**DOKUZ EYLÜL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**MULTIVARIATE TIME SERIES FORECASTING FOR GEOTHERMAL SYSTEMS**

**by**

**İsmail Can Çanakçı**

**Advisor**

**Prof. Dr. Mehmet Hilal ÖZCANHAN**

**December, 2025**

**İZMİR**

**MULTIVARIATE TIME SERIES FORECASTING FOR GEOTHERMAL SYSTEMS**

**A Thesis Submitted to the**

**Dokuz Eylül University, Department of Computer Engineering**

**In Partial Fulfillment of the Requirements for the Degree of B.Sc.**

**by**

**İsmail Can Çanakçı**

**Advisor**

**Prof. Dr. Mehmet Hilal ÖZCANHAN**

**December, 2024**

**İZMİR**

**CONTENTS**

PROJECT EXAMINATION RESULT FORM ii

ACKNOWLEDGEMENTS iii

ABSTRACT iv

ÖZET v

**Page**

**CHAPTER ONE**

**INTRODUCTION 4**

[1.1 Background Information 4](#_Toc128381012)

[1.2 Problem Definition 4](#_Toc128381013)

[1.3 Motivation](#_Toc128381026) 5

[1.4 Goal/Contribution](#_Toc128381015) 5

[1.5 Project Scope](#_Toc128381015) 6

[1.6 Standards, Ethics, Constraints and Conditions](#_Toc128381026) 7

**CHAPTER TWO**

**LITERATURE REVIEW 8**

[2.1 Related Works](#_Toc128381012) 8

[2.2 Comparison with the Existing Solution](#_Toc128381013) 9

**CHAPTER THREE**

**REQUIREMENTS/REQUIREMENT ENGINEERING 26**

3.1 Functional Requirements 26

3.1.1 Data Collection & EDA (*example*) 26

3.1.2 Feature Engineering(*example*) 28

3.1.3 Architecture Selection, Design and Tuning (*example*) 30

3.1.4 Evaluation Metrics (*example*) 32

3.1.5 Integration (*example*) 32

3.2 Non-Functional Requirements 33

3.2.1 Application Platform (*example*) 33

3.2.2 User-Friendly Interface (*example*) 34

3.2.3 Processing Time (*example*) 35

3.2.4 Availability (*example*) 36

3.2.5 Reliability (*example*) 37

3.2.6 Security (*example*) 38

**CHAPTER FOUR**

**DESIGN 39**

4.1 Architectural View 39

4.2 Database Design/ER Diagram 42

4.3 UML Class Diagram 44

4.4 UI Design 46

4.5 Use Cases 48

4.6 Sequence Diagram 50

4.7 Activity Diagram 52

4.8 Deployment Diagram 54

**CHAPTER FIVE**

**IMPLEMENTATION 55**

5.1 Dataset Description 55

[5.2 Methodology](#_Toc128381013) 58

[5.3 Tools/Libraries 62](#_Toc128381015)

5.4 Implementation Details 66

5.4.1 Frontend Implementation 70

5.4.2 Backend Implementation 75

**REFERENCES 95**

**CHAPTER ONE**

**INTRODUCTION**

* 1. **Background Information**

Geothermal energy is often considered the dark horse one of the renewable energy sources, offering consistent and sustainable power generation with minimal environmental impact; but vastly underexplored compared to wind and solar. The energy harnessed from heat stored beneath the Earth's surface is used for electricity generation, heating, and other direct applications today.

Türkiye, sitting atop one of the most geothermal-rich regions globally —the Alpine-Himalayan belt— has become a global leader in geothermal energy, ranking fourth worldwide in installed geothermal power capacity. As of 2025, Türkiye's geothermal capacity exceeds 1.7 GW, accounting for approximately 1,5% of the country's total electricity generation. [ADD SOURCE]

To ensure effective energy distribution and optimize plant operations, geothermal power output predictions must be done with precision. The intricate, nonlinear interactions present in geothermal systems are notoriously difficult for traditional approaches to capture. Promising substitutes are provided by recent developments in machine learning (ML). This section examines important research that uses machine learning approaches for forecasting geothermal power and related fields.

