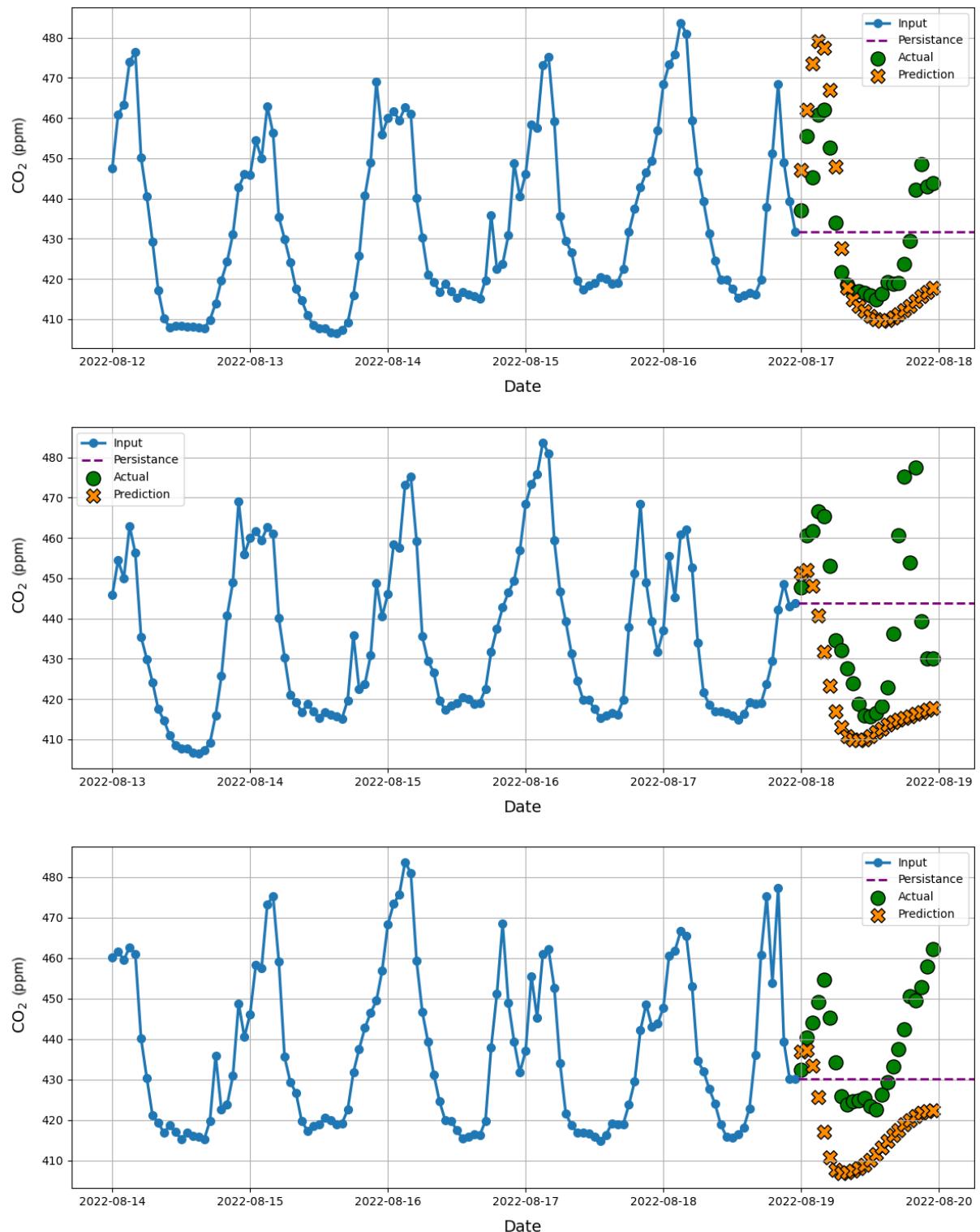
8.2 Forecast Results of CO₂

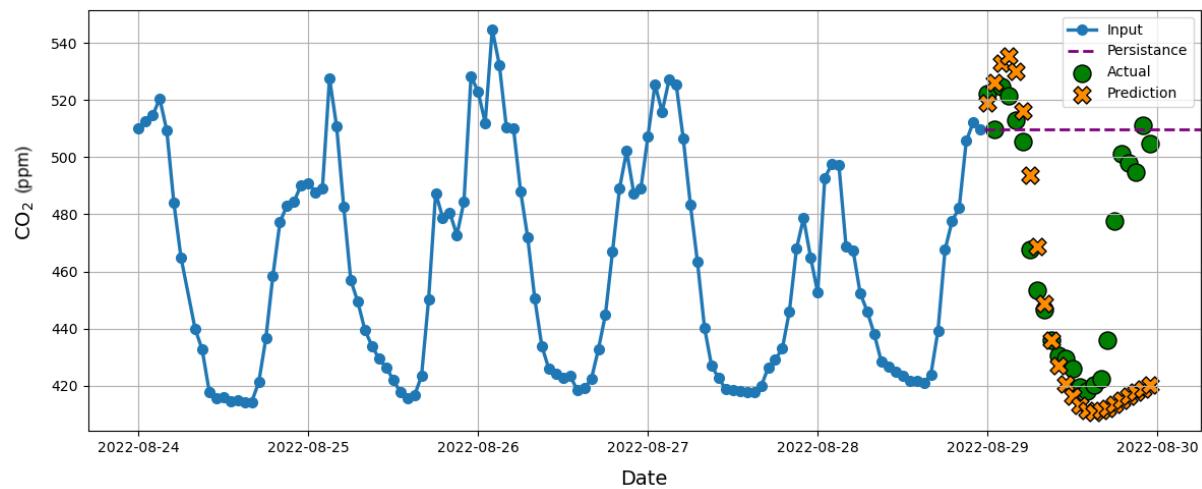
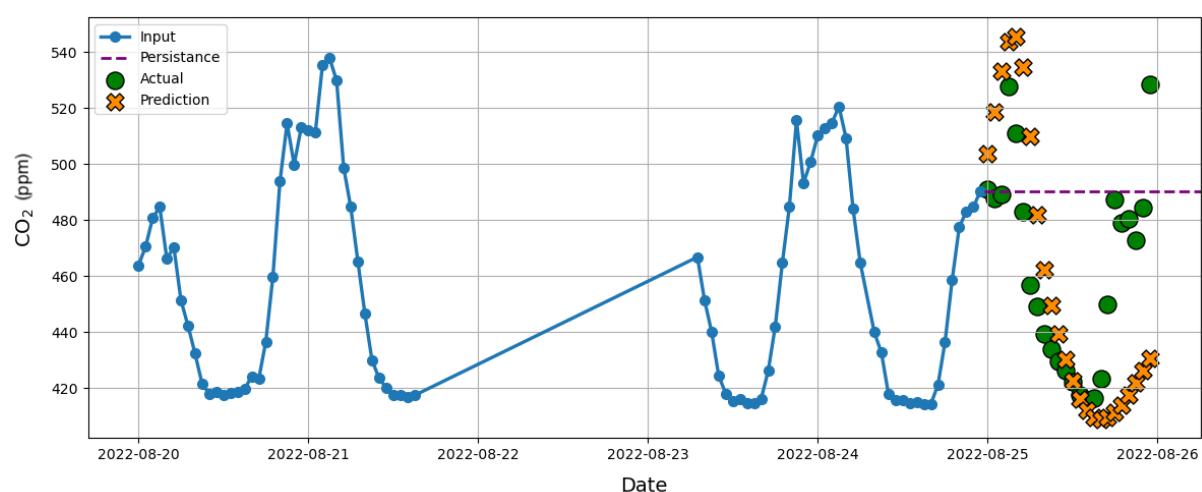
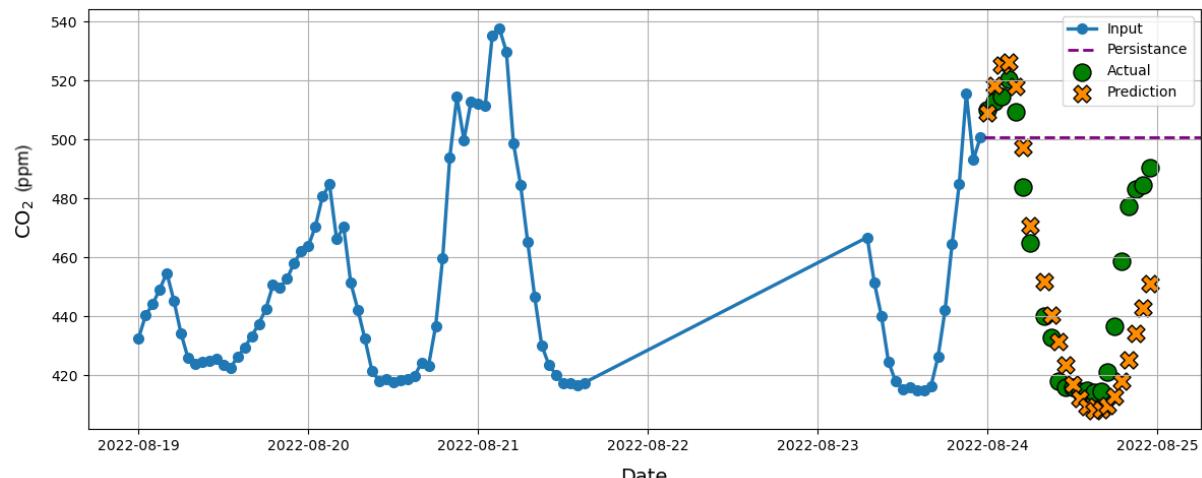
