

**GLOBAL JOURNAL OF ADVANCED ENGINEERING TECHNOLOGIES AND
SCIENCES**
**MODELLING AND OPTIMIZING SHELL-AND-TUBE HEAT EXCHANGERS
WITH AIR INJECTION**

Manglesh Dubey, Khemraj Beragi
dubeymanglesh1998@gmail.com , khemrajberagi@gmail.com

DOI: <https://doi.org/10.29121/gjaets.2025.9.2>

ABSTRACT

This research paper is devoted to optimization of the shell-and-tube heat exchangers (STHEs) with air injection using machine learning tools to estimate thermohydraulic performance. Comparative study of four machine learning models, Random Vector Functional Link (RVFL), Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Random Forest (RF), to assess their usefulness in predicting key performance indicators of temperature of outlets and pressure drops were analysed. As put forth in the study, data preprocess, feature selection, and hyperparameter tuning are evident in enhancing the precision of such models. The obtained results reveal that RVFL is preferable with respect to predictive capabilities as it produced high correlation coefficients in predicting outlet temperature and pressure drops. The work proves that machine learning can have significant capabilities to optimize the design of heat exchangers, with the advantage of using data to guide the pursuit of energy efficiency and thermal performance of an industrial process.

KEYWORDS: Heat Exchangers, Machine Learning, Shell-And-Tube, Random Vector Functional Link, Support Vector Machine, K-Nearest Neighbors, Random Forest.

1. INTRODUCTION

In several industries, heat transfer between liquids may be necessary and heat exchangers are part of such systems. Shell-and-tube heat exchangers (STHEs) are common amongst the many kinds of heat exchangers; its popularity can be attributed to the fact that it is robust, flexible, and made capable to flow high amounts of fluid. The exchangers are widely used in the chemical process industries, where sensible thermal management is of paramount importance; as well as in the power industry (using high-performance exchangers) and the HVAC industries (where lesser-performance units are deployed). Nevertheless, with the use being so common, their performance may be restricted due to reasons including fouling, maldistribution, and energy wastage.

In a bid to increase performance of STHEs, recent research has concentrated on bettering the heat transfer efficiency with minimal pressure drop. A potential solution is to bring about air injection into the shell-side flow which has the potential of intensifying turbulence and stimulating improved heat transfer rates. Nevertheless, the complete effect of air injection on thermohydraulic performance on these exchangers is felt when modeling and optimization is carefully done.

The study will fill the research gap of no optimization of STHEs with the use of machine learning algorithms. In particular, it uses four of the latest machine learning algorithms: Random Vector Functional Link (RVFL), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest (RF) as the means to predict and optimize thermohydraulic behavior of the shell-and-tube heat exchanger under the different operating conditions. These machine learning models offer a powerful data-driven solution to the modeling of non-linear relationships between input parameters (e.g. flow rates and the temperature differences) and output parameters (e.g. outlet temperatures and pressure drops).

This is the main purpose of this study and it will be determined whether it is possible to use these machine learning models to precisely forecast STHEs performance and this will especially be aimed at determining the impacts of the air injection. This research work is also aimed at finding optimal operating regimes and design values that can achieve the best thermal efficiencies of the heat exchanger during operation and with minimal energy use as possible through data driven models. This study will also have implications on the general field of heat exchanger optimization because the findings of this research will help show how machine learning can be delivered to the design enhancements and predictive modelling.

The paper is organized in the following way: the literature review contains information about the latest studies concerning the optimization of heat exchangers and the work on machine learning in on-an-engine territory. The

section on methodology describes research methodology, i.e. the machine learning techniques and data preprocessing applied during the research. Performance of various models is described on the results and discussion part of the paper whereas the conclusion gives a summary of the result and recommendations of future investigation.

2. LITERATURE REVIEW

The study given in [1] explores the optimization of a Compound Parabolic Concentrator (CPC) in use with hybrid solar systems. Carrying out optical and thermal analysis using ANSYS, the paper has been comparing the Discrete Ordinate (DO) and Monte Carlo (MC) approaches to solve radiative transfer equations. MC technique was more precise and numerically effective in three-dimensional simulation. Also, when Maximum Power Point Tracking (MPPT) was included in the MATLAB, the management of mass flow rate was improved and this brought about a rise by 4% in power absorption and 12% rise in power output using evacuated tube collectors. However, such usage of computational models, as DO and MC, may create restrictions, as they are based on assumptions and more complex nature of real-life processes. In addition, the results of the study can be applied to the analyzed setup and cannot be easily applied in other system configuration or different environmental conditions.