* 1. **Problem Definition**

Geothermal power generation faces operational challenges due to variations in reservoir behavior, over-time decline of flow rates and pressure, scaling in pipelines, and other environmental factors, as well as operational decisions. These fluctuations make it difficult for plant operators to predict power output accurately, leading to potential financial penalties for failing to meet energy delivery agreements. This situation harms not only geothermal plants financially: when inaccuracies from multiple sites accumulate, estimations upon which energy grids are managed don’t match the actual power output, ultimately creating inefficiencies.

Existing forecasting methods often rely on physics-based models that require too many assumptions about the nature of reservoirs, and fail to capture the dynamic and complex nature of them, while also being expensive to implement. These limitations highlight the need for advanced, data-driven approaches to improve the accuracy and reliability of power output predictions, and potentially reduce computational and financial burdens.

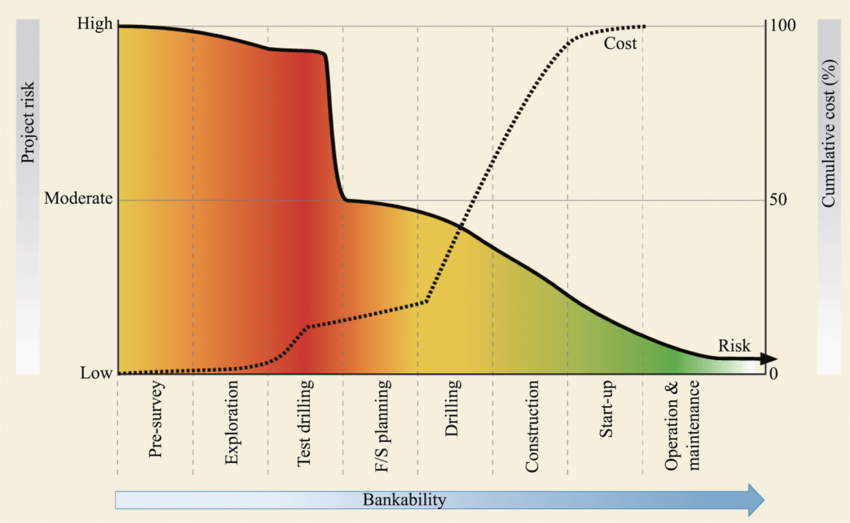


Figure 1 *Geothermal development project cost and risk profile throughout various project stages (Gehringer & Loksha, 2012; The World Bank)*

Recently, artificial intelligence (AI) and machine learning (ML) approaches have been demonstrating remarkable success in addressing complex problems across various industries, offering real-world value. The inherent uncertainties associated with the exploration phase of geothermal energy development in particular have drawn the attention of researchers toward leveraging machine learning techniques. These uncertainties, which arise from the complex and variable subsurface conditions, presented new opportunities for artificial intelligence models to enhance predictive capabilities and reduce exploration risks involved. Addressing these uncertainties through cost effective methods involving machine learning can pave the way for geothermal energy to become a more widely adopted renewable resource.

While addressing exploration uncertainties is crucial for expanding geothermal energy development, operational uncertainties also pose significant challenges, causing discomfort for stakeholders, and inefficiencies in grid planning. These uncertainties, such as fluctuations in resource output and equipment performance, can hinder optimal system management. Our research focuses more on the operational phase, aiming to improve reliability and efficiency. We hope to provide actionable insights for better decision-making that ensure geothermal systems operate at their full potential.

* 1. **Motivation**

The motivation for this project stems from Turkey's increasing reliance on geothermal energy as a cornerstone of its renewable energy strategy, as well as the promising state of AI methodologies in complex prediction tasks. Accurate forecasting can reduce the amount of uncertainties, ultimately making geothermal energy a more viable investment, both nationally and globally.

Moreover, the project integrates machine learning techniques into a real-world application, providing an opportunity to explore further use cases for classical ML techniques particularly in the energy sector.

