Optimization of a Solar Thermal Plant integrated to an industrial process using Machine Learning and Genetic Algorithms

Ana Valentina Puentes[1], Francisco Fuentes[1], Iván Muñoz[1] and Francisco Moraga[1]

[1] Fraunhofer Chile Research, Santiago (Chile)

Abstract

Given Chile's abundant solar resource, solar thermal energy generation has great potential but must be economically viable. This work aims to develop and apply a Machine Learning–based model to optimize the design of solar thermal plants integrated into industrial processes. The methodology involves hyperparameter tuning, testing preprocessing methods, designing optimization algorithms, and validating results through new simulations. Neural Networks and Decision Trees were trained using two datasets, uniform and random, within defined ranges. Various preprocessing techniques were tested to find the most accurate Machine Learning model and Genetic Algorithms were applied for the optimization model. The results showed Standard Scaler preprocessing produced the lowest prediction errors, especially for Neural Networks trained with the uniform dataset, achieving a 0.12% error, Neural Networks outperformed Decision Trees, and uniform datasets showed stronger interpolation. The study concludes that multi-objective optimization with Neural Networks trained on uniform data yields errors below 2%, confirming their reliability for thermal plant design optimization.

Keywords: Solar Thermal Plant, TRNSYS-18, Optimization, Neural Networks, Genetic Algorithms

1. Introduction

One of the major global challenges humanity faces is pollution, largely driven by the burning of fossil fuels for energy generation, whether thermal, electrical, or otherwise. These processes release significant amounts of carbon dioxide and greenhouse gases, contributing to global warming. To address this, the Chilean government has set decarbonization goals, aiming for carbon and greenhouse gas neutrality by 2050. Achieving this requires the development of cost-effective and sustainable solutions, particularly for generating thermal energy to meet industrial demand.

Chile stands out for its exceptional solar resources, with average global horizontal and direct normal irradiance values of 5.33 and 6.96 kWh/m²/day, respectively (Ministry of Energy, Chilean Government). Given that many industrial processes in the country operate below 400°C, solar thermal technologies offer a viable option to meet energy needs sustainably. This thesis aims to contribute to the decarbonization process and the reduction of greenhouse gas emissions by optimizing the design of solar thermal plants. The goal is to shorten design times and enhance the techno-economic appeal of these systems through advanced computational tools and automated optimization algorithms. Specifically, it seeks to evaluate a methodology and a Machine Learning model to optimize solar thermal plants integrated into industrial processes.

2. Background and Literature Review

Solar thermal energy can be used to meet the demand in heating, commonly of low or medium temperatures, and cooling processes across various industrial sectors, which is usually generated by fuel or other sources that are damaging to the environment. But this integration is more complex than conventional technologies, meaning that to make this efficient, there are multiple factors and variables that need to be attended to achieve a technical and economic optimal. A form to quantify the techno-economic viability for solar thermal plants is using the Levelized Cost of Heat (LCoH), that gives the cost per generated heat unit as the following equation:

$$\mathrm{LCoH} = \frac{I_0 + \sum_{t=1}^{T} \frac{C_t}{(1+r)^t}}{\sum_{t=1}^{T} \frac{E_t}{(1+r)^t}} \qquad \text{(eq. 1)}$$

10.18086/swc.2025.02.21 Available at http://proceedings.ises.org

2.1 Optimization using Machine Learning algorithms

The implementation of Machine Learning (ML) models for optimization can be applied to various areas. Weichert, D et al. 2019 conducted a study on this for production processes, which can be applied to energy saving, time reduction resource optimization and waste avoidance. For optimization with parameters changes, the study was based on generating the database, then obtain and train a ML model capable of predicting the physical behavior of the problem at hand, to finally, optimize these parameters using the mentioned behavior.

Lee, J. et al. 2021 used a similar method to optimize hydrogen production using ML models able to predict the performance of the plant. The particularity of this study was the use of hyperparameter (HP) tuning to search and optimize the combination to reach a model with good performance. To finalize using the found model to optimize the problem based on the outputs obtained from the model. The same methodology is used by Magnier, L. and Haghighat, F. 2010, but with the objective to optimize thermal comfort and energy consumption in buildings.

These methods can be applied for optimizing solar thermal energy plants, for minimizing costs making the project profitable. Fadlallah, S et al. 2021 used an Artificial Neural Network (ANN) model and Particle Swarm Optimization (PSO) as an optimization algorithm to optimize the use of material for heliostats. This study uses a loop to obtain the ANN architecture with low error, to then use the optimization algorithm, where the fitness function is calculated with the outputs, until reaching a global optimal. Kalogirou, S. 2006 applies a simpler method using GA as the optimization algorithm, to minimize the tank volume in correlation with the solar collector area. Finally, Dallapiccola, M. et al. 2020 and Starke, A. et al. 2018 applied a multi-objective optimization using NSGA-II to optimize hybrid solar thermal and photovoltaic plants, with the difference that the first one Decision Trees (DT) and the other ANN models. In Fig. 1 a general scheme adapted from Dallapicola, M. et al. 2020 of the presented methodologies can be observed.

