Drought forecasting: modern approaches

Mikhail Kuznetsov [1] **Victor Kozhevnikov** [1] **Artem Gorbarenko** [1] **Ivan Gurev** [1]

Abstract

Drought forecasting is a crucial aspect of climate prediction and water resource management. It involves predicting the severity of droughts in regions ahead of time often spanning several months. This study focuses on predicting the Palmer Drought Severity Index (PDSI) and explores three distinct approaches: regression, binary classification (threshold-based), and 3-class classification. The primary objective is to assess the efficacy of various models, including Linear, Nonlinear, and Deep Linear models: Earthformer and ConvLSTM in forecasting drought conditions. By leveraging a diverse set of techniques, this research contributes to advancing the field of drought forecasting and enhancing our ability to mitigate the impacts of water scarcity. We are using these statistical criteria: regression - R2, MSE, MAE, RMSE; binary classification - ROC-AUC, PR-AUC, F1 and accuracy for multiclass classification. Project repo is https://github.com/MikhailKuz/time-series-forecasting-2d

.

1. Introduction

Droughts represent a significant and recurring challenge for regions worldwide, exerting profound impacts on agriculture, ecosystems, water resources, and human societies. As the frequency and intensity of drought events continue to escalate due to climate change, the ability to forecast these events accurately becomes increasingly vital.

In this context, the Palmer Drought Severity Index (PDSI) stands as a fundamental tool in assessing and predicting drought conditions. It possesses key strengths and limitations in assessing drought conditions. On the positive side, the PDSI excels in identifying long-term drought patterns, particularly in regions with low to moderate latitudes. It also accounts for the impact of global warming through the consideration of surface air temperature and a physical water balance model, providing a holistic perspective on evolving drought conditions. Additionally, the PDSI takes into consideration antecedent conditions from the previous month, enhancing its accuracy. However, it faces challenges when it comes to cross-regional comparability, as it may not be as easily applied across diverse geographical areas as the Standardized Precipitation Index (SPI). [1]

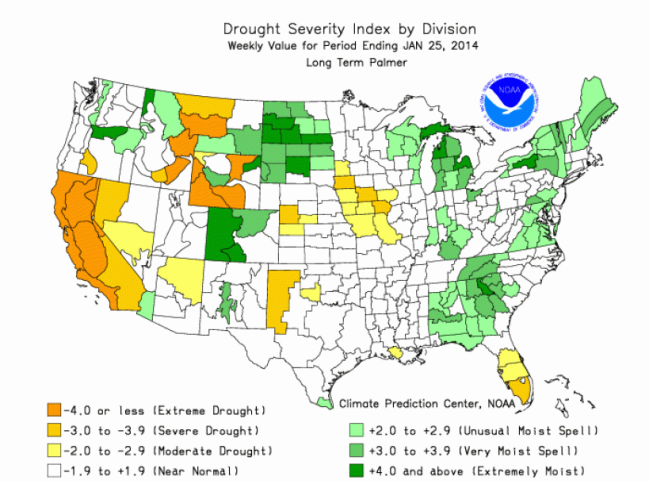

Figure 1. Long-term PDSI over the continental US through January, 2014

The performance and precision of prediction methods are influenced by a range of factors. These factors encompass decisions made regarding data pre-processing, selecting a suitable timescale, and adopting data-driven modeling approaches, whether univariate or multivariate. According to recent studies, hybrid nonlinear models emerge as the most potent tools for enhancing the precision of drought predictions, as indicated by the research findings.[2]

In this paper, we aim to assess and compare the performance of several models, ranging from conventional to modern approaches.

[1]Skoltech University, Moscow, Russia. Correspondence to: Marusov Alexander <Alexander.Marusov@skoltech.ru>.

2. Background

2.1. Linear models

Not all time series are predictable, let alone long-term forecasting (e.g., for chaotic systems). In the work [3] hypothesized that long-term forecasting is only feasible for those time series with a relatively clear trend and periodicity. As linear models can already extract such information, a set of embarrassingly simple models named LTSF-Linear was introduced as a new baseline for comparison to transformers. LTSF-Linear regresses historical time series with a one-layer linear model to forecast future time series directly. The basic formulation of LTSF-Linear directly regresses historical time series for future prediction via a weighted sum operation (as illustrated in Fig. 2). The mathematical expression is

$$\hat{X}_i = W X_i$$

where $W \in \mathbb{R}^{T \times L}$ is a linear layer along the temporal axis. $\hat{X}_i$ and X_i are the prediction and input for each i_{th} variate. Note that LTSF-Linear shares weights across different variates and does not model any spatial correlations.

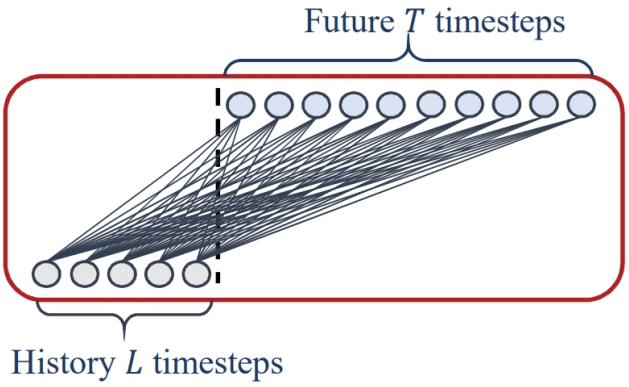

Figure 2. Illustration of the basic linear model.

LTSF-Linear is a set of linear models. Vanilla Linear is a one-layer linear model. To handle time series across different domains (e.g., finance, traffic, and energy domains), model was improved by introducing two variants with two preprocessing methods, named DLinear and NLinear.

Specifically, DLinear is a combination of a Decomposition scheme used in Autoformer [4] and FEDformer [5] with linear layers. It first decomposes a raw data input into a trend component by a moving average kernel and a remainder (seasonal) component. Then, two one-layer linear layers are applied to each component and summed up the two features to get the final prediction. By explicitly handling trend, DLinear enhances the performance of a vanilla linear model when there is a clear trend in the data.

Meanwhile, to boost the performance of LTSF-Linear when there is a distribution shift in the dataset, NLinear first subtracts the input by the last value of the sequence. Then, the input goes through a linear layer, and the subtracted part is added back before making the final prediction. The subtraction and addition in NLinear are a simple normalization for the input sequence.

As a result, the performance of LTSF-Linear surpasses the SOTA FEDformer in most cases by $20\% \sim 50\%$ improvements on the multivariate forecasting, where LTSF-Linear even does not model correlations among variates. For different time series benchmarks, NLinear and DLinear show the superiority to handle the distribution shift and trend-seasonality features.

As shown in Fig. 3, the prediction results on time series dataset with Transformer-based solutions and LTSF-Linear were plotted: Electricity (Sequence 1951, Variate 36) , where it has different temporal patterns. When the input length is 96 steps, and the output horizon is 336 steps, Transformers fail to capture the scale and bias of the future data . Moreover, they can hardly predict a proper trend on a periodic data.