The LSTM and RNN models were also used to forecast CO₂ levels for the entire month of August. Both models, which were configured by the parameters in the Table 12, used the last five days of CO₂ data to predict the current day's CO₂ levels. However, there were some missing days in the CO₂ dataset, resulting in gaps in the forecast results.

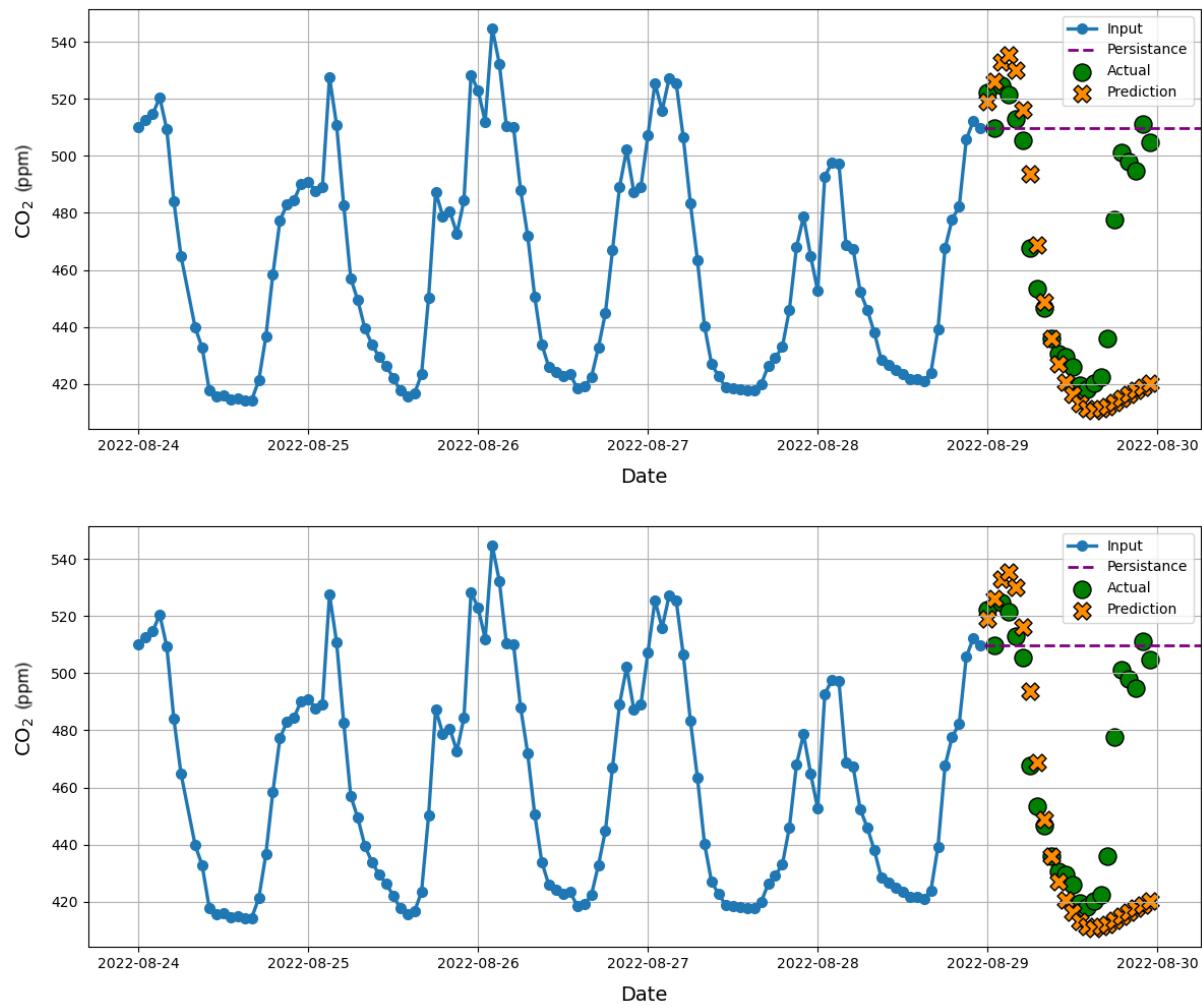
