Temperature Prediction with Recurrent Neural Networks
FYS5429 Project 1

Andrea Myrvang
https://github.com/AMyrvang/FYS5429_Project1
(Dated: March 31, 2024)

In this project, we delve into the predictive capabilities of neural networks concerning global temperature changes, a critical aspect of ongoing climate research. Our focus is on three types of deep learning models: Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRU), each known for its ability to process sequential data effectively. Utilizing a comprehensive dataset spanning from 1961 to 2019, which includes temperature readings from a global network of meteorological stations, we embark on a comparative analysis of these models' performance. Our findings reveal a nuanced landscape of model efficacy, with RNNs displaying unexpectedly superior accuracy in predicting temperature changes, as evidenced by lower Root Mean Squared Error (RMSE) values—RNNs achieved an RMSE of 0.291, compared to LSTMs at 0.303 and GRUs at 0.308. These results challenge the prevailing notion that models equipped with sophisticated mechanisms for handling long-term dependencies inherently provide better forecasts for complex time series data like climate variables. This study not only underscores the importance of selecting the appropriate model based on the specific characteristics of the dataset but also opens up new avenues for research into improving the predictive accuracy of neural networks in the context of climate change. The good performance of RNNs in this setting suggests that when it comes to temperature predictions, a nuanced approach that carefully considers the data's temporal dynamics can yield significant benefits, thereby enhancing our ability to forecast and respond to global warming trends.

I. INTRODUCTION

In recent years, the global climate has undergone unique changes, with 2023 marking a milestone as the hottest year on record since they began in 1850 [12]. The escalating concentration of atmospheric greenhouse gases stands out as a primary driver behind this rapid warming trend, triggering a cascade of climatic shifts and accelerating the pace of global warming. The urgency of addressing climate change is now more apparent than ever, necessitating comprehensive strategies aimed at both understanding and mitigating its effects. Temperature, as a fundamental indicator of the Earth's climate dynamics, holds immense significance for ecosystems, biodiversity, sea levels, and the overall well-being of the human population [9]. Thus, precise forecasting of temperature variations has emerged as a cornerstone of climate science, guiding policy development and informing adaptive strategies across various sectors [9].

This report delves into a comparative analysis of three deep learning models — Simple Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRU) — for the critical task of predicting temperature changes. RNNs, with their inherent sequential data processing capabilities, are well-suited for analyzing time series data prevalent in climate studies. However, their efficacy is often hampered by issues like the vanishing gradient problem, which impedes their ability to model long-term dependencies [7]. LSTMs and GRUs, as evolved versions of RNNs, are designed to overcome these limitations through specialized memory mechanisms that enable them to retain information across longer sequences. This capability is particularly beneficial for capturing persistent trends in climate data, offering a promising avenue for enhancing the accuracy of temperature forecasts [8].

The report starts of with an overview of the theoretical framework and methodological approaches utilized. Then the results are presented leading on to the discussion, before it ends with the conclusion.

II. THEORY

A. Data set

The data used in this project where collected from https://www.kaggle.com/datasets/sevgisarac/temperature-change. It consists of temperature data for several areas from 1961 to 2019. The dataset includes the following variables:
Area: representing the geographic location.
Months: indicating the specific month of the data entry.
Element: details the type of measurement taken (such as mean temperature, minimum temperature, etc.).
Unit: specifying the unit of temperature measurement.
Value: reflecting the temperature reading.
Flag: denoting the data source or reliability.

The dataset presents temperature readings in degrees Celsius, collected from a global network of meteorological stations. It is part of the Global Surface Temperature Change dataset, provided by GISTEMP and managed by NASA's Goddard Institute for Space Studies. The data is presented in both monthly, seasonal and annual temperature anomalies, where the temperature change represents the change in average temperature compared to a baseline period from 1951-1980 [11]. This baseline is selected to provide a pre-industrial climate reference, which is critical for understanding the magnitude and impact of current climatic deviations. The following figure show how the average temperature has changed for the world since the baseline period.