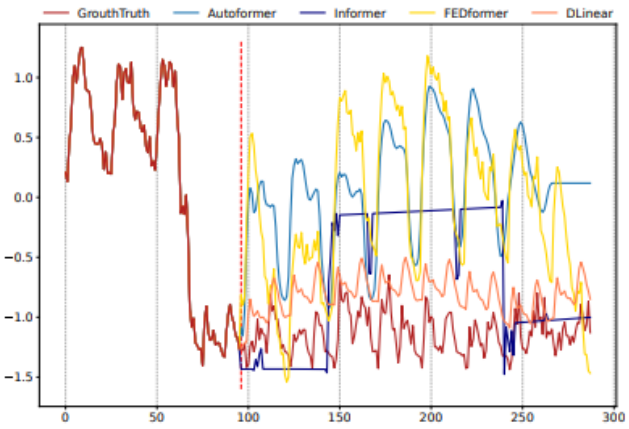

Figure 3. Illustration of the long-term forecasting outputs (Y-axis) of five models with an input length $L = 96$ and output length $T = 192$ (X-axis) on Electricity.

In general, these results reveal that existing complex Transformer-based LTSF solutions are not seemingly effective on the existing nine benchmarks while LTSF-Linear can be a powerful baseline. Another interesting observation is that even though the naive Repeat method shows worse results when predicting long-term seasonal data (e.g., Electricity), it surprisingly outperforms all Transformers on Exchange-Rate (around 45%).

2.2. ConvLSTM

ConvLSTM model - a revolutionary approach that utilize of convolutional neural networks (CNN) and long short-term memory (LSTM) networks, combining the strengths of both to create a powerful tool in earthquake prediction. [6]

CNNs, renowned for their proficiency in processing images and other two-dimensional signals, are ideal for handling the spatial dependencies between earthquakes. They capture and encode various spatial features to construct a comprehensive overview of seismic activities. On the other hand, LSTMs excel in processing time series data, thus capturing temporal dependencies between earthquakes. Equipped with an additional state cell, LSTMs can remember or forget information over extended periods, accounting for time dependencies between sequential data points. As such, the ConvLSTM model's unique strength lies in its ability to capture both spatial and temporal dependencies, crucial for accurate earthquake prediction. Information is passed through the LSTM as a feature map obtained using CNN, creating a robust framework for making precise, future predictions based on past earthquake data. In the proposed by authors pipeline, earthquake data is initially represented as heat maps, with each cell indicating the magnitude of an earthquake on any given day. These heat maps are then processed through a convolutional network to generate an embedded representation.

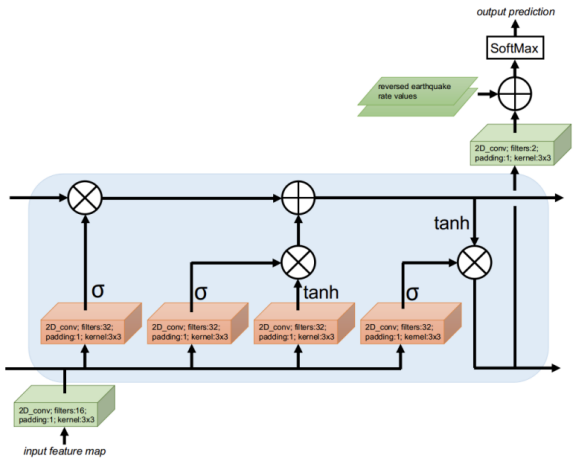

Figure 4. Proposed CNN-LSTM model. Green blocks represent series of convolutions, orange blocks represent convolution layers, circles — multiplications and additions.

Simultaneously, the LSTM component of the model generates a threefold output at each time instance, capturing temporal dependencies in seismic activity. The LSTM output is then reshaped to align with the two potential outcomes of earthquake prediction: the occurrence or non-occurrence of an earthquake. Finally, the model adjusts its predictions based on the historical likelihood of earthquakes at specified locations, enhancing its forecasting precision.

The authors put this ConvLSTM model to the test by developing a data-based classification model designed to predict if an earthquake, with a magnitude above a specified threshold, would occur in a designated 10x10 km area within 10 to 60 days. The ConvLSTM model impressively outperformed traditional baseline feature-based models, showcasing promising results. Especially for historical earthquake data from Japan, the ConvLSTM model achieved a ROC AUC of 0.975 and PR AUC of 0.0890, demonstrating its impressive accuracy and quality metrics.

In conclusion, the integration of CNN and LSTM into the ConvLSTM model signifies a significant leap forward in the realm of earthquake prediction. By effectively capturing both spatial and temporal dependencies, the ConvLSTM model is a testament to the power and potential of modern neural network architectures. It represents a new era in our approach to solving complex scientific problems, demonstrating the potential of such technologies in offering reliable and accurate earthquake predictions. However, as with any machine learning model, it's crucial to continue refining and testing these methods across diverse datasets to ensure their broad applicability and robustness.

2.3. Earthformer

In this paper, the authors propose Earthformer [7], a space-time Transformer designed specifically for Earth system forecasting. To address the challenge of applying Transformers to high-dimensional Earth observation data, they introduce Cuboid Attention as a building block for efficient space-time attention. This involves decomposing the input tensor into non-overlapping cuboids and applying cuboid-level self-attention in parallel. By stacking multiple cuboid attention layers with different hyperparameters, they are able to capture different types of correlations and explore new attention patterns. However, a limitation of this design is the lack of communication between local cuboids. To overcome this, they introduce a collection of global vectors that attend to all the local cuboids, allowing for the gathering of overall system status and information sharing between cuboids.

The data is represented as a spatiotemporal sequence $[\mathcal{X}_i]_{i=1}^{T}, [\mathcal{Y}_{T+i}]_{i=1}^{K}, \mathcal{X}_i \in \mathbb{R}^{H\times W\times C_{in}}, \mathcal{Y}_i \in \mathbb{R}^{H\times W\times C_{out}}$. Generic cuboid attention can be decomposed into *decompose*, *attend* and *merge* steps. Let's look at this closer.

The *decompose* step in the cuboid attention involves breaking down the input spatiotemporal tensor $\mathcal{X}$ into a sequence of cuboids.

$$\mathbf{x}^{(\mathbf{n})} = \text{Decompose}(\mathcal{X}, \text{cuboid\_size}, \text{strategy}, \text{shift})$$

This is done using the Decompose function. The cuboid size determines the dimensions of each local cuboid, while the strategy controls whether a local or dilated decomposition strategy is used. The shift parameter specifies the window shift offset. The number of cuboids in the sequence is determined by the dimensions of the input tensor and the cuboid size. If the dimensions are not divisible, padding is applied to the input tensor.

