Investigating Deep Learning Models for Water Level Prediction in the Everglades National Park

Rahuul Rangaraj[1*], **Jimeng Shi**[1*], Azam Shirali[1], Rajendra Paudel[2], Yanzhao Wu[1], Giri Narasimhan[1]

[1]Florida International University

[2]National Park Service

[*]Equal Contribution

FIU FLORIDA INTERNATIONAL UNIVERSITY

BioRG Bioinformatics Research Group

NATIONAL PARK SERVICE

Agenda

1. Introduction
2. Study Domain & Problem Description
3. Deep Learning Models
4. Methodology & Evaluation
5. Results
6. Conclusion & Future Work

Everglades Ecosystem

- Essential **Subtropical Wetland Ecosystem**.
- Water-level forecasting is crucial for **ecosystem management** and **restoration activities.**
- Existing methods struggle, especially during **extreme events.**

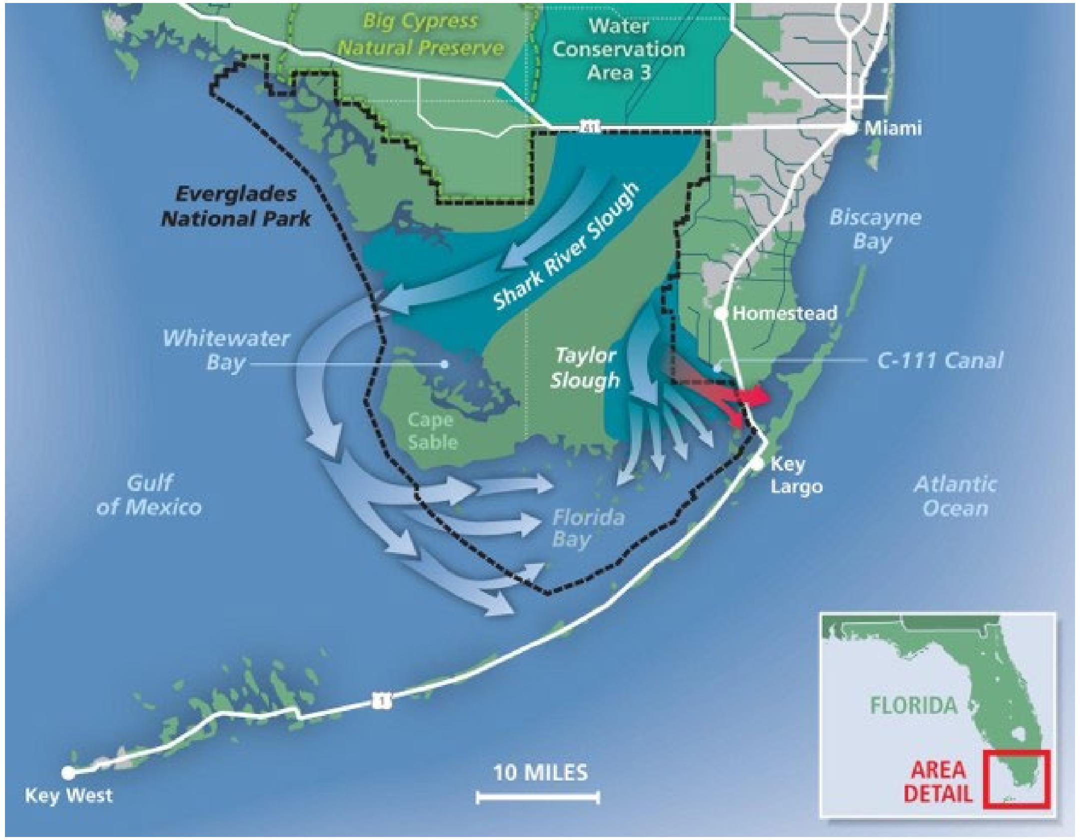


Figure 1: Major water flow paths in Everglades National Park.