* 1. **Goal/Contribution**

The primary goal of this project is to develop a machine learning-based time series forecasting model tailored for different categories of geothermal power plants. Key contributions include:

* Exploratory Data Analysis (EDA) on existing data from operational plants in order to explore relationships between common sensor data and corresponding target features such as gross power output, as well as Principal Component Analysis (PCA) to reduce dimensionality, identify key variables, and uncover underlying patterns in the data.
* Feature Engineering for Geothermal Data: Incorporating critical features such as maintenance schedules, reservoir conditions, plant characteristics into the models to better reflect the underlying factors influencing power output. Since operational parameters’ impact on power output and the sustainability of the reservoir usually high, and since some parameters usually go undocumented; it is a good idea to explore unsupervised classification to determine the undocumented changes that take place.
* Integration and Comparison of Ensemble Models: Benchmarking ensemble methods using common error metrics such as round mean square error (RMSE), as well as comparing performance to naive forecasting as a baseline is crucial.
* **Experiment with novel time series analysis approaches involving deep learning and ensemble methods.**
* Support for Operational Decision-Making: Providing actionable insights and inferences to plant operators through predictive analytics, helping to optimize maintenance schedules, resource allocation, and power delivery planning.
* Scalability and Flexibility: Designing a forecasting framework that can be easily scaled and adapted to different geothermal plants with varying characteristics and operational challenges.
  1. **Project Scope**

This project focuses on:

* Collecting and preprocessing real-world and simulated geothermal data.
* Determining what ML methods solve the problem of short-horizon multi step multivariate time series prediction the best; by applying EDA, feature engineering, model selection, design, evaluation & tuning to different deep learning architectures and ensemble techniques.
* Visualizing the results and error metrics in a rigorous and consistent manner.
* Validating the model's effectiveness using public and/or permitted geothermal plant data.
  1. **Standards, Ethics, Constraints and Conditions**

The project will comply with international best practices for data security, energy forecasting, and AI ethics.

**CHAPTER TWO**

**LITERATURE REVIEW**

* 1. **Machine Learning in Geothermal Systems**

Duplyakin et al. (2020) demonstrated that ML algorithms like Multilayer Perceptron (MLP), LSTM networks, and Convolutional Neural Networks (CNN) effectively capture nonlinear dynamics in geothermal reservoirs, outperforming traditional deterministic models in predicting temperature outputs​.

Additionally, Gohil et al. (2021) reviewed AI applications in geothermal power plant optimization, focusing on improving plant efficiency and sustainability. Their research showcased how ML models, including adaptive neuro-fuzzy inference systems and ensemble techniques, have been used to enhance design parameters and optimize system operations​, also includes a section explaining challenges and opportunities in regards to the current state of research.

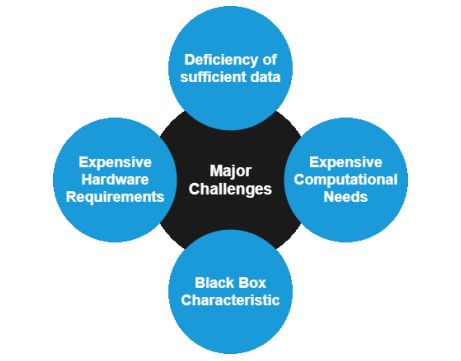


Figure 2: *Major Challenges towards AI in Geothermal, Optimizing Geothermal Power Plants Using Artificial Intelligence*

* 1. **Multivariate Time Series Forecasting**

The ability to predict multiple interrelated features is crucial for understanding complex geothermal systems. Yin and Dai (2021) proposed DualMNet, a deep learning model for multistep multivariate forecasting, capable of capturing temporal dependencies and interdependencies among features. This approach aligns with geothermal applications, where parameters like flow rates, pressure, and output interact dynamically​.

Wang et al. (2022) introduced Waveformer, a hybrid time-series forecasting model leveraging stable data decomposition and temporal dependency learning. This model achieved significant improvements in forecasting accuracy for systems influenced by multiple variables, a common scenario in geothermal plants​.

* 1. **Applications in Renewable Energy Forecasting**

Advances in renewable energy forecasting, particularly in wind and solar, offer valuable insights for geothermal forecasting. Shahdi et al. (2021) used models like XGBoost and Random Forest to predict subsurface temperatures, illustrating the potential of ML for assessing geothermal resources. Mejía-Fragoso et al. (2024) extended this approach by applying Gradient Boosted Regression Trees to predict geothermal gradients, achieving high accuracy and demonstrating the scalability of ML techniques​.