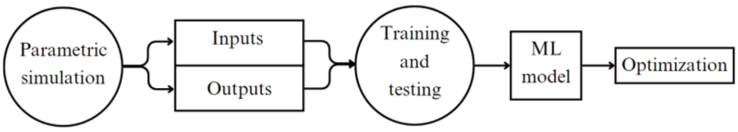

Fig. 1: Scheme for the ML and Optimization algorithm

3. Methods

3.1. Generation and configuration of the datasets

The first step, to train the ML algorithms, is to get the datasets that identifies the solar thermal plant. For this study two datasets were utilized, with the same configurations of five inputs and outputs, for comparing and evaluating performance differences. These were obtained using TRNSYS 18, specifically for the solar thermal plant presented in Fig. 2.

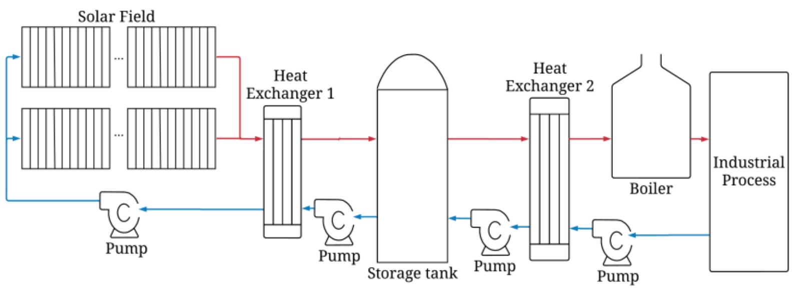

Fig. 2: Solar Thermal plant diagram

The inputs are the area of the solar field (Col_A) [m²], specific volume of the tank (Spec_V) [m³/m²], inclination angle (β) [°], the specific mass flow through the solar plant (Mass_F) [kg/hm²] and the effectiveness of the heat exchangers (Eff_HX) [-]. It is noteworthy to mention that the tank volume (Vol_T) [m³] was the input employed for model training and optimization. This was derived from Spec_V and Col_A calculated as:

$$Vol\_T = Col\_A \cdot Spec\_V \; [m^3] \qquad \text{(eq. 2)}$$

Additionally, the outputs of the datasets were also essential and necessary for model training, allowing them to learn the outcomes and predict results. These were the energies from the plant associated with specific inputs: the incident irradiation to the solar field (Irr_E), the energy transferred to the fluid in the solar field (Fluid_E), the energy transferred in the heat exchangers 1 and 2 (Heat_1 and Heat_2, respectively) and the supplied energy from the boiler to the process fluid (Aux_E), all measured in kWh/year.

The datasets differentiate in the total configurations for each one, and for the numerical sets that the inputs are from, that are depicted in Tab. 1. The first one being equally spaced values, and the second one being randomly selected within a specified range.

Tab. 1: Amount and sets of the data for each dataset

Values	First Dataset	Second Dataset
Total configurations	7761	27559
Col_A [m²]	{4000, 5000, 6000, 7000, 8000, 9000, 10000}	[4000, 60000]
Spec_V [m³/m²]	{0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2, 0.225}	[0.05, 0.35]
β [°]	{0, 10, 20, 30, 40, 50}	[0, 60]
Mass_F [kg/hm²]	{40, 50, 60, 70, 80}	[40, 100]
Eff_HX [-]	{0.5, 0.6, 0.7, 0.8, 0.9}	[0.5, 0.9]

3.2. Preprocessing and data segmentation

To ensure a correct learning from the models, and to compare and analyze the best one for this case, different data preprocessing techniques were applied to the inputs, using methods defined from Scikit-Learn (Buitinck, L, et al. 2013). The first ones being MMS, MAS and SS, and PCA from five (all the inputs) to three (PCA3) and two dimensions (PCA2). The original outputs (Original) were also compared.

Given that two ML models were explored and analyzed, ANN and DT, the datasets were partitioned into test, validation and training sets as appropriate. For ANN models, the datasets are divided as 72% training, 18% validation and 10% test, and for DT models, the split is 70% train and 30% test.

3.3. Hyperparameter tuning and Model selection

Once the datasets were prepared, HP tuning was conducted to build the ANN and DT models, conducted for both datasets with all the different preprocessing applied, meaning optimizing HP for ten models each one.

For the ANN, a function capable of building the models architecture was defined using Keras (Chollet, F. 2019). The function initializes the sequential model, then adds dense layers, first the input layer, the hidden layers, and the output layer, the respective nodes and activation function for each one of the hidden layers, and learning rate, depending on the HP selected. Finalizing with Adam optimizer and Mean Absolute Percentage Error (MAPE) loss. Consecutively, Keras-Tuner (O'Malley, T. Et al. 2019) was used, with RandomSearch for 1000 trials (HP selected randomly within their ranges) and three cross validations, with the objective of minimizing the validation loss. For the DT models, the tuning is conducted using GridSearch from Scikit-Learn, which explores all the possible combinations of models. GridSearch requires the regression DT model, HP in a Python dictionary specifying their respective ranges, the scoring, selected as MAPE, and five cross validations. In this case, were optimize the Criterion, Min_Samples_Split and the Splitter.