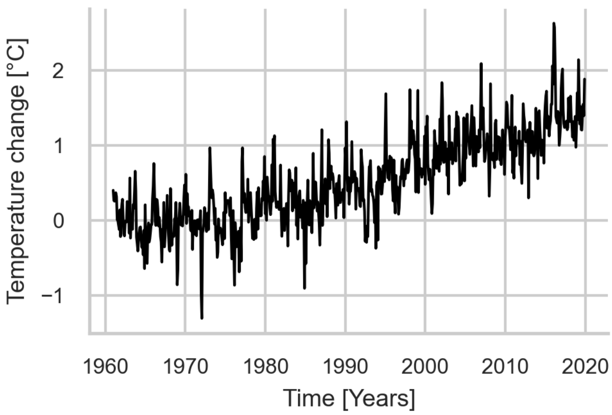

Figure 1. Annual global temperature change over the period from 1961 to 2019

B. Recurrent neural network

Recurrent Neural Networks (RNNs) are a class of neural networks designed to handle sequential data. They are particularly well-suited for analyzing time series data, natural language, and any other dataset in which the order of data points is significant. Unlike regular neural networks, which process inputs independently, RNNs consists of backward-pointing connections which enable them to retain memory of past information (illustrated in figure 2), thus allowing them to process new events in accordance with the historical context [10]. The capacity of Recurrent Neural Networks (RNNs) to hold extensive information from previous inputs is enabled by their distributed hidden state. Additionally, their ability to modify this hidden state through non-linear dynamics lends RNNs remarkable power. Given sufficient neurons and time, an RNN has the potential to process any computation [7].

The core of an RNN's structure is the hidden layer that maintains a state or memory that captures information about previously processed data. In its simplest form, an RNN works by taking an input x_t and a previous state $h_{(t-1)}$ to produce an output o_t and a new state h_t, which is fed back into the network. This process is repeated for each element in the input sequence, allowing the network to build up a context from the data it has previously seen [6].

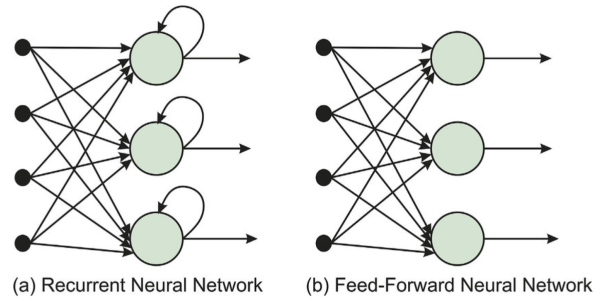

Figure 2. A comparison between Recurrent Neural Network (RNN) and Feed-Forward Neural Network (FNN) retrieved from ResearchGate [1]

Despite their capabilities in sequential data handling, RNNs confront several substantial challenges. Predominantly, they are prone to the vanishing and exploding gradient problems. The vanishing gradient issue arises when the gradients, essential for network training through back propagation, diminish exponentially as they are propagated backward in time. This significantly impedes the network's ability to learn long-term dependencies, as information from initial inputs gradually fades away. Conversely, the exploding gradient problem involves gradients increasing exponentially, potentially leading to divergent training processes[10]. Illustration of the issues encountered in computing the gradients of the loss function is given in figure 3. Which shows vanishing gradients with $|w_{hh}| < 1$, exploding gradients with $|w_{hh}| > 1$, and the desirable state where $|w_{hh}| = 1$. Solutions to the gradient challenges have emerged in the form of advanced RNN architectures, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU).

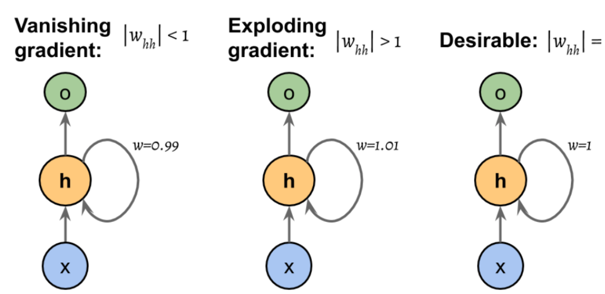

Figure 3. Problems in Computing Gradients of the Loss Function, Adapted from "Machine Learning with PyTorch and Scikit-Learn" by Raschka et al., 2022, p. 510.