Study Domain & Problem Description

- **Task:**
  - Predict water levels at target stations
  - **Considered Inputs:** Rainfall, PET(Potential Evapotranspiration), Gate Flow, Previous Water Levels

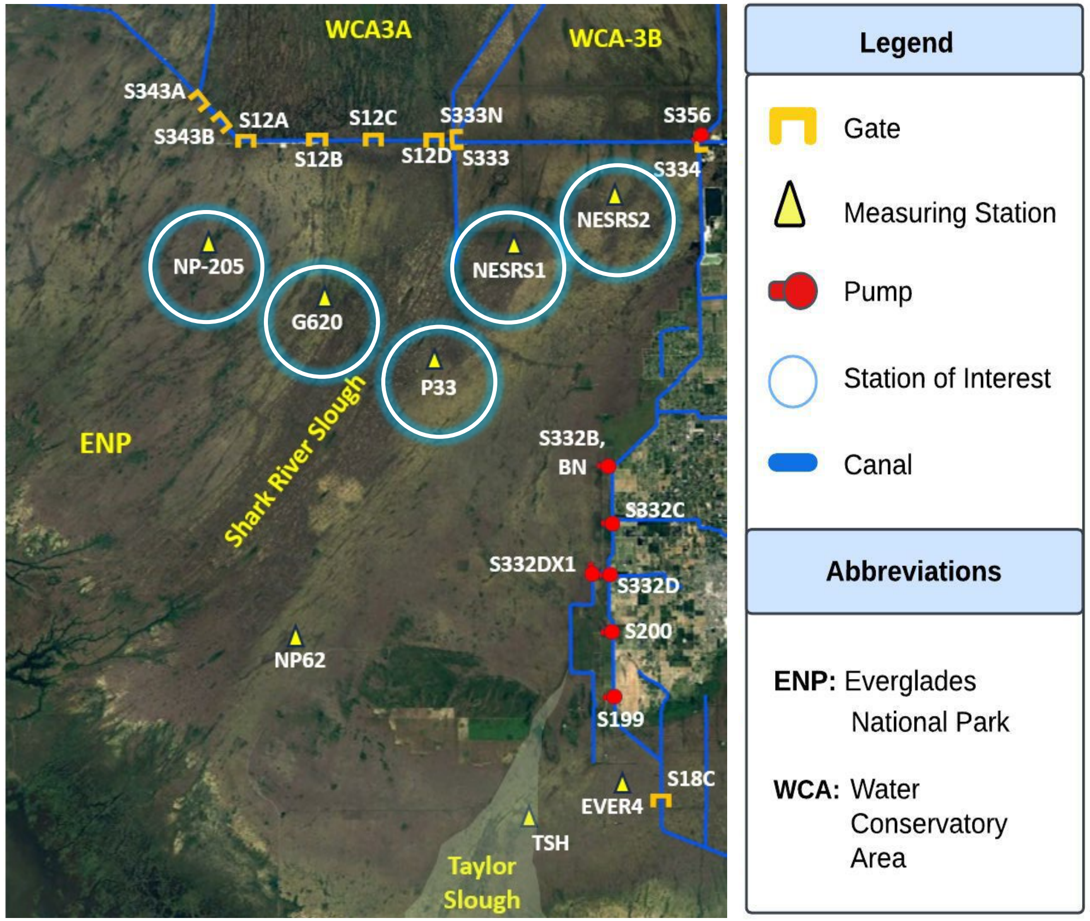


Figure 2: Study domain and selected measuring stations (highlighted).

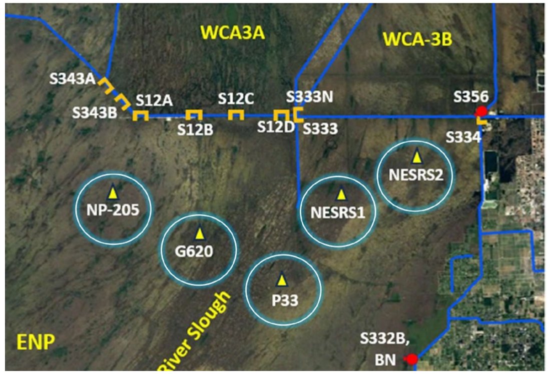
FLORIDA INTERNATIONAL UNIVERSITY

17 Deep Learning Models Examined

- 2 **Linear-based** models
  - **Nlinear[1], Dlinear[2]**
- 4 **MLP-based** models
  - **NBEATS[3], TimeMixer[4], TSMixer[5], TSMixerx[6]**
- 3 **Transformer-based** models
  - **Informer[7], PatchTST[8], iTransformer[9]**
- 2 **KAN-based** models
  - **KAN[10], RMok[11]**
- 1 **LLM-based** model
  - **TimeLLM[12]**
- 5 Time Series **Foundation** models
  - **TimeGPT[13], TimesFM[14], Timer[15], Moirai[16], Chronos[17]**