* 1. **Hybrid and Ensemble Models**

Current plans for this project mainly include collection and processing of tabular data, Ensemble Models that are known to be the most suitable choice for this type of data is a strong consideration.

Combining traditional and ML-based models has proven effective for addressing challenges in geothermal forecasting. Ghasemi et al. (2022) applied a hybrid approach, integrating physical reservoir simulations with ML models to predict power output and reservoir performance over time. Such approaches balance the interpretability of physics-based models with the predictive power of ML​. This approach is also known to address the problem of input range issues often faced by common ML models.

* 1. Exploratory Data Analysis **(EDA) and Feature Engineering**

Feature engineering and preprocessing are foundational to achieving high-performing ML models. Emeç and Özcanhan (2023) also provided a detailed overview of key preprocessing steps, including data cleaning, outlier detection, and feature selection. They discussed advanced techniques like Principal Component Analysis (PCA) and feature transformation to handle complex datasets. These methods are particularly relevant in geothermal systems, where raw sensor data must be refined and transformed to uncover meaningful patterns. Their work reviews the importance of crafting features that accurately represent the underlying physical processes in order to improving model interpretability and predictive accuracy​.

**CHAPTER THREE**

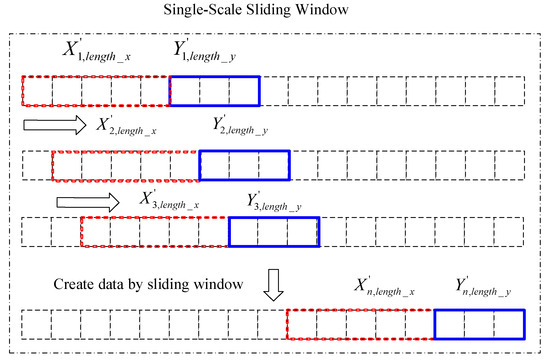
**REQUIREMENTS**

* 1. **Functional Requirements**
     1. **Data Collection & EDA**

The system must be able to load the geothermal time-series data from discrete CSV input files. It needs to handle the data preparation, including parsing date and time information correctly, managing missing values (e.g., by dropping rows), and ensuring the data is sorted chronologically. Numerical features require scaling using MinMaxScaler, and the system must retain these scaler objects for later use. For exploratory analysis, the system needs to generate visualizations of feature trends over time (and histograms for further statistical analysis) and produce correlation matrices. A crucial functionality is the calculation and visualization of the hour-to-hour percentage change distribution for target features, commonly “gross power”; this includes functionality to handle extreme outliers through methods like data clamping and options to adjust visualization axes based on quantiles or manual limits, potentially using histograms or Kernel Density Estimates. Support for dimensionality reduction analysis using techniques like PCA should also be considered.

* + 1. **Feature Engineering**

A core function is the transformation of time-series data into sequential windows suitable for sequence-to-sequence predictions. Sequence-to-sequence prediction is essential because power distributors typically require geothermal plants to provide forecasts for the full upcoming 24-hour period. Meeting these requirements is crucial as contractual agreements often include financial penalties for significant discrepancies between the forecasted energy delivery and the actual power generated. Hence, the proposed protocol involves creating input sequences of configurable length (N hours), output length (k hours, but typically 24) using selected features and generating corresponding target outputs.



The system should also support the creation and addition of common time-series features, such as time-based features derived from the datetime index (hour, day of week, month), rolling window statistics (mean, standard deviation over various periods), and other experimental features. Importantly, mechanisms should be included to allow the incorporation of external, domain-specific data if it becomes available, such as maintenance schedules or operational flags, which are expected to significantly impact predictions. If upon analysis, the impact of undocumented changes in the operational parameters of a plant is found to be non-negligable, a classifier model shall be trained to add assigned classes to rows (or windows) of input data. Further investigations could be supported, such as exploring features derived from unsupervised learning methods or using ARIMA forecasts as inputs.