C. Long short-term memory neural network

Long Short-Term Memory networks (LSTMs) are a further developed of recurrent neural networks (RNNs), designed to overcome the limitations of traditional RNNs,

such as their struggle with learning long-term dependencies. At the middle of the LSTM architecture is what´s known as a memory cell. This memory cell essentially represents or replaces the hidden layer found in standard RNNs. A critical attribute to these cells is their capacity to keep error back propagation consistent over time, attributed to an optimally set weight parameter of 1, thereby ensuring a steady learning experience throughout longer sequences.
The flow of information in the memory cell is controlled by several units, commonly referred to as gates. These gates manage what information is stored, updated or forgotten at each timestep, ensuring that the network retains only relevant information throughout the learning process. The intricate workings of an LSTM cell involve three primary types of gates, each with specific roles and associated computations:

Input Gate: This gate decides how much of the new information will be added to the cell state. It performs this by weighting incoming data, which influences the update magnitude of the cell state. The Input gate are computed as follows:

$$i_t = \sigma(W_{xi}x^{(t)} + W_{hi}h^{(t-1)} + b_i) \tag{1}$$

Output Gate: Determines the next hidden state, which is a filtered version of the cell state, dictating the output at the current timestep based on the information deemed important. Computed as follows:

$$o_t = \sigma(W_{xo}x^{(t)} + W_{ho}h^{(t-1)} + b_o) \tag{2}$$

Forget Gate: It plays a crucial role in the memory management of the cell by determining the extent to which previous information is retained or discarded. This gate allows the LSTM to avoid the issue of indefinitely growing cell state values by selectively letting through only pertinent past information. Computed as follows:

$$f_t = \sigma(W_{xf}x^{(t)} + W_{hf}h^{(t-1)} + b_f) \tag{3}$$

The hidden units at the at current timestep is computed as:

$$h^{(t)} = o_t \odot \tanh\left(c^{(t)}\right) \tag{4}$$

where $C^{(t)}$ is the cell state at time t, calculated like:

$$C^{(t)} = (C^{(t-1)} \odot f_t) \oplus (i_t \odot \tilde{C}_t) \tag{5}$$

Each gate employs its own set of weights and biases, applying a combination of sigmoid and tanh functions to regulate the flow and transformation of information within the cell.
The final output of the LSTM cell is a function of the cell state and the output gate, ensuring that the output is sensitive to the short-term inputs while retaining information about the long-term context. This dual ability to capture both short-term and long-term dependencies makes LSTMs particularly effective for a wide range of sequence modeling tasks, including temperature prediction, where understanding historical patterns is crucial for accurate forecasting [10].

D. Gated Recurrent Unit Neural Network

Gated Recurrent Unit (GRU) networks represent a streamlined variant of LSTM networks, tailored to simplify the model while preserving the ability to capture dependencies in sequence data effectively. Introduced by Cho et al. in 2014, GRUs have gained prominence for their efficiency and performance in tasks where understanding temporal dynamics is crucial [3].
The architecture of a GRU is distinguished by its consolidation of the gate mechanism into two primary gates: the update gate and the reset gate. This simplification leads to fewer parameters than LSTMs, enabling faster training without substantially sacrificing the network's capability to model long-term dependencies [3].

Update Gate: The update gate in a GRU plays a dual role akin to the combined functionality of the input and forget gates in an LSTM. It determines the degree to which the unit retains its previous state and incorporates the new input. This gate ensures the model's adaptability, allowing it to decide when to update its memory based on the relevance of new information. It is computed as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{6}$$

Reset Gate: The reset gate is pivotal for deciding how much past information to forget. It allows the GRU to discard irrelevant information, thus making the model more flexible in handling various sequences' temporal dynamics. The computation of the reset gate is as follows:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{7}$$

The GRU's final state h_t at each timestep is a blend of the previous state and the potential new state, modulated by the update gate:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{8}$$

where $\tilde{h}_t$ is the candidate activation, representing the new state information proposed by the network, computed as:

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b) \tag{9}$$

The GRU's streamlined design offers a balance between complexity and performance, making it an attractive choice for a variety of sequence learning tasks [4]. In the context of temperature prediction, GRUs provide a compelling alternative to LSTMs, capable of efficiently learning from temporal patterns and anomalies in climate data. This efficiency is particularly beneficial when computational resources are limited or when the dataset involves complex temporal relationships that require nuanced modeling.