After the HP tuners were defined, a while cycle is conducted. This re-trains the models and compares the validation loss or score, to obtain the final models that, hopefully, does not depend on the initialization of weights. This cycle takes the 10% best models for ANN or the 30% best for DT of the total that were obtained by the tuner, which are re-trained and takes another 10% or 30% until one model remains, this methodology is depicted in the diagram of Fig. 3. Finally, the last model is re-trained 10 times, to obtain the mean and standard deviation of the losses to compare the models, and the ones with lower loss and more capability of prediction between preprocessing are selected to conduct the optimizations.

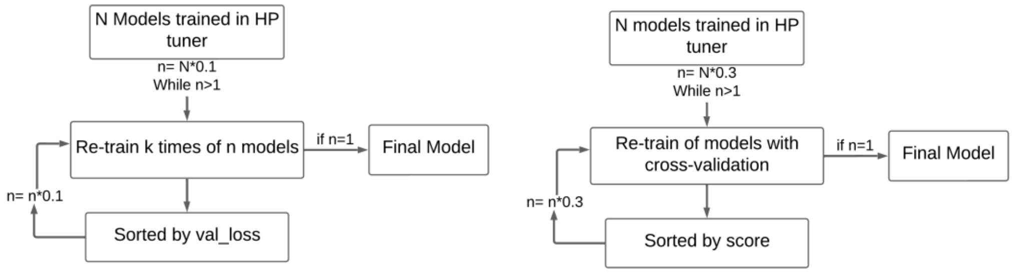

Fig. 3: Machine Learning model selection for ANN and DT models

3.4. One Objective Optimization with Genetic Algorithms

Obtained the final models for each dataset, meaning two for ANN and DT, the one objective optimization was conducted using the PyGad library (Gad, A. 2021), defined as the minimization of the LCoH achieved by the thermal solar plant, and subjected to the efficiency and solar fraction being greater than 50%. Also, a sensibility analysis was conducted for the fuel cost for 30, 50 and 90 [USD/MWh], and for the solar collectors cost for 200, 250 and 300 [USD/ m²], meaning twelve optimizations per model. For this algorithm it is important to note that for each individual (inputs to calculate the LCoH), it must be applied the same preprocessing calibrated for training. Given the defined problem, considering the GA always maximizes the fitness function, this was defined as the following equation:

$$\text{Fitness} = \frac{1}{1 + \text{LCoH}} \qquad \text{(eq. 3)}$$

The LCoH was calculated using eq. 1, where the initial investment was calculated with the Col_A and Vol_T using their respective costs, 1000 [USD/ m³] for the latter, giving a 20% increment to estimate the total. Subsequently, the corresponding preprocessing was applied to the inputs for using the ML model to predict the associated energies. Considering a 20 year operation for the plant, the operational costs were estimated multiplying Aux_E by the fuel cost adding 2% of the investment, and finally, the energy demand was calculated adding Aux_E and Heat_1.

Then, the GA was initialized with a population of 500 individuals, where each input is random within their respective ranges, 500 generations, 150 number of parents, 25 for elitism, and functions from the library, “Russian Roulette for parent selection and “Single Point” for crossover with a probability of 80%. Also, a personalized mutation function was defined where each input is changed, with a 30% probability, within their respective range.

3.5. Multi-objective optimization with Nondominated Sorting Genetic Algorithm-II

With the same models used before, and doing the same sensibility analysis, the multi-objective optimization was defined as minimizing the LCoH and the boiler energy subjected to the efficiency and solar fraction being greater than 50%. For this the pymoo library (Blank, J. and Deb, K. 2020) was used with NSGA-II.

This algorithm minimizes the fitness, one defined as the LCoH the other the Aux_E. Then, the algorithm was initialized using 700 individuals per population for 100 generations and using the parameters of “Random Sampling” for sampling and “Russian roulette” for crossover with 80% probability. For the mutation function Polynomial Mutation was used with η equal to 20, with a probability of 30%. Finally, the best individual was selected by the 80/20 rule, based on the Pareto principle, giving priority to the LCoH.

3.6. Results Validation

As the final step, a result validation was conducted by modeling the twenty-four configurations from both optimizations using TRNSYS-18 to obtain the outputs. Then, these energies were compared from the ones of the ML models, and the LCoH calculated.

4. Results

4.1 Artificial Neural Network model selection for first dataset

After the tuner of HP was completed, six models were obtained, one for each preprocessing. In Tab. 2 are the values for the statistics after the re-training.