8.2.1 LSTM



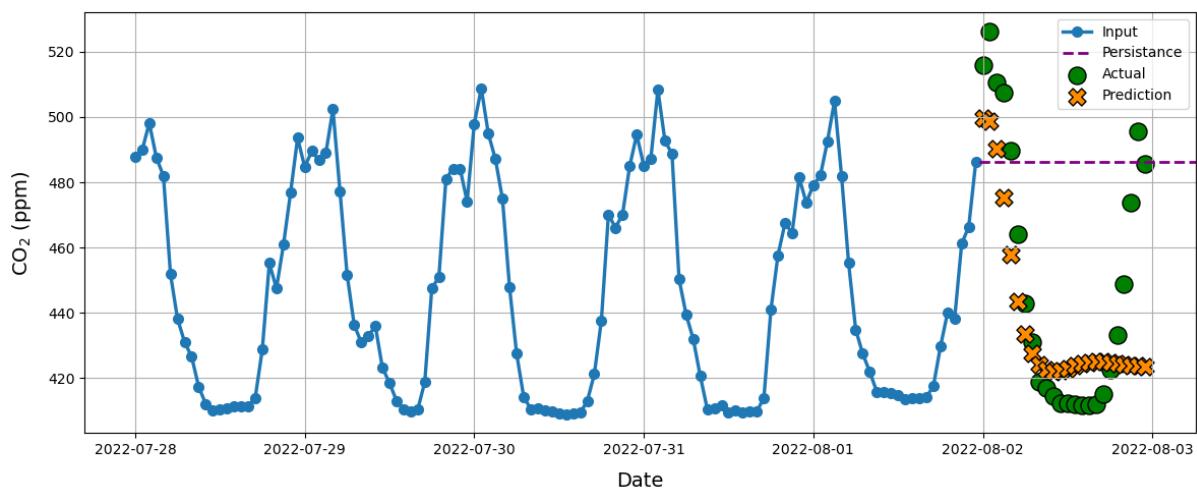


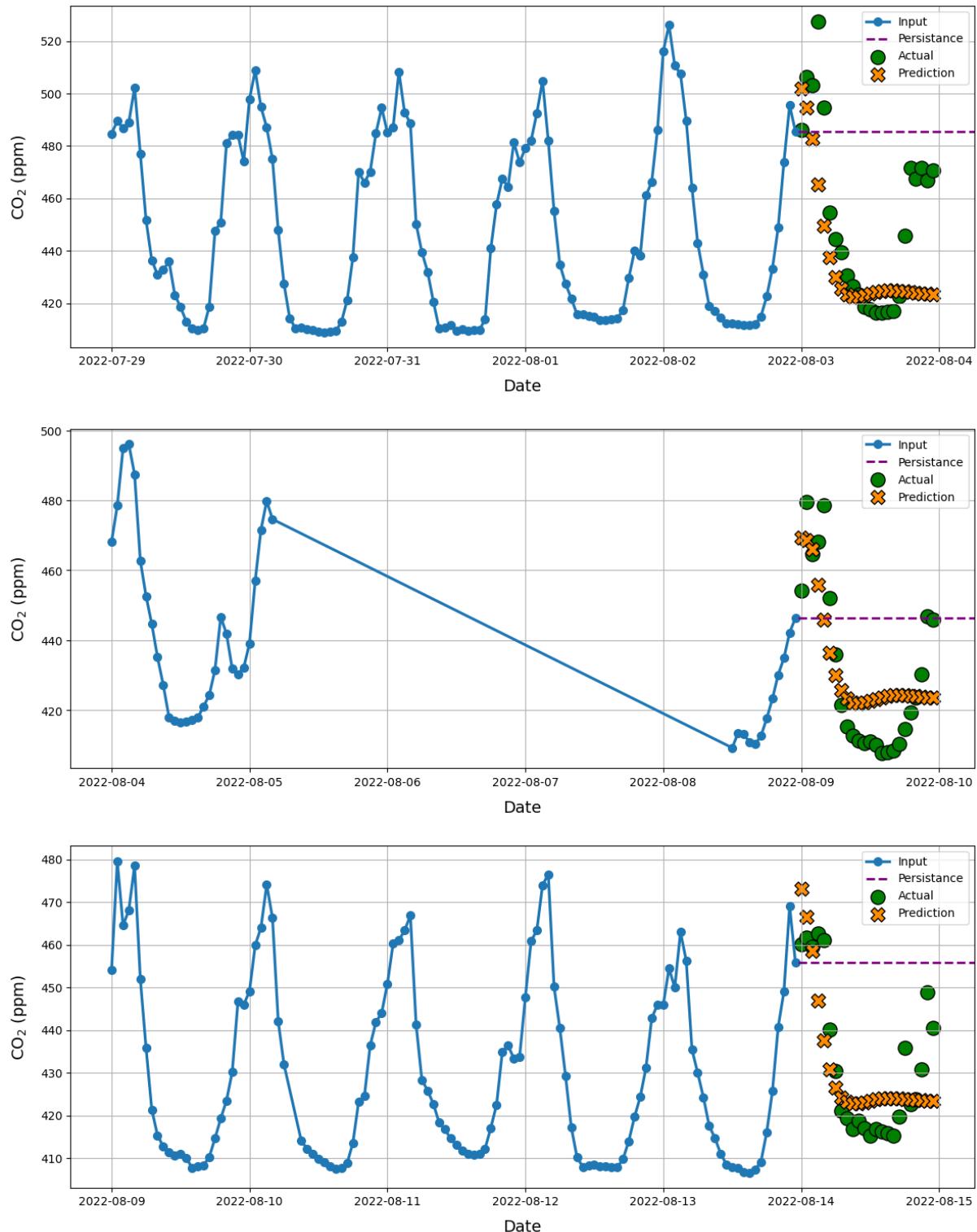


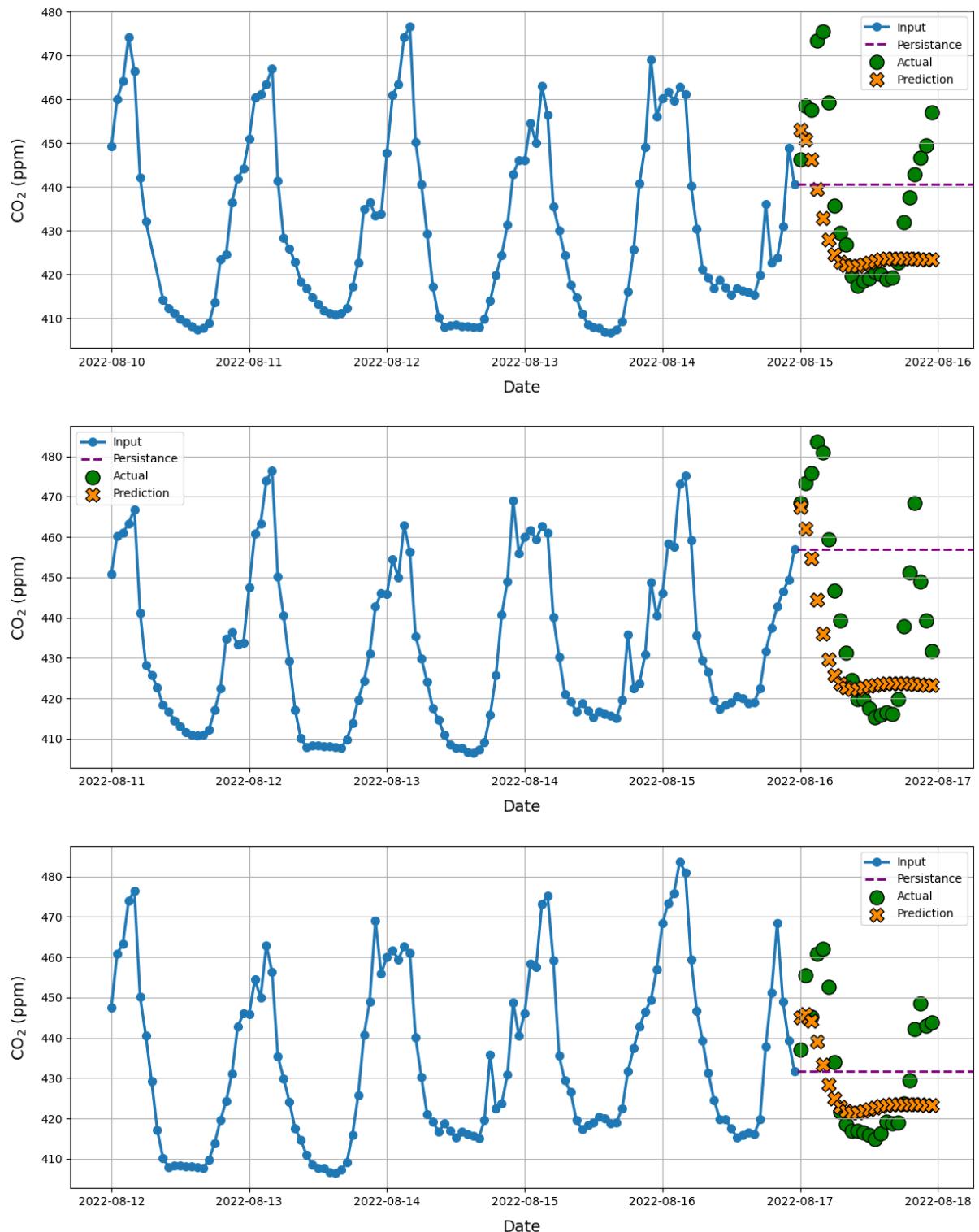


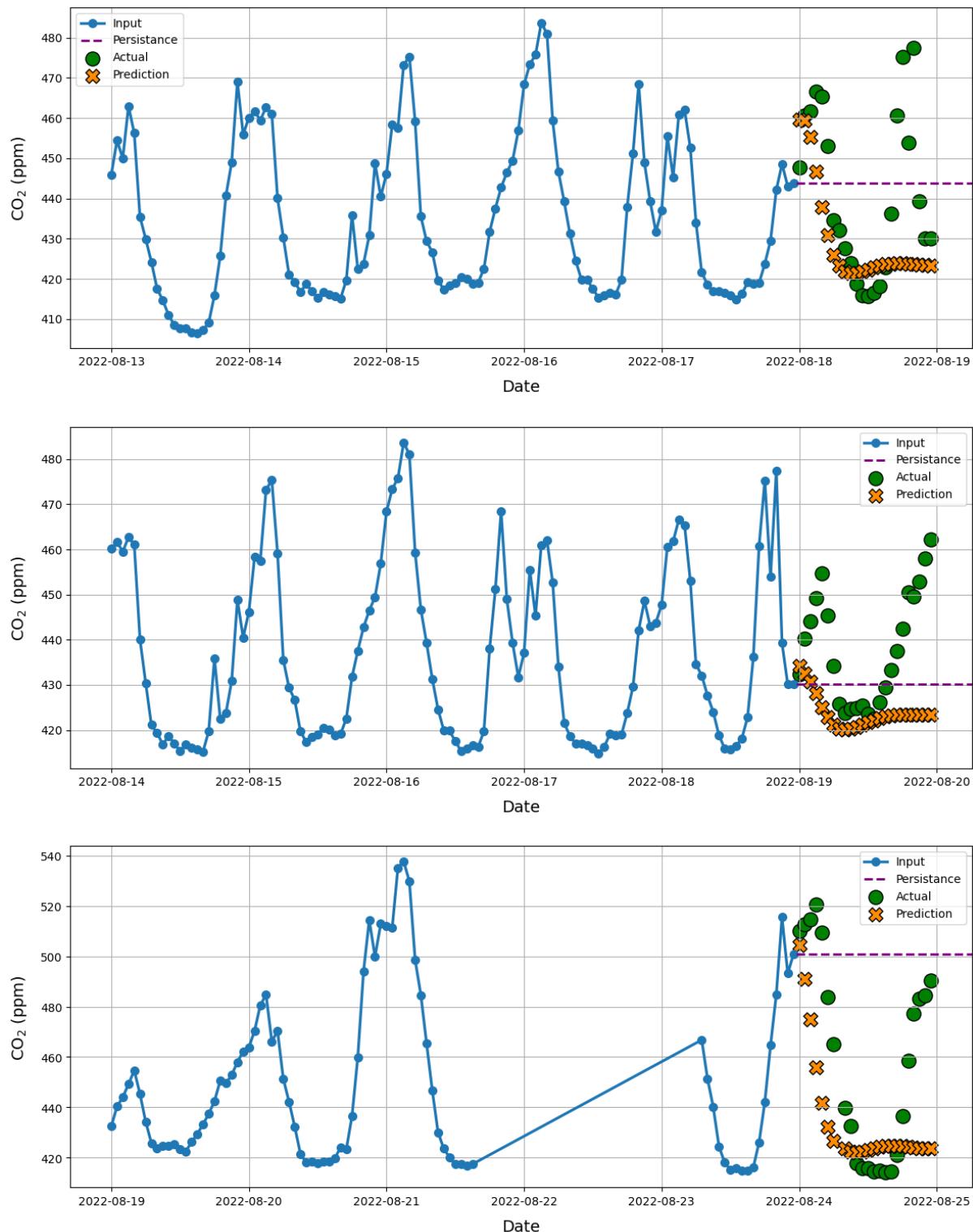