III. RESULTS

The primary dataset consisted of temperature changes recorded at various global meteorological stations from 1961 to 2019. Firstly, the data were preprocessed, which involved selecting data from specific geographical regions, such as a country or continent. The preprocessing stage also included cleaning the dataset for missing or extra values and converting categorical data into numerical format, notably by transforming month names into their corresponding numerical representations. The final data used in the comparison was the temperature changes for the whole world to encompass the global temperature changes. To facilitate efficient convergence of neural network models during training, the temperature change values were normalized using the MinMaxScaler, scaling the data to a range between 0 and 1. Subsequently, the dataset was divided into training (81.36%) and testing (18.74%) sets, with the later representing the last 10 years of the dataset (2009 to 2019), to later evaluate the model performance on unseen data.

The models - the Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and the Gated Recurrent Unit (GRU) - were implemented using the TensorFlow and Keras libraries. They were optimized using the Adam optimizer with a learning rate of 0.001 and evaluated using Mean Squared Error (MSE) as the loss function. The models were trained over 1000 epochs, with a lookback of 12 for the LSTM and GRU models to capture the temporal dependencies in the data effectively. The training process involved fitting the models to the training dataset derived from the preprocessed data, with performance subsequently evaluated on the test dataset

The RNN model was constructed with a single recurrent layer comprising 100 units, chosen to balance model complexity and computational efficiency. This configuration was intended to capture the sequential nature of the temperature data without overburdening the training process. The LSTM model, recognized for its ability to learn long-term dependencies, was configured with 50 units in its hidden layer. This relatively lower number of units, compared to the RNN, was selected to mitigate the increased computational demand inherent to LSTMs, owing to their more complex internal mechanisms (i.e., input, output, and forget gates). Similarly, the GRU model was set up with 50 units. The GRU's architecture, which merges the functionalities of LSTM's input and forget gates into a single update gate, offers a streamlined alternative to LSTMs. This setup was anticipated to provide a balance between capturing long-term dependencies and maintaining computational efficiency.

Table I. Comparative Performance Metrics of RNN, LSTM, and GRU Models on Temperature Prediction

Model	RMSE	MAPE
RNN	0.291	20.3%
LSTM	0.303	21.2%
GRU	0.308	22.1%

The performance of each model was meticulously evaluated using two primary metrics: the Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE). These metrics were meticulously chosen for their capacity to deliver a thorough insight into the models' predictive accuracy and overall reliability. RMSE serves as a direct indicator of model accuracy by computing the square root of the average squared differences between the predicted and actual temperatures. This metric is especially critical for highlighting larger errors, thereby ensuring that models with lower RMSE values are deemed more accurate in their forecasts [5]. Conversely, MAPE quantifies the prediction error as a percentage, offering an intuitive gauge of model performance in relation to the actual temperature readings. This metric is exceptionally valuable for appraising the practicality of the models' outputs, enabling an assessment of their real-world applicability and significance[2]. The final performance metrics of the models are summarized in Table I, which presents a comparative overview of the RMSE and MAPE values for the RNN, LSTM, and GRU models.

Figure 4 illustrates the predictive performance of the RNN, LSTM, and GRU models on the test dataset. The figure compares the actual temperature changes against the predictions made by each model, offering a visual representation of how closely each model's predictions align with the real data.

To ensure a robust evaluation that accounts for the sequential nature of the dataset, a time series cross-validation approach was employed. Unlike standard cross-validation, time series cross-validation respects the temporal order of observations, making it a good fit for forecasting tasks. This method involves dividing the time-ordered dataset into a series of training and test sets, where each test set follows its corresponding training set. This setup simulates a realistic forecasting scenario, ensuring that predictions for future data are based exclusively on past information, thus preventing data leakage and maintaining the integrity of the evaluation process. The performance metrics obtained from this rigorous cross-validation process are summarized in the table II. Here the mean RMSE values with standard deviations for each model under 5 and 10 split cross-validation are presented, reflecting the variability and reliability of model performance across different time segments.