1. Zeng et al., Are transformers effective for time series forecasting? *AAAI'23*.
2. Zeng et al., Are transformers effective for time series forecasting? *AAAI'23*.
3. Oreshkin et al., N-beats: Neural basis expansion analysis for interpretable time series forecasting, ICLR'20.
4. Wang et al., Timemixer: Decomposable multiscale mixing for time series forecasting, ICLR'24.
5. Chen et al., Tsmixer: An all-mlp architecture for time series forecasting, TMLR'23.
6. Chen et al., Tsmixer: An all-mlp architecture for time series forecasting, TMLR'23.
7. Zhou et al., Informer: Beyond efficient transformer for long sequence time-series forecasting, AAAI'21
8. Nie et al., A time series is worth 64 words: Long-term forecasting with transformers, ICLR'22.
9. Liu et al., itransformer: Inverted transformers are effective for time series forecasting, ICLR'24
10. Liu et al., Kan: Kolmogorov arnold networks, ICLR'25
11. Han et al., Kan4tsf: Are kan and kan based models effective for time series forecasting? arXiv'24.
12. Jin et al., Time-llm: Time series forecasting by reprogramming large language models, ICLR'24
13. Garza and Mergenthaler, Timegpt-1, arXiv'23
14. Das et al., A decoder-only foundation model for time-series forecasting, ICML'24
15. Liu et al., Timer: Transformers for time series analysis at scale, *ICML'24*.
16. Woo et al., Unified training of universal time series forecasting transformers, ICML'24.
17. Ansari et al., Chronos:Learning the language of time series, TMLR'24.

Experimental Settings



Water stages (#Var 15)

Rain (#Var 2)

PET (#Var 2)

Flow (#Var 19)

DL Models

Target Station Water levels (#Var 5)

$$Output \in R^{T \times V} = R^{28 \times 5}$$

- Model: DL model for Time Series Forecasting
- Lookback time frame: 100 days.
- Prediction time frame: 28 days
- Output: Water levels at 5 target stations

Daily Data

$$Input \in R^{T \times V} = R^{100 \times 38}$$

Evaluation Metrics

- **MAE** (Mean Absolute Error):

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

- **RMSE** (Root Mean Squared Error):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^{2}}$$

Overall Performance

MAE: Mean Absolute Error

Task-specific Models	Overall (MAE)
NLinear	0.185
DLinear	0.392
NBEATS	0.176
TimeMixer	0.312
TSMixer	0.186
TSMixerx	0.358
Informer	0.478
PatchTST	0.193
iTransformer	0.198
KAN	0.214
RMok	0.191
TimeLLM	0.242

Foundation Models	Overall (MAE)
TimesFM	0.342
TimeGPT	0.238
Timer	0.385
Moirai	0.364
Chronos	**0.088**

*Chronos surpasses all models for **7,14, 21** days prediction period as well!*

SEDI Metric Analysis

$\hat{y}$: ***Predicted Values***

y : ***Actual Values***

$$\text{SEDI}(p) = \frac{\Sigma(\hat{y} < y_{low}^{p} \;\&\; y < y_{low}^{p}) + \Sigma(\hat{y} > y_{up}^{p} \;\&\; y > y_{up}^{p})}{\Sigma(y < y_{low}^{p}) + \Sigma(y > y_{up}^{p})}$$

y : ***Actual Values***

$\hat{y}$: ***Predicted Values***

- To interpret the results:
  - A **higher SEDI value** (closer to 1) indicates model performs well in identifying extreme events correctly.
  - A **lower SEDI value** (closer to 0) suggests model struggles with correctly identifying extreme events.

SEDI Metric Analysis

y^p_{low} : low threshold value

y^p_{up} : high threshold value

$$\text{SEDI}(p) = \frac{\Sigma(\hat{y} < y^p_{low} \;\&\; y < y^p_{low}) + \Sigma(\hat{y} > y^p_{up} \;\&\; y > y^p_{up})}{\Sigma(y < y^p_{low}) + \Sigma(y > y^p_{up})}$$

y : Actual Values

$\hat{y}$: Predicted Values

y^p_{low} : lower threshold

y^p_{up} : higher threshold

- To interpret results:
  - **Higher SEDI value** (*closer to 1*) – the model performs well in identifying extreme events.
  - **Lower SEDI value** (*closer to 0*) – the model struggles to identify extreme events.