Tab. 2: Statistic data for ANN models trained with the first dataset

Statistic	MMS	MAS	SS	PCA3	PCA2	Original
Mean loss [%]	4.53	8.46	0.19	3.27	5.52	29.35
SD loss [%]	7.98	9.75	0.03	0.11	0.11	35.92
Mean val_loss [%]	4.44	8.46	0.16	3.60	5.31	29.69
SD val_loss [%]	7.93	9.75	0.03	0.51	0.31	35.73

Subsequently, for all the ANN models, in Tab. 3 are shown the errors obtained from testing the model, meaning the comparison between the testing set outputs with the ones predicted from the model.

Tab. 3: Testing errors for ANN models trained with the first dataset

Error	MMS	MAS	SS	PCA3	PCA2	Original
Global [%]	0.17	0.09	0.12	2.76	5.05	4.91
Irr_E [%]	0.12	0.06	0.09	0.81	2.74	3.06
Fluid_E [%]	0.18	0.09	0.12	3.22	5.52	5.22
Heat_1 [%]	0.20	0.08	0.12	3.22	5.52	5.18
Heat_2 [%]	0.18	0.09	0.13	3.25	5.57	5.27
Aux_E [%]	0.18	0.10	0.14	3.38	5.91	5.80

Given these results, SS was selected as the best preprocessing. Although it has not the lowest global error, the value does not vary significantly from the other normalizations. Moreover, it exhibited the lowest statistical values, indicating that it does not depend on the initialization of the node weights.

For the selected ANN model, Fig. 4 shows the comparison between the testing and predicted values from each output. For the ML model to demonstrate good predictive capability, the values should ideally be equal, meaning that when plotted they should lie on the identity line (ideal).

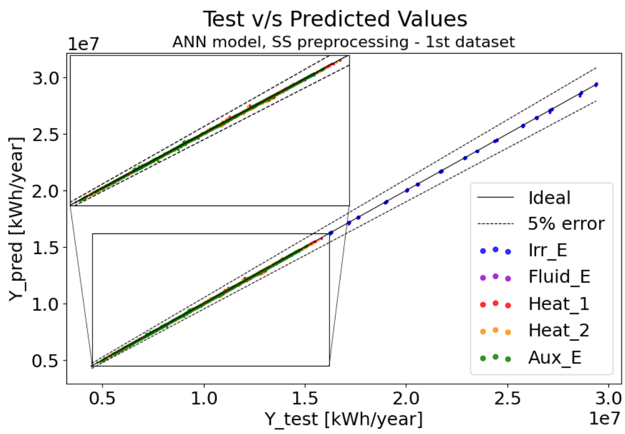

Fig. 4: Test v/s Predicted values obtained with ANN model traind with the first dataset

4.2 Decision Tree model selection for first dataset

For the DT models trained with the first dataset, in Tab. 4 are presented the values obtained for the statistical analysis for each preprocessing.

Tab. 4: Statistic data for DT models trained with the first dataset

Statistic	MMS	MAS	SS	PCA3	PCA2	Original
Mean train score [%]	0	0	0	0	1.85	0
SD train score [%]	0	0	0	0	0.006	0
Mean test score [%]	0.20	0.19	0.19	2.50	3.17	0.20
SD test score [%]	0.024	0.020	0.026	0.10	0.087	0.017

Complementing these values, Tab. 5 shows the errors obtained comparing the test and predicted values for each preprocessing.

Tab. 5: Testing errors for DT models trained with the first dataset

Error	MMS	MAS	SS	PCA3	PCA2	Original
Global [%]	0.18	0.19	0.17	2.43	3.13	0.19
Irr_E [%]	0.013	0.008	0.010	0.001	0.26	0.009
Fluid_E [%]	0.24	0.25	0.23	3.0	3.78	0.25
Heat_1 [%]	0.24	0.25	0.22	3.0	3.78	0.25
Heat_2 [%]	0.18	0.19	0.16	3.03	3.81	0.18
Aux_E [%]	0.24	0.25	0.21	3.03	4.04	0.24

Same as before, to select the best preprocessing, it is clear the best model is the one using Standard Scaler. For this model, in Fig. 5 is shown the comparison of the testing and predicted values for each output.

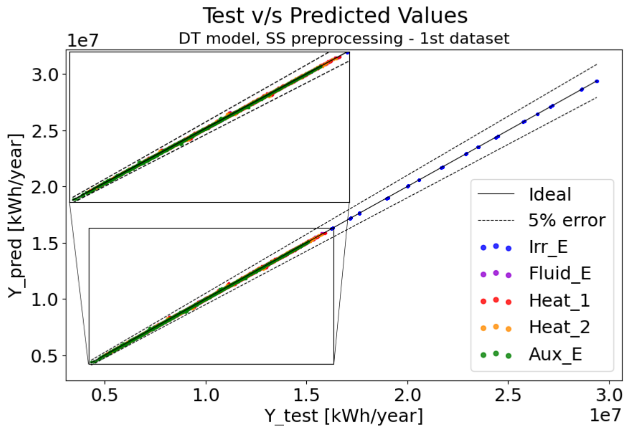

Fig. 5: Test v/s Predicted values obtained with DT model trained with the first dataset

4.3 Artificial Neural Network model selection for second dataset

On the other hand, for the models trained with the second dataset,

Tab. 6 shows the statistical values from the re-trainings done for each preprocessing. Subsequently, Tab. 7 shows the errors obtained comparing the test and predicted values for each output.