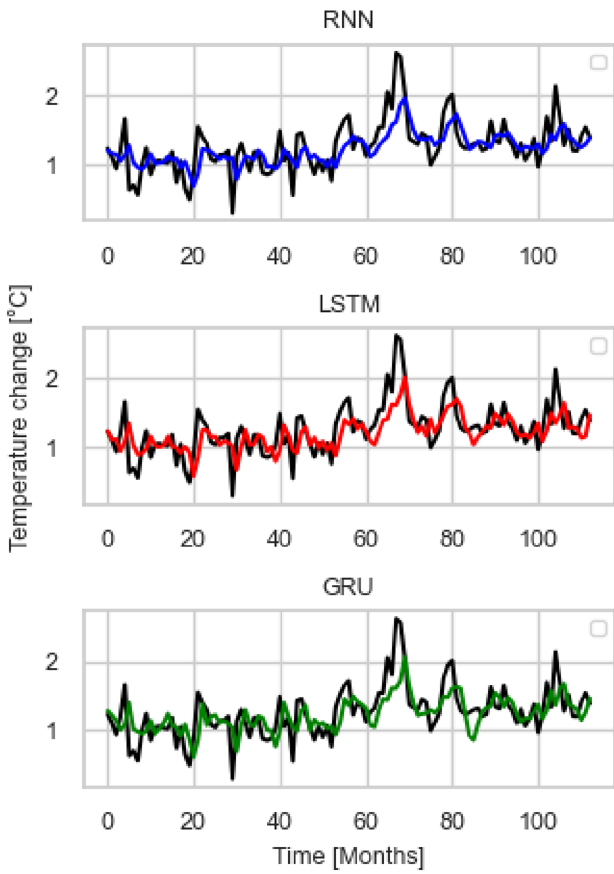

Figure 4. Comparative performance of RNN, LSTM, and GRU models on the test dataset. The graph illustrates the models' predictions (RNN in blue, LSTM in red, GRU in green) against the actual temperature changes (in black).

Table II. Cross-Validation Performance Metrics for RNN, LSTM, and GRU Models.

Model	5 splits	10 splits
RNN	0.3390 ± 0.0214	0.3276 ± 0.0429
LSTM	0.3400 ± 0.0156	0.3291 ± 0.0504
GRU	0.3453 ± 0.0143	0.3370 ± 0.0608

IV. DISCUSSION

The evaluation of Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) models in the task of temperature prediction shows differences in performances demonstrated by the Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) metrics. The RNN showcases superior performance with the lowest RMSE at 0.291, outperforming its counterparts. It also surpasses the LSTM and GRU in terms of MAPE, recording a percentage of 20.3 compared to 21.2 and 22.1, respectively. The lower these metrics, the closer a model's predictions are to actual data, enhancing its forecasting accuracy for temperature changes. The fact that the RNN outperforms both the LSTM and GRU is particularly noteworthy, given the prevailing belief that LSTMs and GRUs, with their advanced gating mechanisms, excel at capturing long-term dependencies in time-series data. The RNN's exceptional performance may stem from the specific characteristics of the temperature dataset used. The dataset might present patterns that are more effectively captured by the simpler structure of RNNs rather than the more complex LSTM and GRU models. This suggests that the temporal dependencies in this dataset may not require the sophisticated memory management features of LSTMs and GRUs.

Furthermore, there are subtle differences between the LSTM and GRU models. Although both aim to overcome the limitations of traditional RNNs, especially in handling long-term dependencies, the LSTM model slightly outperforms the GRU model in this study. With RMSE and MAPE values of 0.303 and 21.2% respectively, the LSTM's predictive accuracy was marginally higher than that of the GRU, which had scores of 0.308 in RMSE and 22.1% in MAPE. This slight distinction could be due to the inherent architectural differences between the two models. LSTMs, with their distinct forget and input gates, offer more refined control over the memory cell's state and content. This feature potentially provides a slight advantage in capturing the temperature dataset's temporal patterns, which may benefit from the LSTM's capacity to selectively maintain or discard information, compared to the GRU's streamlined gating mechanism.

The slight superiority of the LSTM over the GRU, in this case, implies that the complexity of temporal patterns in the temperature dataset might align more closely with the LSTM's architectural strengths. However, this advantage is relatively minor, indicating that both models are strong contenders for time-series forecasting tasks. The choice between them may depend on specific characteristics of the dataset or computational resource considerations.