In [2], the authors study the optimization of performance of fin-tube heat exchangers (FTHes) with the help of vortex generators (VGs) with the help of Response Surface Methodology (RSM) and artificial neural networks (ANN). This research is concerned with three VG design parameters, which include, length, arc angle and attack angle with the view to predict Nusselt number and friction factor. ANN was better in predictive accuracy than the RSM whereby R-squared value was higher in ANN. The used variables of the design may however limit the flexibility of the models at a larger application. Moreover, ANN will be more accurate but more computing resources are demanded and this may be a drawback to high design optimizations.

The application of the supervised machine learning algorithms using air-injected, transverse baffled shell and tube heat exchangers is identified in the paper [3] regarding the prediction of thermohydraulic performance. Four machine learning algorithms, which are Random Vector Functional Link (RVFL), Support Vector Machine (SVM), Social Media Optimization (SMO), and k-Nearest Neighbors (k-NN) are tested. RVFL became the best model having reduced root mean square error (RMSE) and mean relative error (MRE). Even though RVFL demonstrated good performance, the limitations of the applicability of the findings in this study to other types of heat exchangers can be the focus of the study on a particular design. Moreover, the efficiency of the machine learning models used in a more complex or diverse settings of heat exchangers is yet to be proved.

The test that occurs in [4] is using deep learning modes to have an accurate power prediction of turbines in real time, especially using Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). The paper uses a large dataset of a power plant to compare these deep learning models with some other machine learning solutions. Compared to the other models, the accuracy and performance of RNN shown a lesser relative error of lower than 1% of most of the predictions made. The use of a certain dataset, however, complicates the applicability of the models to various situations in which it is operational and might also be unable to cater to real-life experiences due to overfitting in practice.

The thermal performance of heat exchanger fin in an axial finned-tube heat exchanger through k-Nearest Neighbor (k-NN) algorithm is examined by the authors of the article [5]. The paper considers the impacts of the spacing between the fins, their thickness, the material used and convective heat transfer coefficient on performance parameters. The k-NN algorithm was quite accurate when predicting, with the current results possibly limited to the scopes of materials and geometries covered. It is difficult to conclude that the study results can be generalizable to the other fin configurations or other types of material.

The study by [6] looks into ways of modeling the rates of heat transfer in shell and tube heat exchangers using Support Vector Machine (SVM) techniques. SVM was predicted more accurately compared with Artificial Neural Networks (ANN) and the several old tried and tested methods. Nevertheless, the fact that the study bases itself on certain experimental evidence and has few points of reference to alternative approaches can make the overall application of the results limited. It may happen that the SVM is not as strong in predicting the performance of the heat exchanger in case it is projected to other or more complex heat exchangers design.

The authors of [7] created a deep learning architecture based on ensemble neural networks to forecast fouling resistance on cross-flow heat exchangers that implement waste heat recovery. The model was shown to be very

accurate as the mean absolute prediction errors (MAPE) were comparable to 10^{-4} KW-1 and the coefficients of determination (R^2) were more than 99%. Even though the model works, there is a chance that some modification to the network structure as well as the input characteristics of the model will have to be adjusted when applied to other types of heat exchangers. The fact that the study was done on a particular category of heat exchanger would reduce the generalizability of study.

Lastly, [8] targets at designing the thermal and hydraulic effectiveness of a water-to-water chevron kind of plate heat exchanger with an objective-based genetic algorithm in MATLAB. The paper reports a Pareto Front of pressure drop versus heat transfer trade off. It was found that despite the fact that genetic algorithm gave a set of optimal solutions, there exists trade-off between pressure drop and heat transfer because no solutions can give optimal values of a combination of the two objectives. The sensitivity analysis allowed showing how geometrical parameters influenced the research results, however, it also revealed that there could be no single solution to the problem of balancing efficiency and operation costs.

The authors of [9] explore optimization of the heat exchanger, a two-dimensional cross flow tube bank heat exchanger, in the spirit of maximizing heat transfer effectiveness and minimizing the loss pressure. They have built a Pareto front using the Non-dominated Sorting Genetic Algorithm II (NSGA-II) and numerous CFD simulations thus having a trade-off of the two objectives. The research showed that it was not entirely a good idea to introduce constraints to minimize the search space since it added a superfluous complexity and compromised the optimization procedure. Rather, the search domain was broader and produced better results. The assumption of laminar flow set-up used in the study is one of its major limitations since it does not represent a complete description of heat exchanger behaviour under turbulent conditions. It also incurred high cost in computation, as it involved substantial number of CFD evaluation thus it might be difficult to apply the method practically. The state of the results on a certain geometric layout can also limit the findings of a certain correspondence to other geometrical layouts of heat exchanger.