Tab. 6: Statistic data for ANN models trained with the second dataset

Statistic	MMS	MAS	SS	PCA3	PCA2	Original
Mean loss [%]	5.41	3.49	3.03	5.41	14.79	31.43
SD loss [%]	7.81	5.84	2.17	7.81	1.54	10.04
Mean val_loss [%]	4.48	3.56	3.05	16.14	15.20	31.41
SD val_loss [%]	8.02	5.81	2.39	8.02	1.80	10.02

Tab. 7: Testing errors for ANN models trained with the second dataset

Error	MMS	MAS	SS	PCA3	PCA2	Original
Global [%]	0.58	0.63	0.39	11.10	11.20	21.15
Irr_E [%]	0.14	0.31	0.18	3.86	3.87	2.49
Fluid_E [%]	0.14	0.18	0.13	5.38	4.92	1.12
Heat_1 [%]	0.13	0.16	0.12	5.41	4.88	1.10
Heat_2 [%]	0.13	0.13	0.12	5.57	5.03	1.04
Aux_E [%]	2.34	2.38	1.44	35.27	37.29	100

Analyzing the previous tables, once again it can be concluded that the best preprocessing is Standard Scaler. Given this, Fig. 6 shows the test and predicted values plotted to compare them.

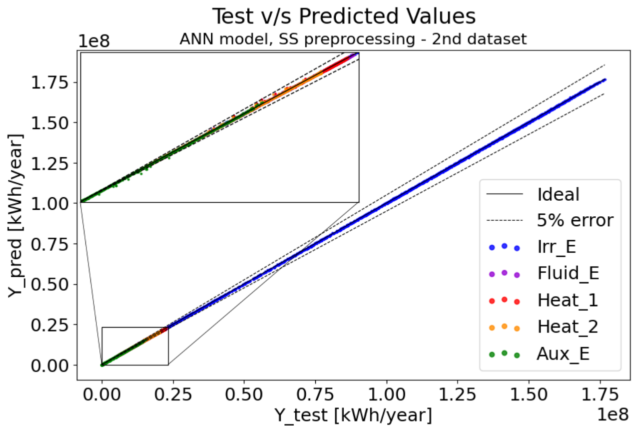

Fig. 6: Test v/s Predicted values obtained with ANN model trained with the second dataset

4.4 Decision Tree model selection for second dataset

Lastly for the selections of the ML models, Tab. 8 shows the values obtained from the statistical analysis done for the DT models trained with the second dataset, for each preprocessing. Also, in Tab. 9 are presented the errors from comparing the testing and predicted values.

Tab. 8: Statistic data for DT models trained with the second dataset

Statistic	MMS	MAS	SS	PCA3	PCA2	Original
Mean train score [%]	1.03	0.63	1.03	6.35	7.10	0.64
SD train score [%]	0.011	0.007	0.01	0.21	0.18	0.008
Mean test score [%]	3.54	3.36	3.58	12.32	13.82	3.63
SD test score [%]	0.30	0.40	0.30	2.37	1.94	0.25

Tab. 9: Testing errors for DT models trained with the second dataset

Error	MMS	MAS	SS	PCA3	PCA2	Original
Global [%]	3.61	3.61	3.50	12.80	14.48	3.62
Irr_E [%]	0.87	0.85	0.86	3.25	3.85	0.86
Fluid_E [%]	1.01	1.01	1.01	2.09	2.41	1.04
Heat_1 [%]	0.91	0.90	0.91	1.93	2.16	0.93
Heat_2 [%]	0.57	0.56	0.56	1.79	2.10	0.60
Aux_E [%]	14.71	14.72	14.17	54.92	61.90	14.70

In this case, once again the Standard Scaler preprocessing was selected as the best one, showing low statistics and the lowest errors. Taking this into account, in Fig. 7 is depicted the test v/s predicted values plot for each output.