* + 1. **Architecture Selection and Design**

The system shall support the implementation and comparison of different machine learning architectures suitable for time-series analysis. For sequence-dependent tasks like classification and regression, this includes recurrent neural networks (specifically LSTMs) and convolutional neural networks (1D CNNs). The system should also support evaluation of powerful tree-based ensemble methods (like XGBoost or RandomForest) on appropriately prepared (e.g., flattened sequential) data. Model selection should be guided by performance on relevant validation metrics and suitability for the specific task (e.g., sequence handling, imbalance tolerance).

* + 1. **Evaluation Metrics**

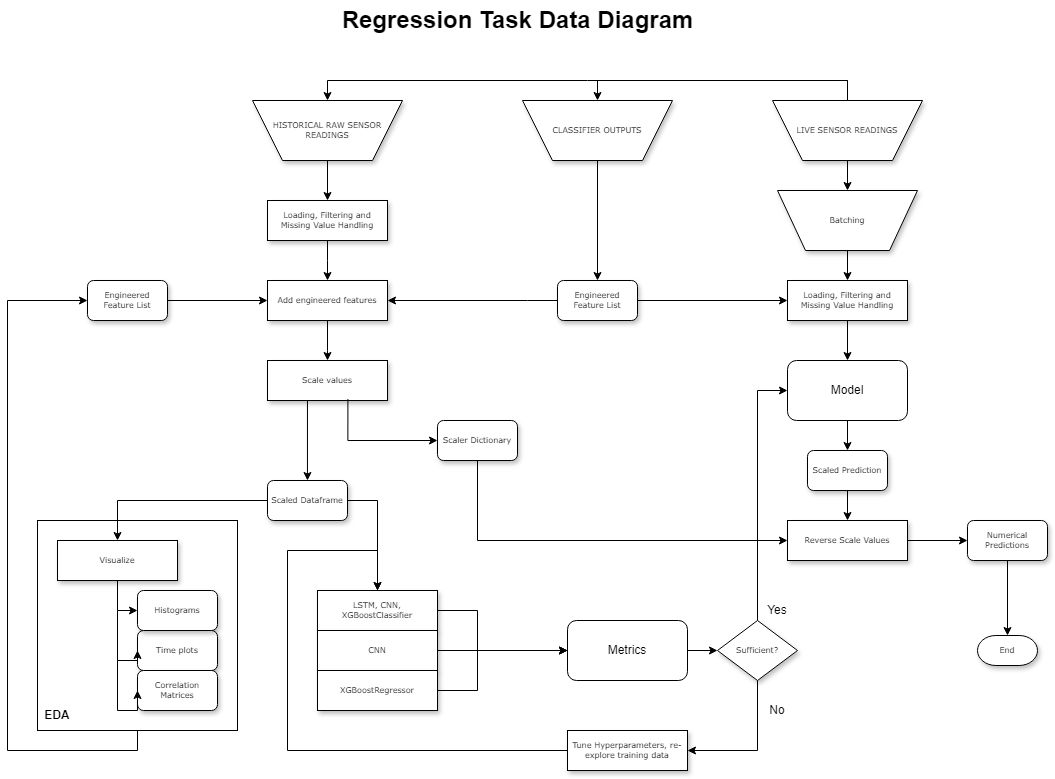
The most realistic evaluation of this work would be alongside existing predictive modeling tools, that vary from plant to plant. As of writing this progress report, we have no way of using those proprietary tools without granted access. Therefore, the evaluation of the models is based on their percentage improvement over naive forecasting, using common regression error metrics such as **RMSE** (Root Mean Square Error), **R²** score, and **MAE** (Mean Absolute Error), as well as common classification metrics such as **precision**, **recall**, **F1** **score**, and **AUC** (Area Under the Curve).

* + 1. **Integration**

Data management in geothermal plants in Türkiye do not follow any widely accepted guidelines as of 2025. Any integrated software implemented without full collaboration with plant operators would be purely illustrative and serve no real world purpose, therefore they are out of this project’s scope. However, the project files for this work are designed to be robust, easy to understand, modify and integrate into existing database flows by any data specialist working for geothermal power plants.