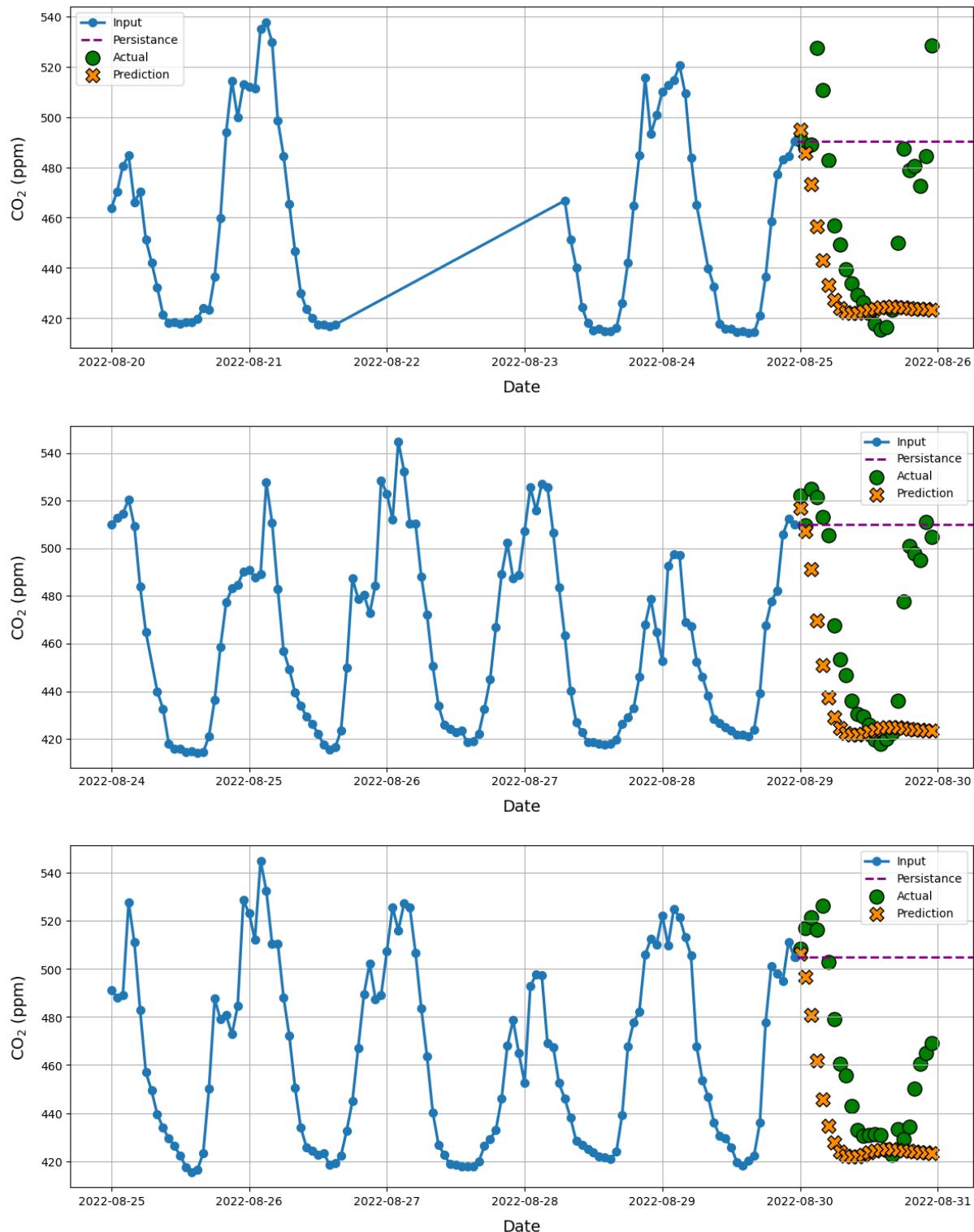
8.2.2 RNN

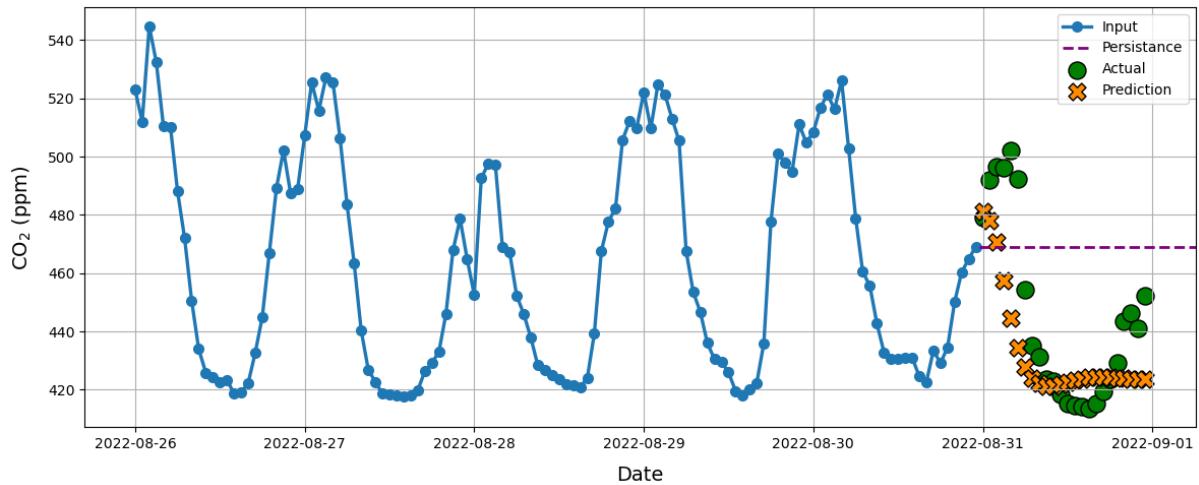












8.3 Tools and Literature

Tools

- ChatGPT version 3.5, OpenAI: openai.com/chat.
 - “Could you give me information about Payerne?”, answer to the author, 24 January 2024.
 - Help with editing of Declaration
 - Help with editing of Acknowledgement
 - “What does EDA stand for?”, answer to the author, 28 February 2024.
 - “What is model sensitivity in deep learning?”, answer to the author, 5 May 2024.
 - “What is scaling and scalers in deep learning?”, answer to the author, 10 May 2024.
 - “Why is activation function important for LSTM and RNN models?”, answer to the author, 13 May 2024.
 - Help with enhancing English language quality for the thesis