The concept of a neural network-based per-PINN architecture proposed by authors of [10] was used to predict the time-varying heat transfer coefficient in an operating heat exchanger using closed-loop operational information. This method tackles the difficulties in the process of fouling and effectively offers precise estimations without bringing in outside stimulations or the parameters of the controllers. Per-PINN demonstrated better performance in inverse parameter learning compared to the available Physics-Informed Neural Networks (PINNs) and, therefore, an appealing solution to real-time monitoring and optimization of heat exchanger performance.

The authors of [11] have used condition their heat transfer coefficient in a micro-channel system by using a multi-output Gaussian Process Regression (GPR) model. High-fidelity experimental data were used to train the model and it showed data efficiency, reduced hyperparameters and automatic trade-off between data fit and model complexity. The approach has resolved the limitations to conventional machine learning including overfitting and uncertainty estimation. The approach is a good solid medium that can be used to design and operate compact heat exchangers.

Machine learning algorithms (Polynomial Regression (PR), Random Forest (RF) and Artificial Neural Networks (ANN)) have been employed by the authors of [12] with the purpose to model thermohydraulic correlations of heat exchangers with twisted tape inserts. It was shown that ANN offered better predictions of heat transfer coefficients than PR and RF and thus there is a potential of machine learning to improve the design and heat exchanger performance prediction using a passive flow augmentation method.

The authors of [13] introduced DualMS, a computation framework optimizing dual-channel minimal surfaces in the context of heat exchanger geometry design. In contrast to conventional triply periodic minimal surfaces (TPMS) DualMS has a higher degree of accommodating topological modifications of the surfaces and provides a better thermal performance owing to reduced pressure drops. It employs neural network to identify flow skeletons and identify surface boundaries automatically to consequentially design more efficient and flexible heat exchangers.

Research Gap: Although important progress has been achieved regarding the application of theoretical procedures of artificial intelligence (AI) in the foreseeing and optimization of the performance of shell-and-tube heat exchangers, especially by using certain artificial neural networks (ANNs) successfully, there are some significant (or rather noticeable) gaps. The previous studies have mostly revolved around the different parameters

of design like the type of baffling, cuts and porosities amongst others, there has also been a high degree of usage of ANN to facilitate the multi objective optimization as well as sensitivity analysis. Nevertheless, the literature lacks the research on the issue of integrating AI models in predicting and optimization in heat exchangers with transverse baffles in particular. The given research gap is of special concern since the interaction between thermal and hydraulic parameters of such systems is rather complex and has not been studied widely. Through this, the proposed research will produce a hybrid photovoltaic/thermal (PV/T) system together with a loop thermosyphon to extract solar PV panel heat effectively. Moreover, the research work will concern designing and modeling of heat exchangers by means of machine learning and soft computing methods. This entails the increase of predictability of air injection impacts on thermohydraulic responses of shell-and-tube heat exchangers, as well as general accuracy and efficiency of design. These are the particular areas that are to be addressed by the research in order to realize the lack of such applications of AI to the complex heat exchanger layout and improve the process of heat transfer using sophisticated algorithms of machine learning.

3. PROPOSED METHODOLOGY

The current section outlines the methodology that combines machine learning and optimization procedure to optimize shell-and-tube heat exchangers designing and predicting the performance. The data-driven models offer a superior, flexible, and an efficient tool on the heat exchanger design, which has a greater benefit (in terms of performance and optimization) as compared to the conventional methods. This study enhances the knowledge in thermal systems since it will be possible to predict heat exchanger performance by favorable and unfavorable design parameters and operating conditions. Figure 1 provides an overview of the consecutive steps of the proposed methodology and explains the necessary modifications in the development and optimization of the PV/T system to the implementation of the modern machine learning to allow designing and predicting the performance of the heat exchanger.