In *attend* step self-attention is applied within each cuboid in parallel with weight $\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_V}$ that are linear projection of $\mathbf{x^{(n)}}$. So we get

$$\mathbf{x}_{\text{out}}^{(\mathbf{n})} = \text{Softmax}\left((\mathbf{W_Q Q})(\mathbf{W_K K})^T/\sqrt{C}\right)(\mathbf{W_V V})$$

with flattening and unraveling when needed. The computational complexity of the attend step is $O(THWb_Tb_Hb_W)$.

The *merge* step is the inverse of the decompose operation. After applying the attention step to the sequence of cuboids, the cuboids are merged back to the original input shape to produce the final output of cuboid attention.

Cuboid attention patterns are explored in the paper by stacking multiple cuboid attention layers with different choices of cuboid_size, strategy, and shift. These patterns subsume previously proposed space-time attention methods like axial attention, video swin-Transformer, and divided space-time attention. The authors manually selected patterns that are reasonable and not computationally expensive as their search space.

One limitation of the previous formulation is that the cuboids in the system do not communicate with each other, which hinders their understanding of the global dynamics. To address this, the document proposes the use of global vectors $\mathcal{G} \in \mathbb{R}^{P\times C}$, inspired by the [CLS] token in BERT. These global vectors help the cuboids scatter and gather crucial global information during self-attention. This vectors are concatenated with $\mathbf{x^{(n)}}$ in attend step and updated by self-attention with input (K, Q, V): $(\mathcal{G}, \text{Cat}(\mathcal{G}, \mathcal{X}), \text{Cat}(\mathcal{G}, \mathcal{X}))$. The additional complexity caused by the global vectors is approximately $O(THW * P + P)$.

Earthformer adopts a hierarchical encoder-decoder architecture, as illustrated in Fig. 5. This architecture gradually encodes the input sequence into multiple levels of representations and generates predictions through a coarse-to-fine procedure. Each hierarchy consists of D cuboid attention blocks. The encoder's cuboid attention block uses one of the patterns, while the decoder's cuboid blocks adopt the Axial pattern. To reduce the spatial resolution of the input for cuboid attention layers, initial downsampling and upsampling modules are included, which consist of stacked 2D-CNN and Nearest Neighbor Interpolation (NNI) layers.

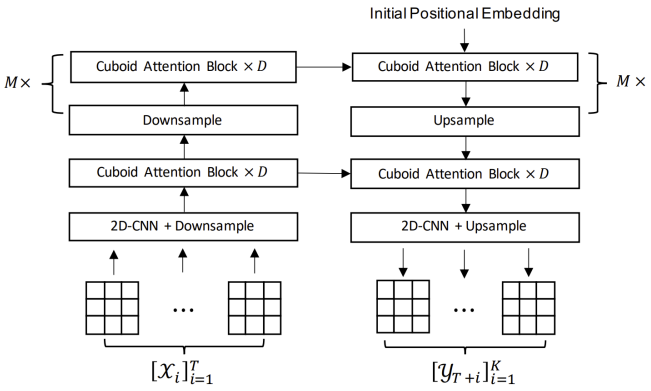

Figure 5. Illustration of the Earthformer architecture. It is a hierarchical Transformer encoder-decoder based on cuboid attention. The input sequence has length T and the target sequence has length K. $\times D$ means to stack D cuboid attention blocks with residual connection. $timesM$ means to have M layers of hierarchies.

3. Proposed enhancements

Given the two-dimensional spatial characteristics of our data, transforming it into a linear format for training linear models presented significant challenges. To mitigate this problem we can use 2 approaches:

1. To construct a model in such a way that it takes into account connections between neighboring points

2. To prepossess the data to make more interactions in new cordinates space

One of the straightforward ways to comply with point 1. is the use of convolutions. But we manage to propose another approach which don't use it.

3.1. Embedding maps

First of all while processing with image we are dealing with patching. It has their advantages and disadvantages. For example we need less number of weights to store and have bigger amount of training data per weight. But on another side we basically are not far away from the problem of the interconnectedness of different parts of the image.

Now image that we want to find universal basis patches that their linear span can express all patches in our dataset. To do this we encode this basis patches in our model's embedding space and try to predict coefficients which sums up to 1 (see Fig 10). In this kind of approach we can search for optimal temperature (T) to make Softmax distribution more sparse or dense (table 1). So partially we solve problem of connectivity because embeddings will learn a complete picture

T emb_maps_len	0.5	1.0	2.0
16	0.823	**0.836**	0.825
32	0.826	0.832	**0.864**
64	0.874	**0.882**	0.876

Table 1. R^2 metric for regression task (horizon=1). The best temperature is highlighted in bold. The more maps we have, the more diverse the linear manifold we can get. At any temperature, increasing the number of embedding maps leads to grow in score. However we can see difference in metric, it's hard to distinguish it on Fig. 11

use_rtl	model data	Linear
False	CentralKZ	0.792
	MadhyaPradesh	0.739
True	CentralKZ	**0.905**
	MadhyaPradesh	**0.883**

Table 2. R^2 metric for regression task (horizon=1). Fig. 14

straightaway (inner connections). But how to deal with independence of patches from each other (outer connections). Let's look at the RTL features.

3.2. RTL like feature

RTL (Region-Time-Length) features was introduced in [8]. Their main idea is to pass about neighbors of each point by precomputing distance and time information. Inspired by this method we use following statistic:

$$\hat{\mathrm{RTL}}(\mathrm{x},\mathrm{y},\mathrm{t}) = \sum_{\substack{i\in(-k_w,k_w)\\ j\in(-k_h,k_h)\\ z\in(-k_t,0)}} \frac{\mathrm{value}(\mathrm{x}+\mathrm{i},\mathrm{y}+\mathrm{j},\mathrm{t}+\mathrm{z})}{\mathrm{distance}^2(x+i,y+j,t+z)},$$

where $k_\cdot$ is kernel size, distance — euclidean relative distance to point at (x, y, t).

The outcome of using $\hat{\mathrm{RTL}}$ is that points on boarder of the patch will know something about points from adjacent patches. It drastically boost performance (see table 2). You can see on Fig [**fig:rtl**] that boarders of patches become more smooth.

3.3. Overall

Table 3 shows that both enhancement improves quality for all models. DLinear gets the best results.

We additionally compare linear models (see Fig 6) by computing following metric:

use_emb_rtl	model data	Linear	NLinear	DLinear
False	CentralKZ	0.183	0.124	**0.654**
	MadhyaPradesh	0.196	0.060	**0.555**
	Missouri	0.435	0.313	**0.857**
True	CentralKZ	0.905	0.905	**0.919**
	MadhyaPradesh	**0.883**	0.818	0.883
	Missouri	0.934	0.935	**0.943**

Table 3. R^2 metric for regression task (horizon=1).