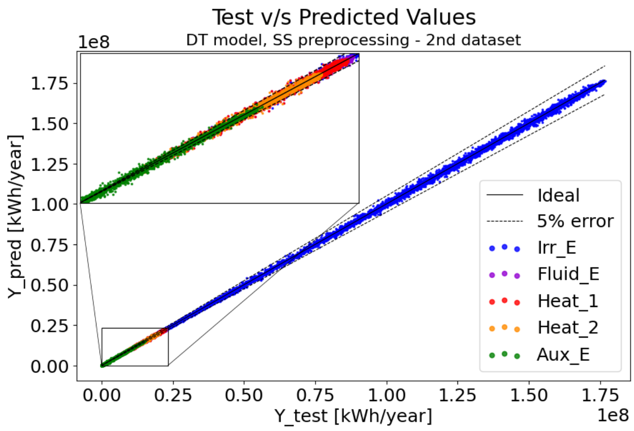

Fig. 7: Test v/s Predicted values obtained with DT model trained with the second dataset

4.5 One objective optimization

Analyzing all the results shown previously for the ML model selection, the ANN model using Standard Scaler preprocessing and trained with the first dataset was selected as the best model, based on predictive capability. Because of this, optimizations were made using this model, to ensure predicted energies that allow an effective optimization of the solar thermal plant. In Fig. 8 are shown the values of LCoH obtained for the one objective optimization, for each cost of fuel and solar collectors.

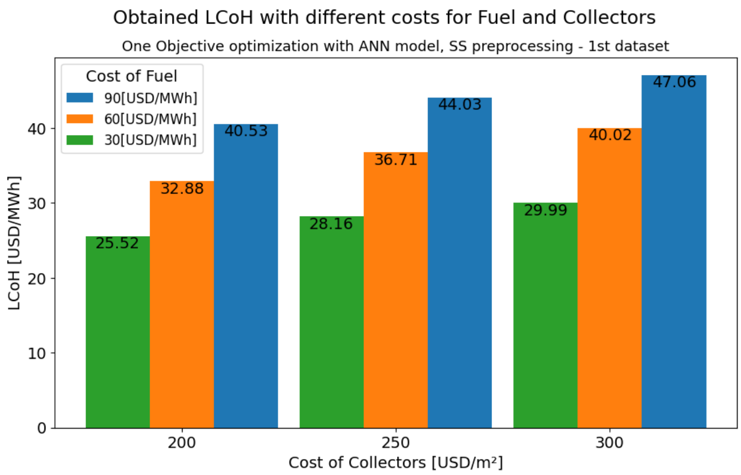

Fig. 8: Optimized LCoH from the one objective optimization, for each cost from the sensibility analysis

Subsequently, in Tab. 10 are presented the inputs obtained from the optimization, meaning these are the configurations of the solar thermal plant that minimizes the LCoH for each cost of the sensibility analysis.

Tab. 10: Configurations obtained from the one objective optimization

Costs Fuel-Collector	Col_A [m²]	Vol_T [m³]	B [°]	Mass_F [kg/hm²]	Eff_HX [-]
30-200	9768	510	34	68	0.9
30-250	6167	215	30	57	0.9
30-300	4870	208	29	75	0.8
60-200	9802	724	31	69	0.9

60-250	9884	741	28	40	0.9
60-300	9849	766	27	49	0.9
90-200	9917	766	29	41	0.9
90-250	9854	745	26	57	0.9
90-300	9959	816	26	71	0.9

Giving these results, Tab. 11 shows the comparison between the LCoH obtained from the optimizations and the calculated with the outputs of the new simulations using TRNSYS-18.

Tab. 11: Optimized and simulated LCoH for the one objective simulation

Costs Fuel-Collector	Optimized LCoH [USD/MWh]	Simulated LCoH [USD/MWh]	Error [%]
30-200	25.52	25.85	1.28
30-250	28.16	28.56	1.41
30-300	29.99	30.06	0.22
60-200	32.88	32.76	0.36
60-250	36.71	36.51	0.56
60-300	40.02	39.91	0.28
90-200	40.53	40.29	0.59
90-250	44.03	43.66	0.84
90-300	47.06	46.66	0.84

4.6 Multi-Objective Optimization

For the multi-objective optimization, the plant LCoH and the Aux_E were minimized to minimize the cost of energy while keeping the boiler emissions as low as possible. The results obtained from both objectives, respectively, are shown in Fig. 9 and Fig. 10, for every cost of fuel and solar collectors analyzed. It's important to note that to obtain each of these values, the 80/20 rule was considered based on the Pareto principle, from the Pareto front formed from the optimization, giving priority to the LCoH.

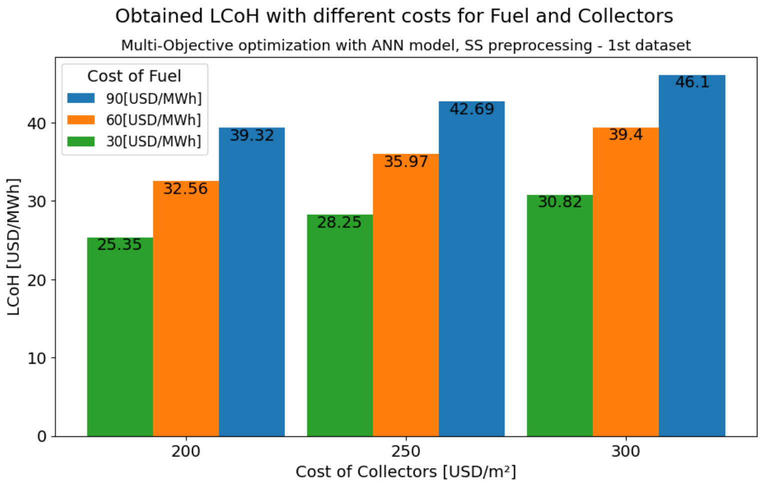

Fig. 9: Optimized LCoH from the two-objective optimization, for each cost from the sensibility analysis