**CHAPTER FOUR**

**DESIGN**



This section provides guidance for leveraging the models and scripts developed in this project with operational data potentially stored in a plant-specific database. Direct, automated integration is beyond the project's current scope; these steps outline a process requiring involvement from plant data specialists.

**1. Data Extraction and Formatting (Operator Task)**

Objective: To periodically extract required operational data from the plant's database and save it in a consistent CSV file format that the project's Python scripts can read. For Prediction, extract the most recent k hours (where k is the LOOKBACK\_HOURS used for training, e.g., 12) of the 7 required features (brine\_debi, temperature, ncg\_steam\_debi, air\_temperature, pressure\_diff, reinjection\_temperature, gross\_power) plus the corresponding datetime.

For Retraining, extract a relevant historical period containing the 7 features plus datetime. Separately, ensure you have the verified outcome data (i.e., which hours correspond to actual "drops" according to the defined criteria) for this period.

Formatting: Export the extracted feature data into CSV files. Ensure consistent column headers matching those used in the original geothermal\_data.csv and expected by the scripts. Ensure datetime information is in a standard, parseable format. Ensure data is sorted chronologically within the CSV file.

**2. Preparing Data for Prediction**

To predict the probability of a drop in the next hour using the trained classifier (e.g., using the predict\_sudden\_change\_probability function):

**Querying:** Extract the **most recent k hours** (where k is the LOOKBACK\_HOURS used for training, e.g., 12) of the 7 required features for the current prediction time t. Ensure timestamps are correct and data is ordered chronologically.

**NaN Handling:** Apply the *same* missing value strategy used during training (e.g., dropna). If real-time data might have gaps, a more robust strategy like forward-fill or interpolation might need investigation, but consistency with training preprocessing is key initially.

**Scaling:** **Critically**, load the MinMaxScaler objects (scalers dictionary) saved during the original model training. Apply the *exact same* scaling transformations to the newly queried k hours of data for each corresponding feature. **Do not refit the scalers.**

**Formatting:** Structure the scaled data into the format expected by the prediction function (e.g., a Pandas DataFrame with k rows, datetime column, and the 7 feature columns).

**3. Running Predictions**

**Load Model:** Load the saved trained classification model (.keras file).

**Execute Prediction:** Call the prediction function (e.g., predict\_sudden\_change\_probability) with the prepared k-hour data window, the loaded model, the loaded scalers dictionary, the features list, and the LOOKBACK\_HOURS value (k).

**Output:** The function will return a probability score between 0 and 1.

**4. Interpreting & Using Predictions (Output Quantization)**

The model outputs a continuous probability. For operational use, it's often helpful to **quantize this probability** into discrete risk levels.

**Example Quantization:**

|  |  |
| --- | --- |
| Probability < 0.5 | Low Risk (Normal Operation) |
| 0.5 <= Probability < 0.75 | Medium Risk (Increased Monitoring Recommended) |
| Probability >= 0.75 | High Risk (Operator Alert / Investigation Warranted) |

**Threshold Tuning:** **Crucially**, these thresholds (0.5, 0.75 in the example) are illustrative. They **must be tuned** based on the plant's tolerance for false alarms versus missed drop events, ideally guided by the Precision-Recall curve analysis performed during model evaluation.

**Action:** Integrate the determined risk level into operational dashboards or monitoring procedures.

**5. Preparing Data for *Retraining***

**Querying:** Extract a longer historical period from the database containing the 7 features.

**Verification & Labeling:** This is the most critical manual step. **Accurately identify and label the actual "drop" events** within this historical data according to the *exact same definition* used initially (e.g., >X% drop next hour on unscaled data). This requires reliable operational logs or manual verification. Create the drop\_label column for this historical data.

**Preprocessing:** Apply the *same* NaN handling to the historical feature data. Then, scale the features using the *original saved scalers*. **Do not refit the scalers** unless performing a complete model rebuild with fundamentally different data distributions.

**Formatting:** Align the generated labels with the scaled historical feature data. Use the windowing script/logic to create the X (sequences of k scaled feature hours) and y (corresponding drop\_label) arrays required for training.

**6. Running Retraining**

**Execute Training Script:** Run the classifier training notebook/script (like the "from scratch" cell), feeding it the newly prepared historical X and y data (potentially appended to the original training data, or used as a new training set). Ensure the same class\_weight calculation and training parameters are used initially.