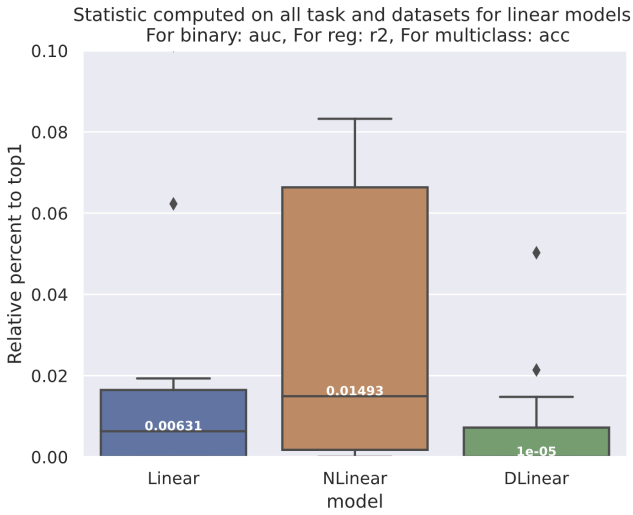

Figure 6. Performance of the linear models

1. For each dataset
2. For each task
3. Calculate relative difference to top1 score (metric name is stated in Fig)
4. Aggregate it in graphic

(Lower — better). We see that DLinear is much more superior in comparison with others.

4. Conducted experiments

4.1. Dataset

The PDSI datasets for the selected regions, representing monthly Palmer Drought Severity Index (PDSI) values, were obtained from the Google Earth Engine service and structured as spatiotemporal tensors with dimensions (T, H, W). In this context, T corresponds to the temporal aspect of the data, while H and W correspond to latitude and longitude,

respectively. This article, in turn, investigates the modeling of these PDSI values for three geographically diverse regions: the state of Missouri (USA), the state of Madhya Pradesh (India), and the Central Kazakhstan region. The geographical distribution of the datasets under investigation is essential for enabling models to grasp the spatial-temporal variations in PDSI values. This geographical diversity not only allows the models to capture the intricate dynamics of drought patterns but also ensures their adaptability to different climatic and geographical conditions, thereby enhancing their robustness and effectiveness in drought assessment and prediction.

Figure 7. Geographical distribution of the studied regions

To address the classification and regression tasks at hand, data preprocessing work was carried out. The preprocessing task involved creating a data window of a length of 16 previous values and a lookahead of 12 future values for prediction. This window served as a temporal context, allowing the model to consider the influence of the previous 16 data points while also making predictions for the next 12 data points. The need for this data window arises from the importance of capturing temporal dependencies within the dataset. By including information from the preceding data, the model gains insight into the historical trends and patterns in the data, enabling it to make more accurate and contextually informed predictions. Additionally, by predicting 12 points into the future, the model provides a forward-looking perspective, which is particularly valuable when dealing with time series data, as it enhances the model's ability to understand and forecast the dynamic nature of the underlying phenomena.

4.2. Experiments setup and evaluation metrics

In our experimental setup, we have employed three distinct tasks: regression, binary classification and multiclass classification, all centered around the prediction of the Palmer Drought Severity Index (PDSI). We use distinct target metrics for each of these tasks.

- **Regression task:** modeling the continuous PDSI data;

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

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

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

- **Binary classification task:** we've chosen a threshold of -2. If the PDSI value is below -2, we label it as "drought", otherwise, its labeled as "non-drought";

$$PRAUC = \int_0^1 \text{Precision}(t)\, dt$$

$$ROCAUC = \int_0^1 \text{Sensitivity}\, d(1 \text{ - Specificity})$$

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Multiclass classification task:** thresholds were set to -2 and 2 (three classes).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

4.3. Models setup

- *Linear/NLinear/DLinear* The configuration is folowing: We use patch with equal width and height = 16, temperature = 1 and number of embedding maps = 16.
- *ConvLSTM* The parameters of the ConvLSTM model in the experiments were set as follows: The hidden dimension in each LSTM layer was set to 24. The size of the kernel in the convolutional layer was set to (3,3). Model consisted of 3 LSTM layers
- *Earthformer* In Earthformer realization we use the encoder-decoder block of depth 1. The number of attention heads in the multi-head self-attention mechanism was set to 4. We addded 16 global vectors to enhance communication among the local cuboids.

4.4. Training setup

All the models were trained for 30 epochs. To optimize the training process, we employed the AdamW optimizer with a learning rate of $1 \cdot 10^{-3}$ ($1 \cdot 10^{-4}$ for EarthFormer) and a weight decay of $1 \cdot 10^{-5}$. The choice of loss functions was tailored to the specific tasks: for regression, we utilized the Mean Squared Error (MSELoss); for binary classification, we employed the Binary Cross-Entropy with Logits

(BCEWithLogitsLoss); and for multiclass classification - Cross-Entropy Loss.

After splitting the dataset into two subsets, with a split ratio of 7:3 for the (training, validation) and testing sets, respectively, we split (train, val) subset again in proportion of 90%/10% to get training and validation data accordingly. All implementations were imported from the official github repositories.

5. Results

Based on the computed metrics during our regression analysis (refer to Table 7), it's evident that ConvLSTM excelled the EarthFormer in predicting the first month. It achieved an R-squared (R^2) score of at least 0.85 for all regions. Moreover, the basic Linear model outperformed it as well as its modificatied models, such as NLinear and DLinear. These modified linear models demonstrated the ability to forecast the forthcoming PDSI distribution for a horizon of up to 6 months with remarkable scores: 0.91 for Central Kazakhstan and Missouri, and 0.82 for Madhya Pradesh, respectively 7. Even though, we believe that ConvLSTM could achieve the same result after the training continues. Conversely, EarthFormer's performance was subpar, which could potentially be attributed to the lack of sufficient data.

Comparable results were obtained in the binary classification task (Table 7). Here, even the standard Linear model exhibited strong performance, as well as its modifications with the incorporation of embedding and RTS features, reaching the AUC score about 0.99 in the horizon of 1 month.

The situation has altered for multicalss problem (thresholds=[-2, 2]) (Table 7). The performance of linear models was subpar in most regions, except for Missouri, where the DLinear model exhibited an accuracy of 0.79. However, the accuracy for the first few months in other regions was notably lower, with values of 0.22 and 0.33.

On the *Figures 10-14* we demonstrate the PDSI values predictions in the horizon of 1 month compared to the target. The results are indicative of the capability of linear models to discern and account for the temporal patterns present in the data. However, it is crucial to note the presence of anomalies or artifacts that emerge when attempting to represent spatial data using linear models. These anomalies could be classified in 3 subclasses:

- **Overall image graininess**:
  It occurs on the stage where we represent our space as separate patches that do not know about their neighbors inside and outside the patch (first stage of implementation of linear models)Fig.15.
- **Grid**:
  When we generate an embedding map (as shown in Fig. 14), the disturbance within all patches vanishes. This enhancement arises because, rather than predicting each individual value (pixel) separately, we calculate and optimize the gradient within each patch. However, it's important to note that the global grid for dividing the image into patches is still maintained. After the incorporation of RTL features (representing distance R, time distance T, and rupture coefficient L), the outputs of the linear models remained largely unchanged. However, the DLinear model noticeably enhances the smoothness of the grid, as illustrated in Figure 13 for the region of Madhya Pradesh.
- **Image borders**:
  This becomes evident when examining the image borders, as the model is compelled to predict a constant value due to the minimal variations observed at these border regions.