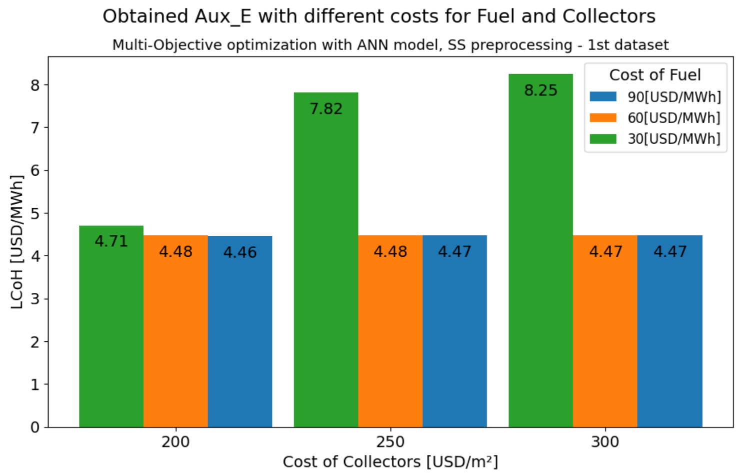

Fig. 10: Optimized Aux_E from the two-objective optimization, for each cost from the sensibility analysis

For each case, in

Tab. 12 are presented the inputs, the solar thermal plant configuration, obtained with the multi-objective optimization, each selected based on the Pareto front and 80/20 rule.

Tab. 12: Configurations obtained from the multi-objective optimization

Costs Fuel-Collector	Col_A [m²]	Vol_T [m³]	B [°]	Mass_F [kg/hm²]	Eff_HX [-]
30-200	9999	687.27	32	80	0.9
30-250	7910	454.82	28	80	0.9
30-300	7632	424.67	28	80	0.9
60-200	10000	818.41	31	80	0.9
60-250	10000	817.01	31	80	0.9
60-300	10000	821.59	31	80	0.9
90-200	10000	839.79	31	80	0.9
90-250	10000	832.56	31	80	0.9
90-300	10000	830.68	31	80	0.9

Lastly, in Tab. 13 and Tab. 14 are shown the optimized and simulated, with TRNSYS-18, LCoH and Aux_E with their respective percentage error.

Tab. 13: Optimized and simulated LCoH for the multi-objective simulation

Costs Fuel-Collector	Optimized LCoH [USD/MWh]	Simulated LCoH [USD/MWh]	Error [%]
30-200	25.35	25.26	0.35
30-250	28.25	28.36	0.41
30-300	30.82	30.94	0.39
60-200	32.56	32.50	0.19
60-250	35.97	35.91	0.18
60-300	39.40	39.34	0.16
90-200	39.32	39.22	0.26
90-250	42.69	42.60	0.20
90-300	46.10	46.01	0.21

Tab. 14: Optimized and simulated Aux_E for the multi-objective simulation

Costs Fuel-Collector	Optimized Aux_E [kWh/year]	Simulated Aux_E [kWh/year]	Error [%]
30-200	4,709,457.5	4,650,117.24	1.26
30-250	7,820,685.5	7,891,344.39	0.90
30-300	8,248,605.5	8,327,604.21	0.96
60-200	4,475,711.0	4,452,018.35	0.53
60-250	4,476,506.0	4,452,797.77	0.53
60-300	4,473,280.0	4,448,898.97	0.55
90-200	4,461,215.0	4,436,977.95	0.54
90-250	4,465,799.0	4,441,552.44	0.54
90-300	4,467,019.0	4,442,701.65	0.54

5. Discussion and Conclusions

Given the nature of the data, in Tab. 3, Tab. 5, Tab. 7 and Tab. 9 is possible to compare the training errors between preprocessing techniques for every ML model and dataset used, where is shown that is important to use a preprocessing technique to the input data to ensure a good training of the ML models. In this case, the best techniques are the normalizations, these being SS, MMS and MAS. Comparing the ML models, ANN demonstrated superior performances. In Tab. 3 it's shown that for the first dataset the model achieved low errors when using normalization preprocessings (<0.2%) and exhibited excellent predictive and interpolation capabilities. On the other hand, DT models have poorer generalization ability. When using the second dataset, both models showed higher errors, due to the increased heterogeneity of the data, struggling particularly with the prediction of the boiler energy, which significantly impacted overall accuracy, especially in the case of DT models. On the other hand, for the SS preprocessing using ANN trained with the first dataset, in Tab. 2 it can be seen that the standard deviation of the training errors is 0.03%, which indicates that for all the re-trainings of the model, this preprocessing helps the model to converge and gives it stability. This means that this preprocessing gives the ANN model independence for the HP initialization, being this the best indicative that this is the best model to perform the parameter optimization for the solar thermal plant.