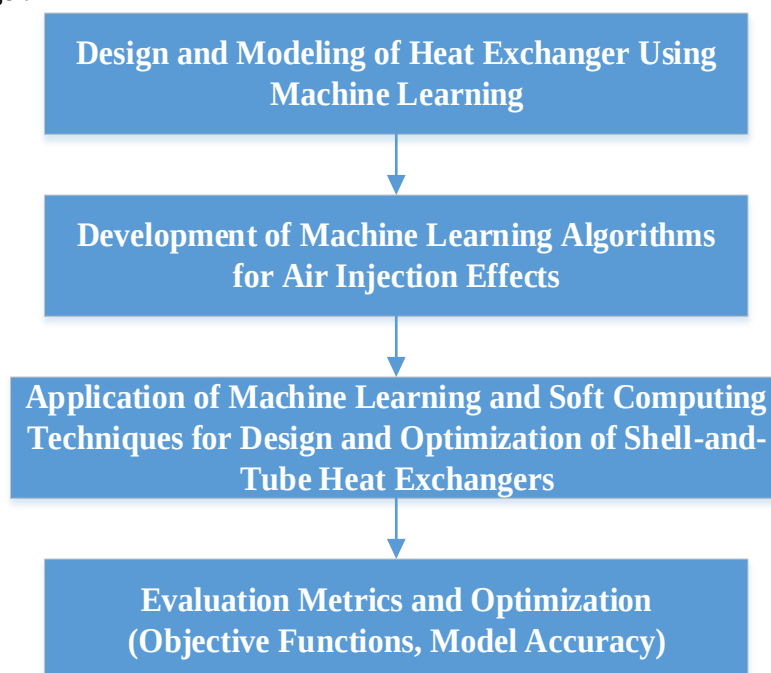


Figure 1: Flow Diagram for Optimizing Heat Exchanger Performance Using Machine Learning Techniques

3.1 Data Collection and Preprocessing

Data collection and preprocessing processes present a key step towards the acquisition of quality, consistent and meaningful data which becomes available to the machine learning models. Training, validating, and testing of machine learning model relies on the experimental data. The quality of information obtained is critical and plays a central role regarding the results of the predictions made and general performance of the heat exchanger models. This is the part where we explain in details the steps of the collection and preprocessing of data employed in this study.

3.1.1 Experimental Data Collection

The key variables obtained in primary data include:

- **Cold Fluid Flow Rate (Q_{MC}):** The volumetric flow rate of the cold fluid passing through heat exchanger is known as cold fluid flow rate (Q_{MC}) and is measured in liters per minute (LPM). One of the main design features which affect the rate of heat transfer and the pressure drop down the system is the flow rate.
- **Hot Fluid Flow Rate (Q_{AI}):** The same thing as the cold fluid flow rate but it is the volumetric splitting the hot fluid passing into the heat exchanger and still in LPM. The rate of flow of the hot fluid has a direct influence to the thermal performance of the heat exchanger because it will change the rate of the heat exchange between the two fluids.
- **Cold Outlet Temperature (T_{cout}):** It is the outflow temperature of the cold fluid at the end of the heat exchanger and it is commonly measured in degrees Celsius ($^{\circ}\text{C}$). The variable is an important output because it depicts the effectiveness of the heat exchanger to absorb heat by the hot fluid. The lower the outlet of the cold the better the performance level of heat transfer.
- **Hot Outlet Temperature (T_{hout}):** The temperature of the hot fluid at outlet in degree Celsius ($^{\circ}\text{C}$). The temperature recorded at the hot outlet gives an indication on how well the cold fluid takes in heat as transferred by the hot fluid. It is one of the most important criteria to determine the work of the heat exchanger.
- **Pressure Drop across Shell (ΔP_{shell}):** It is the pressure drop that an exchanger has on the shell side in a heat exchanger, which is recorded in bar. The drop in pressure is one of issues that greatly contribute to the amount of energy utilized by the heat exchanger system since it determines the pumping power needed to pass the fluid.

3.1.2 Data Preprocessing

3.1.2.1 Cleaning of the Data

Cleaning of the experimental data was done by the following ways:

1. **Handling Missing Value:** The omitted values are also a typical issue of the experiment in the gathering of information. As an example, gaps in the data of particular parameters may be achieved due to the measurement errors or malfunctioning of the sensors. It is necessary to tackle these missing values to avoid loss of data and biasness in the models. They are two popular strategies:
 - **Imputation:** In a situation when it is missing, the missing values are estimated, using the existing data by means of imputation. The two most used types of imputation include:
 - **Mean Imputation:** The missing values are filled up by the mean of the non-missing information of that feature. This is the easy type of a method, which is appropriately applied in case of moderately normal distribution of data.