Performance for Extreme Values

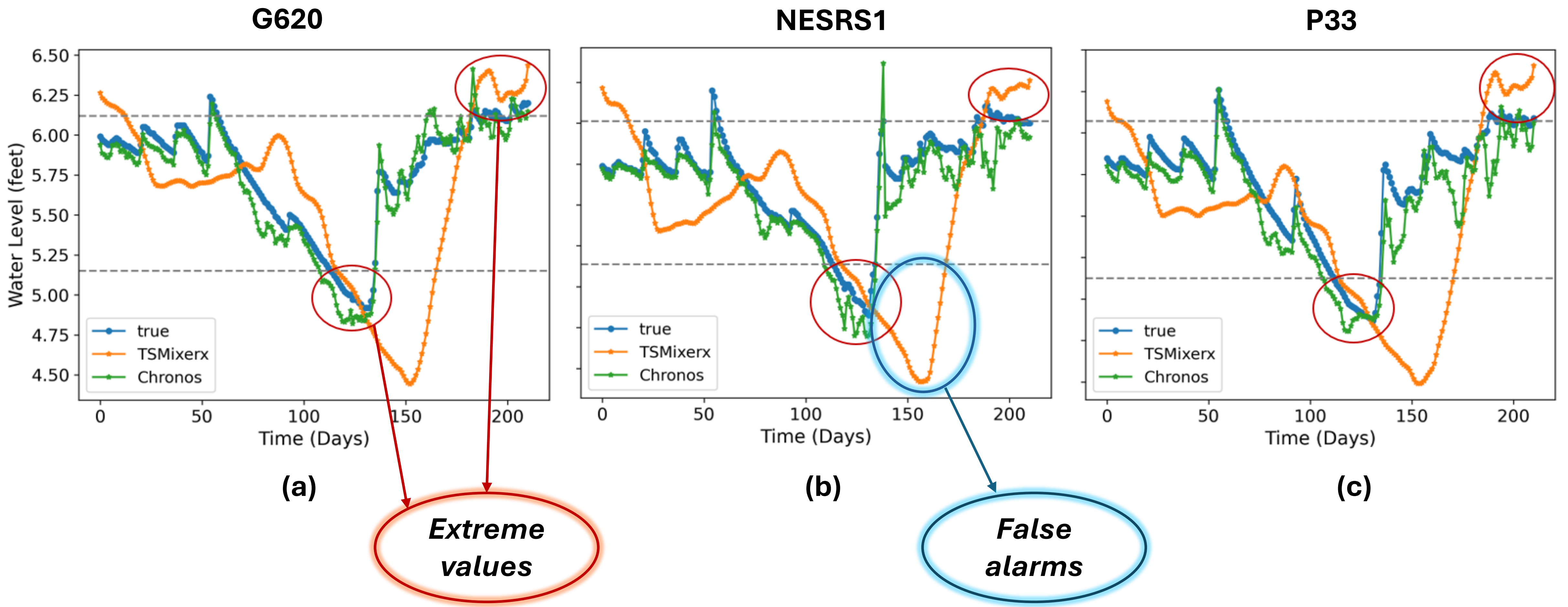
Task-specific Models	Overall (MAE)
NLinear	0.185
DLinear	0.392
NBEATS	0.176
TimeMixer	0.312
TSMixer	0.186
TSMixerx	0.358
Informer	0.478
PatchTST	0.193
iTransformer	0.198
KAN	0.214
RMok	0.191
TimeLLM	0.242

Foundation Models	Overall (MAE)
TimesFM	0.342
TimeGPT	0.238
Timer	0.385
Moirai	0.364
Chronos	**0.088**