Figure 4 shows that the RNN model's predictions, highlighted in blue, aligns closely with the real temperature trends, confirming its top performance with the lowest RMSE and MAPE scores. This suggests the RNN effectively captures temperature changes despite its simplicity. The LSTM and GRU models, in red and green, follow the temperature pattern well but show slight deviations from the actual data, explaining their somewhat higher RMSE and MAPE values. However, these differences are minor, indicating both models can track temperature trends reasonably well, but with slightly less accuracy than the RNN. The LSTM and GRU sometimes predict larger fluctuations more accurately than the RNN but also predict some nonexistent fluctuations. Meanwhile, the RNN tends to better generalize in new situations. The figure underscores the difficulty all models face in predicting sudden and significant temperature changes, which could reflect extreme weather or abrupt trend shifts. These are tough to forecast due to their complex causes. Merging these visual insights with the quantitative metrics underlines that while all models are competent in forecasting temperature changes, the RNN slightly outperforms the others in capturing the overall trend and minimizing errors. This analysis enhances our understanding of each model's interaction with climate data's temporal dynamics, blending numerical evidence with a qualitative perspective.

The thorough evaluation of the RNN, LSTM, and GRU models went beyond just looking at their basic performance measures (RMSE and MAPE) by using time series cross-validation. This method is especially important in forecasting over time because it makes sure the models' predictions for future values aren't mistakenly shaped by seeing future data while they're being trained. Table IIshows the results from cross-validation, highlighting the average RMSE values and their standard deviations for each model in both 5 and 10 split setups. Interestingly, the RNN model showed an average RMSE of 0.3390 ($\pm$0.0214) for 5 splits and 0.3276 ($\pm$0.0429) for 10 splits, pointing out its strong performance across different parts of the dataset. The LSTM and GRU models had slightly higher average RMSEs, with the LSTM recording averages of 0.3400 ($\pm$0.0156) for 5 splits and 0.3291 ($\pm$0.0504) for 10 splits, and the GRU showing averages of 0.3453 ($\pm$0.0143) for 5 splits and 0.3370 ($\pm$0.0608) for 10 splits. The small standard deviations linked to the average RMSE values for each model show a steady performance across different time segments of the dataset. This steadiness is key to confirming the models' dependability in predicting temperature changes across various conditions and time periods. The comparative results from cross-validation match up with the initial observations, where the RNN model slightly outdoes the LSTM and GRU models in terms of prediction accuracy. This repeated success backs up the RNN model's skill in handling the dataset's time patterns, even when it's tough to predict temperature shifts for times it hasn't seen. However, the minor differences in average RMSE values among the models, especially with the 10 split setup, imply that the best model to choose might depend on the specific needs of the forecast task. This includes how far ahead it forecasts and how detailed the data is. For example, while the RNN model generally performs the best, the LSTM and GRU models' strength in picking up on big changes and dealing with complex patterns might be more useful in some situations.

The evaluation of the RNN, LSTM, and GRU models, based on RMSE and MAPE values, shows good predictive accuracy, especially given the dataset's limited range of temperature variations. The RNN model stands out for closely mirroring actual temperature changes, showing it can pick up on the dataset's patterns despite its simpler design. But, despite these encouraging findings, several issues could affect how well these models predict and their accuracy. Firstly, the complexity and unpredictable nature of climate data can challenge any prediction model. Climate data are affected by many interconnected elements and there complex relationships might not be fully represented in the dataset, possibly overlooking some detailed patterns. Secondly, extreme weather events or sudden changes in climate patterns, becoming more common with global climate change, add unpredictability that the models might not catch without more detailed data or more complex methods that understand nonlinear relationships. Another important point is how the models are trained. Finding the right balance between learning general trends and not just memorizing the training data so that the models can apply what they've learned to new, unseen data. The dataset's size and what features it includes for training also greatly influence how well the models can learn and predict. While the models do well in predicting temperature changes, it's essential to remember these results come with the caution of climate data's inherent complexity and the difficulties in modeling it accurately. Looking ahead, improving the models could mean using a wider variety of data, trying more advanced neural network designs, or using ensemble methods that combine different models to get better accuracy and reliability. Additionally it could be a good idea to implement external factors and human activities that can influence temperature but might not be fully accounted for in the dataset. Including or adjusting for these factors could potentially make the models more accurate. These elements introduce additional variability and complexity into climate modeling, highlighting the need for comprehensive approaches that can integrate a broad spectrum of influences to refine predictions.