The DLinear model stands out as particularly remarkable when compared to the other models that were subjected to testing. Its superior performance and unique attributes make it a standout choice in our analysis. It is able to mitigate most spatial artifacts and strike a balance between predictive power and interpretability.

6. Conclusion

In this study, we trained five types of models to predict PDSI values in three distinct regions. To account for spatial characteristics when training linear models, we introduced an embedding layer to encode the relationships among neighboring values. We also applied a normalization technique by subtracting the last value of the sequence to create NLinear. Finally, we developed DLinear that achieved high metrics that demonstrate its high quality and potential good preformance when applied in other regions for drought prediction.

References

1. *Alley W. M.* The Palmer Drought Severity Index: Limitations and Assumptions // Journal of Applied Meteorology and Climatology. — Boston MA, USA, 1984. — Vol. 23, no. 7. — P. 1100–1109. — DOI: 10.1175/1520-0450(1984)023<1100:TPDSIL>2.0.CO;2.
2. Drought Forecasting: A Review and Assessment of the Hybrid Techniques and Data Pre-Processing / M. A. Alawsi [et al.] // Hydrology. — 2022. — June. — Vol. 9, no. 7. — P. 115. — DOI: 10.3390/hydrology9070115.
3. Are transformers effective for time series forecasting? / A. Zeng [et al.]. — 2023.

4. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting / H. Wu [et al.] // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 22419–22430.
5. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting / T. Zhou [et al.]. — 2022.
6. *Kail R.*, *Zaytsev A.*, *Burnaev E.* Recurrent Convolutional Neural Networks help to predict location of Earthquakes. — 2020. — DOI: 10.48550/ARXIV.2004.09140.
7. Earthformer: Exploring space-time transformers for earth system forecasting / Z. Gao [et al.] // Advances in Neural Information Processing Systems. — 2022. — Vol. 35. — P. 25390–25403.
8. Usage of multiple RTL features for earthquakes prediction / P. Proskura [et al.] // Computational Science and Its Applications–ICCSA 2019: 19th International Conference, Saint Petersburg, Russia, July 1–4, 2019, Proceedings, Part I 19. — Springer. 2019. — P. 556–565.

7. Appendix

model data	Linear	NLinear	DLinear	ConvLSTM	EarthFormer
CentralKZ	0.905	0.905	**0.919**	0.854	0.184
MadhyaPradesh	**0.883**	0.818	0.883	0.870	-0.220
Missouri	0.934	0.935	**0.943**	0.885	0.557

Table 4. R^2 metric for regression task (horizon=1)

model data	Linear	NLinear	DLinear	ConvLSTM	EarthFormer
CentralKZ	**0.991**	0.990	0.985	0.895	0.703
MadhyaPradesh	0.994	**0.995**	0.994	0.922	0.900
Missouri	0.997	**0.998**	0.998	0.773	0.672

Table 5. AUC metric for binary task (horizon=1)

model data	Linear	NLinear	DLinear	ConvLSTM	EarthFormer
CentralKZ	0.228	0.341	0.222	**0.822**	0.502
MadhyaPradesh	0.331	0.330	0.334	**0.831**	–
Missouri	0.784	0.451	0.791	**0.863**	0.617

Table 6. Accuracy metric for multiclass task (horizon=1). The best among linear models is highlighted in blue color.

Table 7. Regression task

		Region								
		Central KZ			Madhya Pradesh			Missouri		
Model	Horizon / Metric	1	6	12	1	6	12	1	6	12
Linear	MAE	0.586	0.590	1.750	0.724	0.775	2.402	0.603	0.644	1.983
	MSE	0.635	0.653	4.953	1.416	1.455	9.919	0.660	0.844	5.929
	RMSE	0.797	0.808	2.225	1.190	1.206	3.149	0.813	0.919	2.435
	R^2	0.905	0.903	0.291	0.883	0.878	0.166	0.934	0.914	0.366
NLinear	MAE	0.566	0.556	1.889	0.952	0.875	2.375	0.564	0.686	2.018
	MSE	0.634	0.598	5.677	2.210	1.799	9.668	0.648	0.861	6.129
	RMSE	0.796	0.773	2.383	1.487	1.341	3.109	0.805	0.928	2.476
	R^2	0.905	0.911	0.187	0.818	0.850	0.187	0.935	0.913	0.345
DLinear	MAE	0.538	0.519	1.880	0.726	0.796	2.364	0.517	0.572	1.919
	MSE	0.544	0.552	5.991	1.422	1.544	9.551	0.575	0.670	5.583
	RMSE	0.737	0.743	2.448	1.192	1.242	3.090	0.758	0.818	2.363
	R^2	0.919	0.918	0.142	0.883	0.871	0.197	0.943	0.932	0.403
ConvLSTM	MAE	0.746	1.793	2.299	0.868	2.356	2.719	0.852	2.065	2.464
	MSE	0.976	5.143	7.115	1.579	9.176	11.392	1.154	6.238	8.788
	RMSE	0.988	2.268	2.667	1.257	3.029	3.375	1.074	2.498	2.964
	R^2	0.854	0.238	-0.019	0.870	0.233	0.042	0.885	0.367	0.060
EarthFormer	MAE	1.903	2.148	2.299	2.969	3.219	3.466	1.681	2.401	2.750
	MSE	5.449	6.775	7.669	14.806	17.175	19.581	4.447	8.853	11.253
	RMSE	2.334	2.603	2.769	3.848	4.144	4.425	2.109	2.975	3.354
	R^2	0.184	-0.003	-0.098	-0.220	-0.435	-0.646	0.557	0.102	-0.203