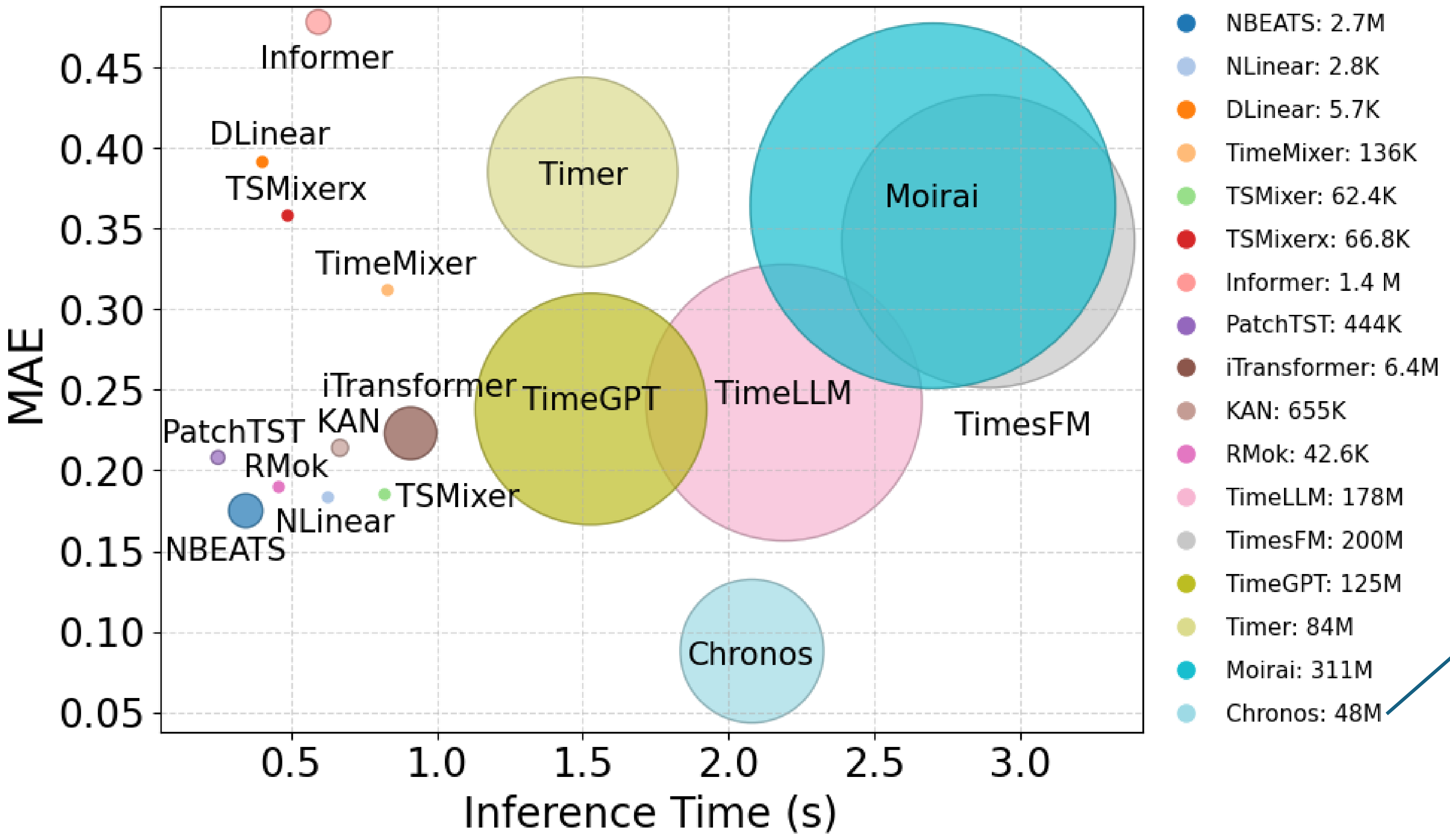
Chronos excels at identifying extreme values

$$\mathbf{SEDI(p)} = \frac{\Sigma(\hat{y} < y^{p}_{low} \;\&\; y < y^{p}_{low}) + \Sigma(\hat{y} > y^{p}_{up} \;\&\; y > y^{p}_{up})}{\Sigma(y < y^{p}_{low}) + \Sigma(y > y^{p}_{up})}$$