Concerning the optimization, observing Tab. 11, Tab. 13 and Tab. 14 where are depicted the comparative errors between the results obtained from one and multi-objective optimization, it can be interpretated that the ANN model using the first dataset and SS preprocessing exhibited excellent consistency. This high level of accuracy demonstrates that the ANN was able not only to fit the training data effectively but also to interpolate accurately within the design space, even for configurations that were not explicitly included in the original dataset. Such interpolation capability is crucial in optimization tasks, as it ensures smooth and realistic transitions between design variables, enabling the discovery of new optimal configurations beyond the discrete training points. Finally, from a design perspective, the configurations shown in Tab. 10 and Tab. 12 often reached the upper limits of collector area within the dataset. This suggests a saturation effect where the current dataset boundaries constrained the search space, and further improvement in the LCoH could be achieved by extending the dataset. Nonetheless, the ANN model demonstrated strong generalization and stability, making it a reliable surrogate for complex simulations. Overall, the results confirm that the ANN trained with the first dataset and preprocessed with Standard Scaler provides the most robust and accurate framework for optimization in the studied system.

In conclusion, the study successfully developed and applied a Machine Learning based optimization framework for an industrial solar thermal plant. The results showed that the problem formulation and model design were appropriate, achieving very low prediction errors. In terms of optimization, both single- and multi-objective models successfully minimized the LCoH and boiler energy within the imposed constraints.

However, satisfactory results were only achieved when using well-trained ANN models with low prediction errors. In contrast, models trained with the second dataset, characterized by higher heterogeneity, produced large deviations, confirming that accurate optimization depends strongly on the precision of the ML model. In summary, the ANN trained with the first dataset and Standard Scaler preprocessing proved to be the most effective approach, capable of delivering consistent, low-error, and physically coherent optimization results for solar thermal plant applications.

6. Acknowledgments

The authors gratefully acknowledge Fraunhofer Chile Research for providing valuable guidance, technical resources, and support during the development of this study.

7. References

Blank, J. and Deb, K. 2020 “pymoo: Multi-objective optimization in python,” IEEE Access, vol. 8, pp. 89497–89509.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., y Varoquaux, G. 2013 “API design for machine learning software: experiences from the scikit-learn project,” en ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122.

Chollet, F. 2019 “Keras.” https://github.com/fchollet/keras.

Dallapiccola, M., Trentin, F., Barchi, G., Dipasquale, C., Fedrizzi, R., y Moser, D. 2020. “Machine learning driven optimization of a hybrid electrical and thermal system,” en Proceedings of the 37th European Photovoltaic Solar Energy Conference and Exhibition, Lisboa, Portugal, pp. 7–11.

Fadlallah, S. O., Anderson, T. N., y Nates, R. J. 2021. “Artificial neural network–particle swarm optimization (ann-pso) approach for behaviour prediction and structural optimization of lightweight sandwich composite heliostats,” Arabian Journal for Science and Engineering, vol. 46, no. 12, pp. 12721–12742.

Gad, A. 2021 “Pygad: An intuitive genetic algorithm python library” https://www.researchgate.net/publication/352373466_PyGAD_An_Intuitive_Genetic_Algorithm_Python_Library.

Kalogirou, S. 2006. “Artificial neural networks and genetic algorithms for the optimization of solar thermal systems,” Artificial Intelligence in Energy and Renewable Energy Systems. https://ktisis.cut.ac.cy/bitstream/10488/824/2/C82-Inv\%20ST7.pdf.

Lee, J., Hong, S., Cho, H., Lyu, B., Kim, M., Kim, J., y Moon, I. 2021. “Machine learningbased energy optimization for on-site smr hydrogen production,” Energy Conversion and Management, vol. 244, p. 114438.

Magnier, L. and Haghighat, F. 2010. “Multiobjective optimization of building design using trnsys simulations, genetic algorithm, and artificial neural network,” Building and Environment, vol. 45, no. 3, pp. 739–746.

Ministry of Energy, Chilean Government, “Explorador solar,” 2017, https://solar.minenergia.cl/inicio.

O’Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L. 2019. “Keras Tuner.” https://github.com/keras-team/keras-tuner.

Starke, A. R., Cardemil, J. M., Escobar, R., y Colle, S. 2018. “Multi-objective optimization of hybrid CSP+PV system using genetic algorithm,” Energy, vol. 147, pp. 490–503.

Weichert, D., Link, P., Stoll, A., Rüping, S., Ihlenfeldt, S., y Wrobel, S. 2019 “A review of machine learning for the optimization of production processes,” The International Journal of Advanced Manufacturing Technology, vol. 104, no. 5, pp. 1889–1902.