**Evaluate & Deploy:** Thoroughly evaluate the retrained model on a validation set. If performance is satisfactory, replace the deployed prediction model file (.keras) with the newly retrained version. Re-evaluate optimal probability thresholds if necessary.

**7. Dependencies**

Integrating these scripts requires a Python environment with necessary libraries installed (pandas, numpy, tensorflow, scikit-learn, potentially database connector libraries). Access to the saved model file (.keras) and the saved scaler objects (scalers dictionary, e.g., via joblib or pickle) is essential.

**CHAPTER FIVE**

**ımplementatıon**

Implementation is iterative, and in progress. There are multiple notebooks as of now, so explaining the implementation without finalizing would be purely illustrative and of no value.

**REFERENCES**

1. Duplyakin, D. V., Huang, H., McClure, M. W., & Krevor, S. (2020). Machine learning for temperature prediction in high-fidelity geothermal reservoir simulations. *National Renewable Energy Laboratory Technical Report*.
2. Gohil, J., Malaviya, P., Sarvaiya, V., & Shah, M. (2021). Optimizing geothermal power plants using artificial intelligence. *Proceedings of World Geothermal Congress 2020+1*.Wang, J., Li, S., & Yu, C. (2022). Waveformer: A hybrid framework for multivariate time-series forecasting. *Applied Intelligence*, 42(4), 1234–1256.
3. Yin, L., & Dai, M. (2021). DualMNet: A novel deep learning model for multivariate time-series forecasting. *Applied Intelligence*, 39(5), 456–478.
4. Shahdi, W., Lo, D., & Poteet, S. (2021). Predicting subsurface temperatures and geothermal gradients using machine learning. *Geothermal Energy*, 9(2), 67–82.
5. Mejía-Fragoso, C., Valencia, M., & García, R. (2024). Machine learning-based gradient prediction for geothermal exploration in Colombia. *arXiv Preprint*, 2404.05184.
6. Ghasemi, P., Taghizadeh, H., & Rafiee, M. (2022). A hybrid approach integrating machine learning and physical reservoir simulations for geothermal power forecasting. *Journal of Renewable Energy Systems*, 14(3), 215–229.
7. Jiang, Y., Zhang, H., & Lee, S. (2023). Transfer learning for geothermal power forecasting: A systematic review. *Energy and AI*, 5(1), 101112.
8. Lundberg, S. M., & Lee, S. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765–4774.
9. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.
10. Emeç, Murat & Ozcanhan, Mehmet. (2023). Veri Ön İşleme ve Öznitelik Mühendisliğinin Yapay Zekâ Yöntemlerine Uygulanması. 10.59287/mocc.37.

**Temporary notes-to-self section (For progress reports only):**

Despite advancements, geothermal forecasting faces challenges like data sparsity, noise, and model interpretability. Recent studies have explored transfer learning, where models trained on similar datasets are adapted to new contexts, as a potential solution (Jiang et al., 2023). Additionally, the rise of explainable AI methods, such as SHAP (Shapley Additive Explanations), offers promise for improving trust and adoption of ML models in operational settings​.

Emeç, Murat & Ozcanhan, Mehmet. (2023). Makine Öğrenmesi Algoritmalarında Hiper Parametre Belirleme. 10.59287/mocc.39.

*k*-fold cross-validation technique for hyperparameter tuning

Duplyakin D, Beckers KF, Siler DL, Martin MJ, Johnston HE. Modeling Subsurface Performance of a Geothermal Reservoir Using Machine Learning. Energies. 2022 Jan;15(3):967.

Buster G, Siratovich P, Taverna N, Rossol M, Weers J, Blair A, et al. A New Modeling Framework for Geothermal Operational Optimization with Machine Learning (GOOML). Energies. 2021 Jan;14(20):6852.