Table 8. Binary classification task

		Region								
		Central KZ			Madhya Pradesh			Missouri		
Model	Forecast / Metric	1	6	12	1	6	12	1	6	12
Linear	ROC-AUC	0.991	0.994	0.749	0.994	0.993	0.754	0.997	0.999	0.886
	PR-AUC	0.984	0.978	0.656	0.980	0.984	0.569	0.970	0.994	0.791
	F1	0.936	0.940	0.571	0.931	0.945	0.642	0.951	0.968	0.805
NLinear	ROC-AUC	0.990	0.993	0.734	0.995	0.993	0.750	0.998	0.997	0.886
	PR-AUC	0.981	0.971	0.549	0.984	0.976	0.564	0.986	0.983	0.774
	F1	0.929	0.932	0.448	0.938	0.945	0.644	0.957	0.959	0.809
DLinear	ROC-AUC	0.985	0.994	0.796	0.994	0.995	0.718	0.998	0.999	0.900
	PR-AUC	0.955	0.979	0.699	0.977	0.990	0.507	0.988	0.996	0.732
	F1	0.921	0.937	0.605	0.937	0.956	0.305	0.959	0.969	0.817
ConvLSTM	ROC-AUC	0.895	0.747	0.617	0.922	0.596	0.670	0.773	0.666	0.655
	PR-AUC	0.770	0.427	0.401	0.888	0.503	0.525	0.531	0.358	0.444
	F1	0.784	0.022	0.000	0.836	0.532	0.000	0.639	0.194	0.000
EarthFormer	ROC-AUC	0.703	0.631	0.679	0.900	0.692	0.595	0.672	0.787	0.628
	PR-AUC	0.609	0.356	0.547	0.807	0.641	0.482	0.433	0.473	0.331
	F1	0.431	0.283	0.268	0.664	0.320	0.399	0.321	0.123	0.140

Table 9. Multiclass classification task

		Region								
		Central KZ			Madhya Pradesh			Missouri		
Model	Forecast / Metric	1	6	12	1	6	12	1	6	12
Linear	Accuracy	0.228	0.607	0.208	0.331	0.341	0.348	0.784	0.793	0.455
NLinear		0.341	0.226	0.203	0.330	0.695	0.349	0.451	0.624	0.457
DLinear		0.222	0.229	0.204	0.334	0.717	0.349	0.791	0.803	0.451
ConvLSTM		0.822	0.525	0.422	0.831	0.523	0.384	0.863	0.606	0.433
EarthFormer		0.502	0.481	0.500	-	-	-	0.617	0.522	0.437

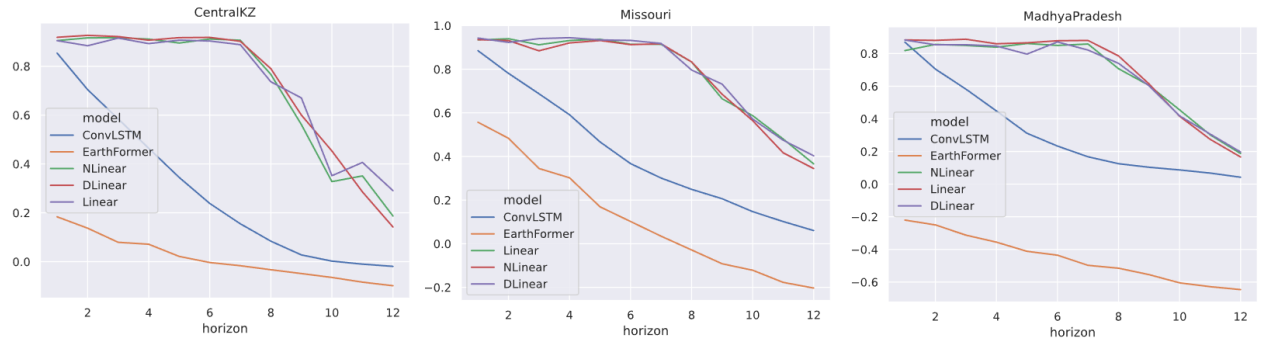

Figure 8. R2 scores in the horizon of 12 months for the regions Central Kazakhstan, Missouri, Madhya Pradesh. In the horizon of 7 months, the linear models outperform ConvLSTM and EarthFormer reaching about 0.9 score each month.

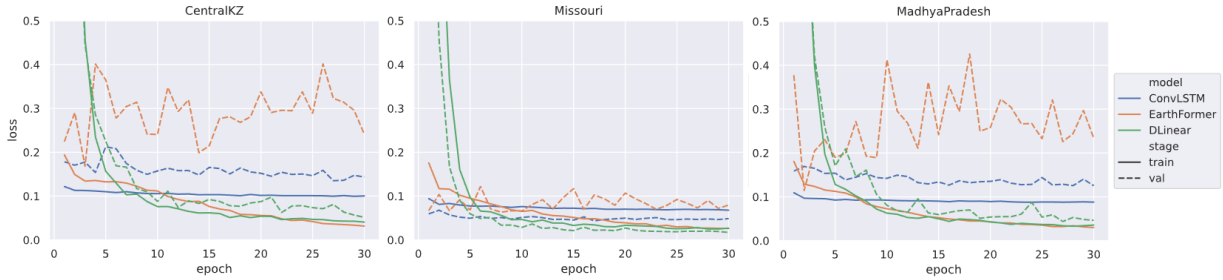

Figure 9. Train and validation losses for regression task. According to graphics for ConvLSTM model there are space to optimize it further.