Extreme Value Predictions



Accuracy vs Efficiency vs Model Size

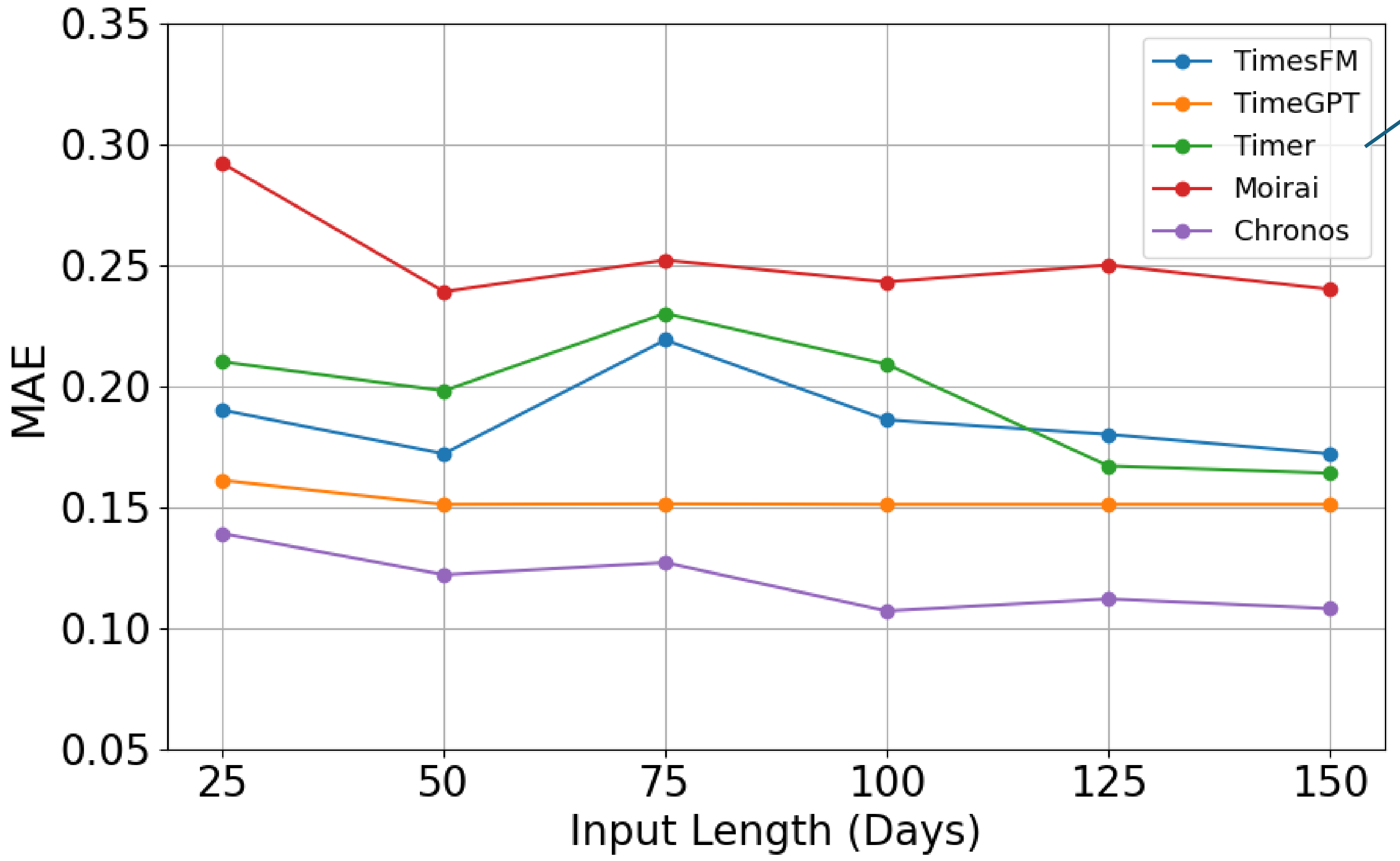


simply scaling model parameters may not improve prediction accuracy

model size does not solely determine overall performance.

Chronos surpasses task-specific models in performance.

Performance vs Input Length



ability to generate predictions for inputs of varying lengths without being retrained.

Differs from task-specific models, requires retraining, when input length changes

MAE values drop as the input length increases from 25 to 100 days

Conclusion

- For time series foundation models,
  - **Chronos** is the best-performing model
  - Unique feature: without retraining for different input lengths
  - Optimal input length for 28-day forecast identified
- For task-specific models,
  - Perform relatively poor
  - Require retraining
- For extreme event prediction,
  - Both model types struggle with extreme event prediction

Future Work & Research Directions

- Retrieval-augmented time series forecasting
  - By retrieving similar past data, the model can use additional context to handle anomalies or trends more accurately
  - **Expected Outcome:** Improved performance on rare or complex events
- Leveraging ensemble methods
- Explainability
  - To understand why a model predicts certain water levels
  - **Expected Outcome:** Greater trust and adoption to see transparent reasoning behind the forecasts

Acknowledgements

- **Dr. Giri Narasimhan** – Research Mentor
- **Dr. Yanzhao Wu** – Faculty Advisor & Project Guide
- **Dr. Rajendra Paudel** – National Park Service
- **Jimeng Shi, Azam Shirali** – Research Collaborators
- **National Park Service & DBHYDRO** – Data Providers
- **Florida International University (FIU)**

THANK YOU!

Happy to take your questions!