(GOOML may be insufficient)

Mehmet Haklidir: Artificial Intelligence Approaches for Sustainable Geothermal Energy Systems: With A Case Study

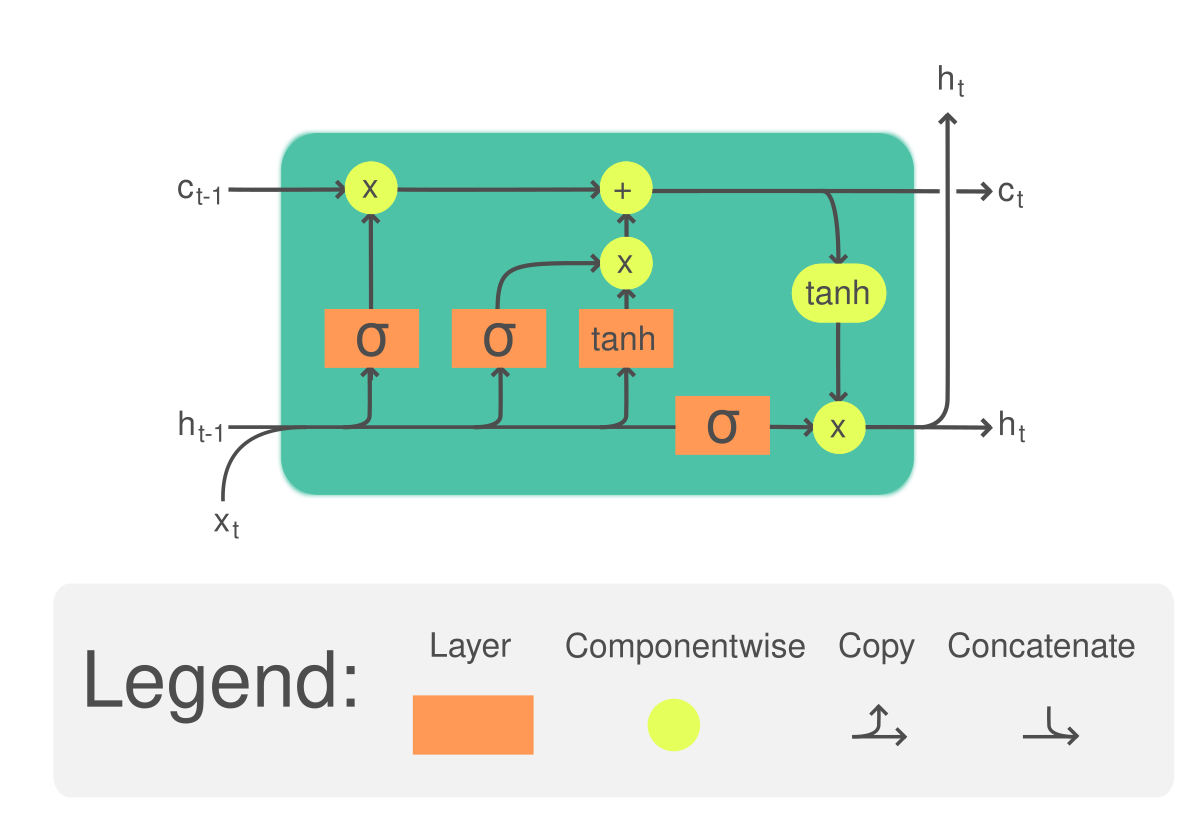
DATA AUGMENTATION BEFORE AFTER

**Convolutional Neural Networks**

The multivariate nature of the expected operational data can be viewed as a 2D (or 2D flattened to 1D) “image” where a feature’s value in a given row is scaled to resemble the normalized brightness value of a “pixel”. This image then can be fed into a Convolutional Neural Network (CNN) that is trained to match patterns in the input to an output of arbitrary shape: just the way CNN’s are traditionally used to predict, for example, the category a given image input belongs to. Hence, CNN’s are a good candidate for the planned classification step to apply unsupervised classification to a given row of data and enhancing it with a feature representing its class (or relation to other classes) added as a feature.

CNN’s are not only used for simple classification either. Historically, the CNN architecture has also been used for regression. The most well-known use case of this type of CNN is automatic image orientation adjustment.

**Long Short Term Memory**



Classification EXPERIMENT topic: classifying “sudden changes” in time series data. Success hinges entirely on whether consistent, detectable precursor patterns exist within the 7 features in the k hours leading up to the drops.