V. CONCLUSION

This project has embarked on an intricate journey to predict temperature changes using neural networks, specifically exploring the effectiveness of Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRU). Our exploration has revealed that despite the complexities inherent in climate data, the RNN model displayed a surprising edge in performance, showcasing a slightly superior predictive accuracy as indicated by lower RMSE and MAPE values. This outcome challenges the common expectation that LSTMs and GRUs, with their advanced mechanisms for handling long-term dependencies, would outshine simpler RNN structures in tasks involving intricate time series data like temperature fluctuations. Notably, our study underscores the critical influence of data intricacy and model compatibility, highlighting that the simpler RNN framework can, under certain circumstances, more effectively capture the dynamics of temperature changes over the more complex LSTM and GRU models. This finding suggests that the key to enhancing predictive accuracy lies not just in the sophistication of the model but in its alignment with the specific patterns and characteristics of the dataset. The models' performances, though promising, bring to light the multifaceted challenges of accurately modeling climate phenomena. Factors such as the unpredictable nature of climate dynamics, the occurrence of extreme weather events, and the limitations posed by the dataset itself suggest that there is significant room for improvement. For instance, incorporating a broader array of features, including those indicative of solar activity, atmospheric compositions, and oceanic patterns, could potentially enhance the models' ability to predict temperature changes with greater precision. Future endeavors could explore the integration of additional data sources, the application of more complex neural network architectures, or the utilization of ensemble methods to combine the strengths of multiple models and investigate how varying the complexity of the models (e.g., by changing the number of layers or units) affects the performances. Such advancements could usher in a new era of predictive accuracy in temperature forecasting, thereby offering valuable insights for climate research and global warming mitigation strategies. To conclude, this look into neural network-driven temperature prediction underscores the nuanced challenge of climate modeling. Our findings illuminate the path forward: constant innovation and a collaborative, multidisciplinary approach are vital. As we refine our methods and merge insights from various fields, we edge closer to a better understanding of climate change and temperature predictions.

VI. ACKNOWLEDGEMENT

I would like to acknowledge the valuable inspiration provided by ChatGPT throughout the report-writing process.

REFERENCES

[1] The comparison between recurrent neural network (rnn) and feed-forward neural network (fnn). https://www.researchgate.net/figure/The-comparison-between-Recurrent-Neural-Network-RNN-and-Feed-Forward-Neural-Network_fig1_338672883. Accessed: 2024-03-26.

[2] Zach Bobbitt. How to interpret mape values. https://www.statology.org/how-to-interpret-mape/, 2021. Accessed: 2024-03-26.

[3] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[5] Deepchecks Community Blog. The role of root mean square in data accuracy. https://deepchecks.com/the-role-of-root-mean-square-in-data-accuracy/, 2024. Accessed: 2024-03-26.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Accessed: 2024-03-25.

[7] Morten Hjorth-Jensen. Week 45, Recurrent Neural Networks. https://compphysics.github.io/MachineLearning/doc/LectureNotes/_build/html/week45.html#recurrent-neural-networks-rnns-overarching-view, 2023. Accessed: 2024-03-15.

[8] Eric Muccino. Lstm vs gru: Experimental comparison. *Medium*, 2019. Accessed: 2024-03-15.

[9] National Geographic Society. Earth's changing climate. https://education.nationalgeographic.org/resource/earths-changing-climate/, October 2023. Accessed: 2024-03-15.

[10] Sebastian Raschka, Vahid Mirjalili, and Yuxi (Hayden) Liu. *Machine Learning with PyTorch and Scikit-Learn*. Packt Publishing, Birmingham, 2022. page: s502-519.

[11] Sevgi Sarac. Temperature change dataset. https://www.kaggle.com/datasets/sevgisarac/temperature-change/data, 2023. Accessed: 2024-03-08.

[12] World Meteorological Organization. Wmo confirms that 2023 smashes global temperature record. January 2024. Accessed: 2024-03-15.