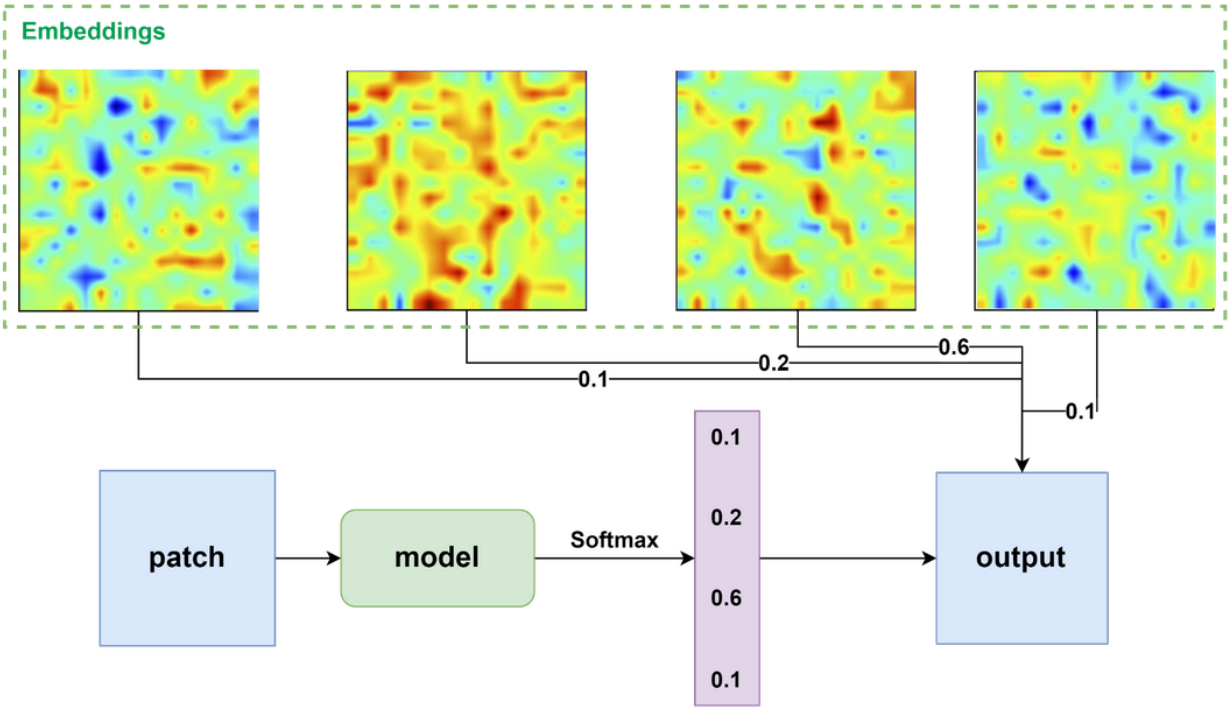

Figure 10. Linear model architecture

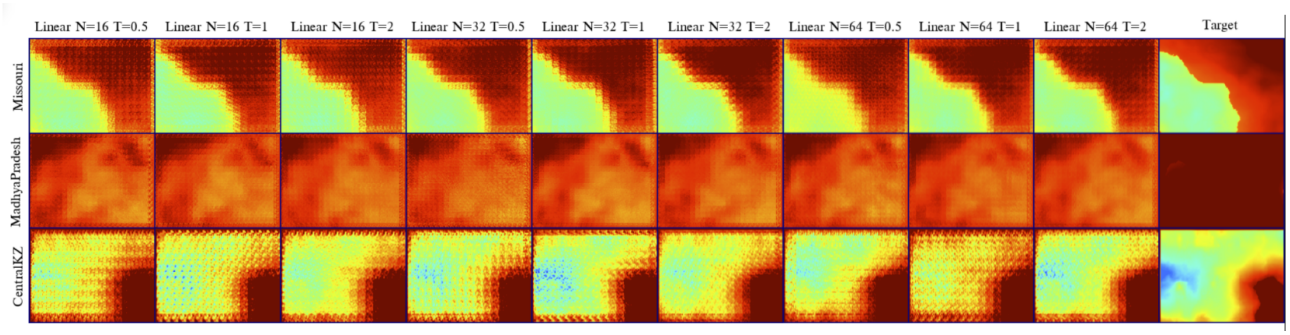

Figure 11. Addition of an embedding layer and temperature

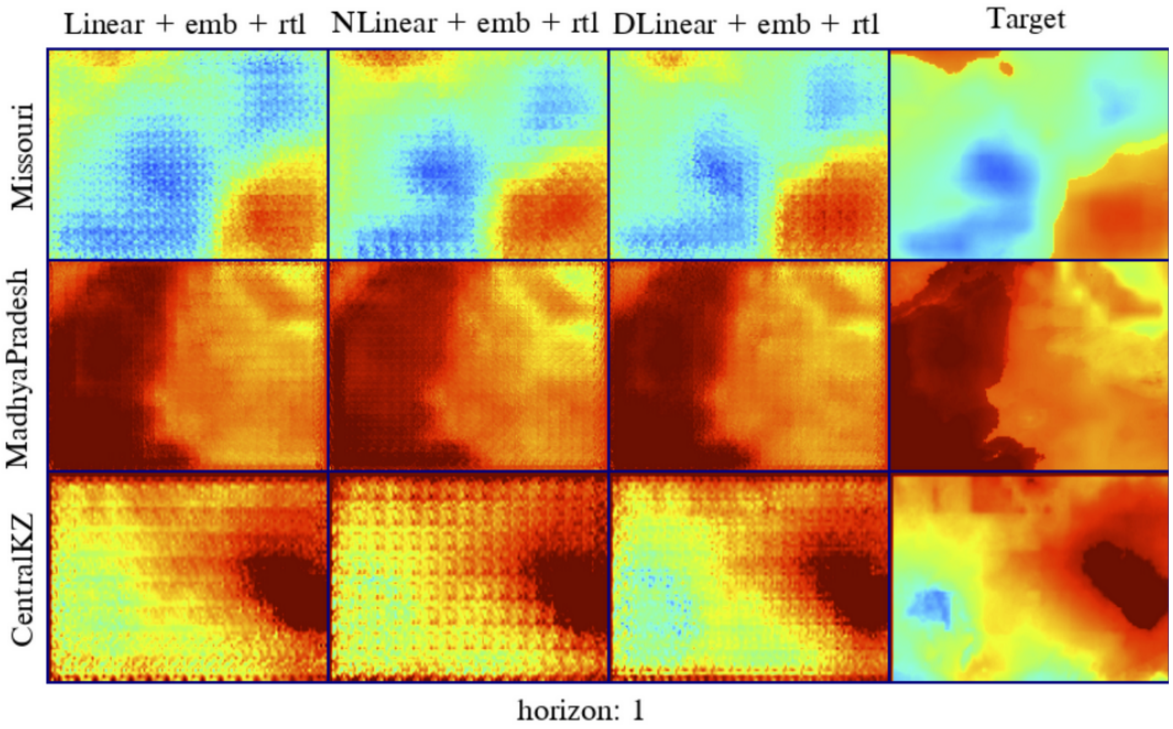

Figure 12. Predictions of linear models with RTL features and embeddings maps

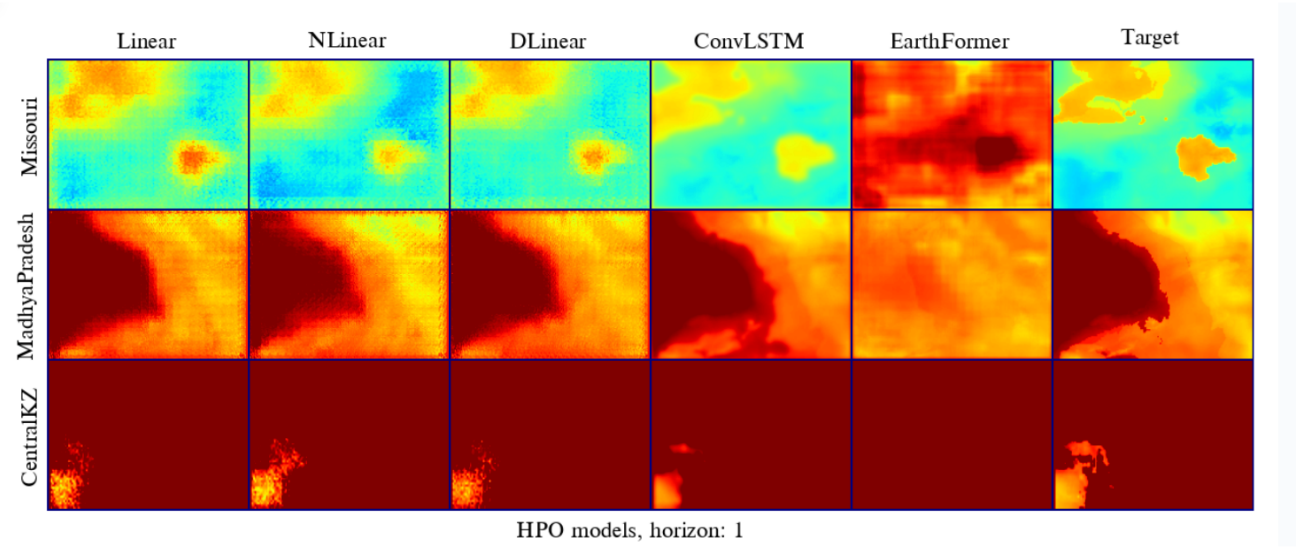

Figure 13. Predictions of all models after modifications

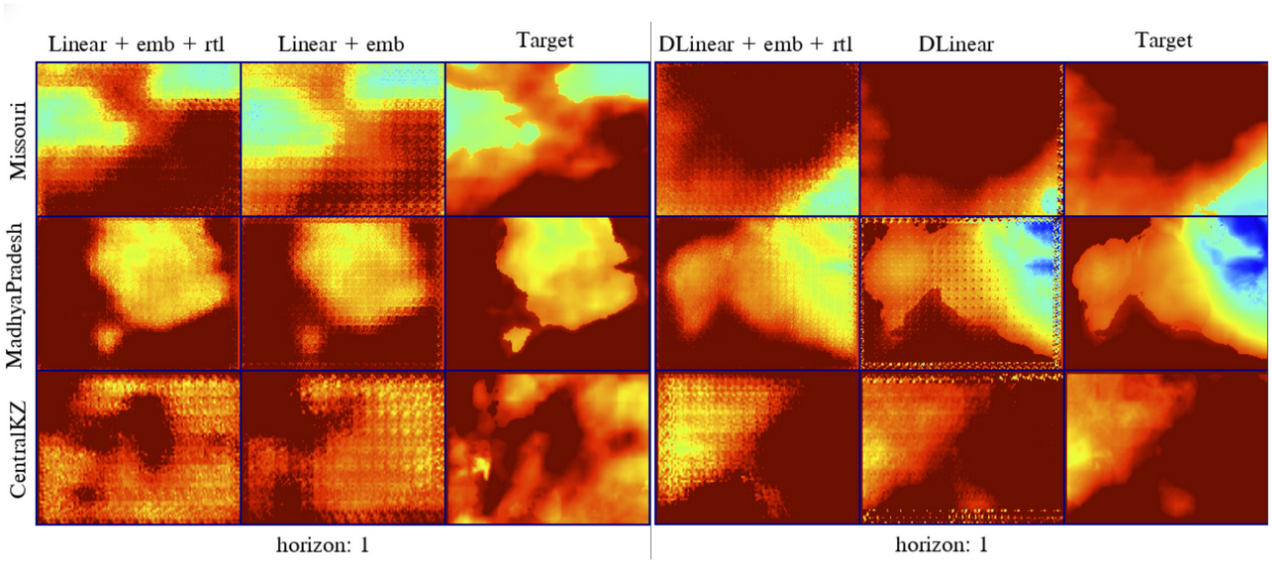

Figure 14. Predictions of models after RTL addition

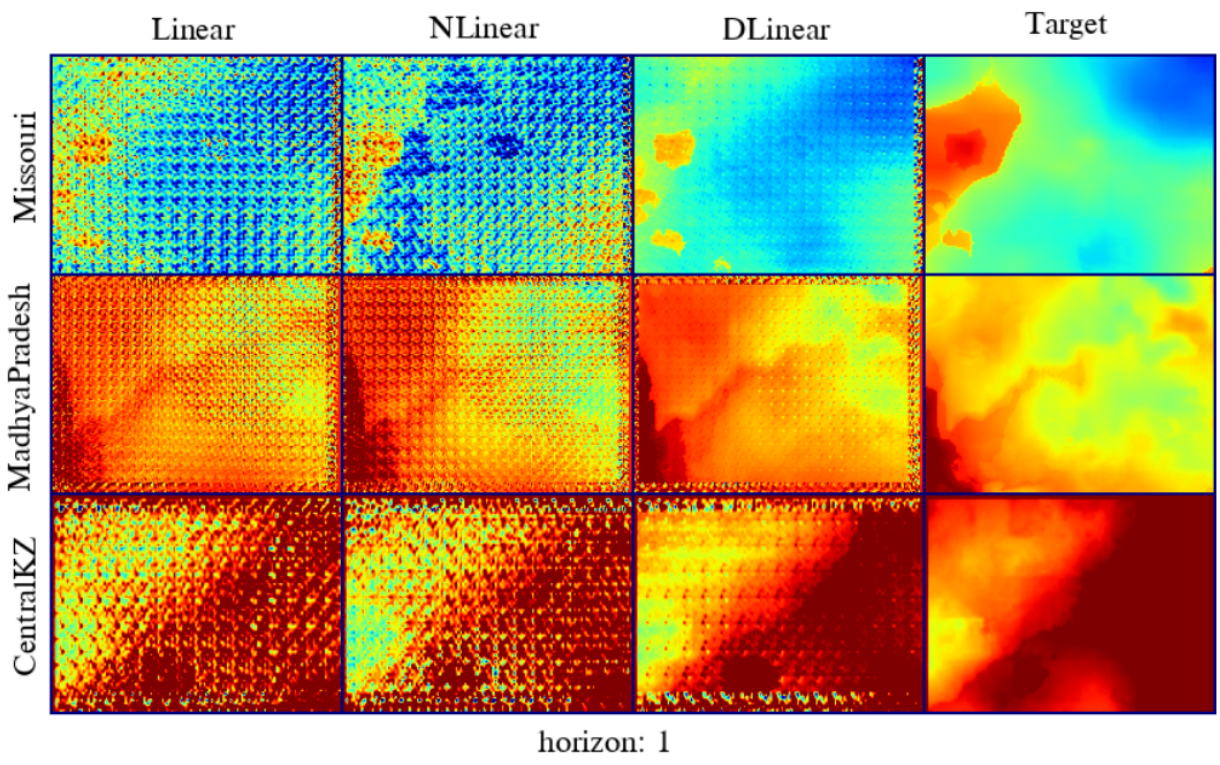

Figure 15. Predictions of linear models

Project plan

Milestones	Doer	Deadline
Study materials and write first report		25/09/2023
1.1. Earthformer	Mikhail Kuznetsov	
1.2. ConvLSTM	Artem Gorbarenko	
1.3. Linear models	Ivan Gurev	
1.4. Introduction + Abstract	Victor Kozhevnikov	
Implementation for:		10/10/2023
2.1. ConvLSTM	Artem Gorbarenko	
2.2. Earthformer	Victor Kozhevnikov	
2.3. Linear models	Ivan Gurev, Mikhail Kuznetsov	
2.4. Benchmarking all models (speed and memory consumption, metrics)	Mikhail Kuznetsov	12/10/2023
Comparing three models and concluding experiments		
Write final project report	All team members	16/10/2023
Make a presentation	All team members	16/10/2023