$$X_{\text{imputed}} = \mu_x \quad (1)$$
 Where the mean of the feature called X over all possible values is given by the term μ_x .
 - **Regression Imputation:** In the event that the missing data has a foreseeable pattern with other characteristics, the regression imputation is applied. This includes computation of the missing value based on a regression model that is developed based on the existing data. The advantage of this technique is that it is applicable in a situation where the missing variable has been related to the other characteristics.
 - **Deletion:** Where the proportion of missing values is small, and the deletion of this missing data does not greatly decrease the quality of the data set, the removal of the missing values row could be made. This method is mainly adopted where missing data is not biased and random.
2. **Outliers Detection and Dealing:** The outliers have the capability to massively distort the output of the machine learning models by altering the means, the variance, and the distribution. The most representative data was learned on the model as statistical techniques were used to find outliers in the dataset.
 - **Z-Score Computation:** Z-score is a method of identifying outliers, where observation which is extremely distant to mean is considered an outlier. When the value of Z-score is above 3 or below -3, then an outlier occurs.

$$Z = \frac{x - \mu}{\sigma} \quad (2)$$

Where:

- X is the individual data point,

- μ is the mean of the feature,
- σ is the standard deviation of the feature.
- **IQR Method (Interquartile Range):** The IQR method determines the outliers by determining those values which are not included within the range, next to the lowest and the highest quartile Q_1 and Q_3 . The formulation for IQR is given by:

$$IQR = Q_3 - Q_1 \quad (3)$$

Any value smaller than $Q_1 - 1.5 \times IQR$ or any value that are greater than $Q_3 + 1.5 \times IQR$ are regarded as outliers. These radical values were deleted or rectified.

3. **Removing Duplicates:** Duplicate entries happen when the same point enters the data base twice or more times either as a result of fault in data gathering or malfeasance in the system. To prevent the presence of redundancies in the information, duplicate rows were identified and eliminated, which could distort the results and over-emphasize some findings. Duplicates were detected through examination of the rows in which all the values of the features were the same. Then these rows were cancelled out to have the single data points.

3.1.2.2 Data Normalization

The Z-score normalization carries out the data in such a manner that the mean and the standard deviation of the data will become one and zero respectively. The normalization formula of Z-scores is the following:

$$X_{norm} = \frac{X - \mu}{\sigma} \quad (4)$$

Where:

- X is the raw data point,
- μ is the mean of the feature,
- σ is the standard deviation of the feature.

Through the Z-score normalization of all features, every feature is normalized to equal standard normal distribution so that no single feature dominates the model and causing differences in scales issues. As an example, the flow rates might vary in hundreds, and temperatures can vary in a range of 30-80°C, therefore, by normalizing such features, the model training can be balanced.

Since such machine learning algorithms like Support Vector Machines (SVM) and K-Nearest Neighbor (KNN) are used to process the data, the normalization process promises better performance and the ability to quickly converge as the size of the features is not considered.

3.1.2.3 Features Selection

In the given study, the feature selection is carried out according to the given techniques:

- **Correlation Analysis:** Correlation analysis is also used to determine the linear relationship relationships among input features and output variables. A feature whose relationship with the target variable is strong will be retained and the poorly related one will be discarded. Each of the feature-target pairs undergo the calculation of the Pearson correlation coefficient:

$$Correlation(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2 \sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (5)$$

Where:

- X and Y are the input features and the target variable,
- $\bar{X}$ and $\bar{Y}$ are the means of X and Y , respectively.

When the value of correlation coefficient is near to 1 or -1 the feature is said to have high relationship with the target and is retained to go further into analysis. On the other hand, those features whose correlation values are low are eliminated.

- **Variance Thresholding:** Some features have variance that is relatively weak and not significant to the model. All these characteristics are eliminated in order to increase the output of the model. A feature with near zero variance contains no or minimal variability and can hardly get used in separating various data points. Variance of a feature is given by:

$$Variance(X) = \frac{1}{n} \sum_{i=1}^n (X_i - \mu_X)^2 \quad (6)$$

Where X gives the data on features, μ_X is the mean of the feature and n is the number of data points. The features whose variance is less than the established threshold are removed.

- **Recursive Feature Elimination (RFE):** It is the wrapper technique and it picks the optimal group of features by iteratively fitting the model and deleting least important features according to the performance indication. The model is trained severally deleting one feature each time, and the features of least importance are located and dropped. This technique allows eliminating the number of features, assuring that only the most important ones will be included in the final model.

3.1.2.4 Data Splitting

After data cleaning, normalization and selection of the most important features is complete, the data is split in training and testing sets. This division is meant to enable the model to learn based on a part of the data then be tested using the other part of the data. It is usually a standard method that 80% of the data is used in training and 20% in testing.

K-fold cross validation is further adopted in order to ensure that fair assessment is done to the models. In k-fold cross-validation the data is divided into k subsets (folds) which are formed randomly and are equal in size. Model training is done on the $k - 1$ folds and the remaining k^{th} folds used in testing the model. This is repeated k times and each fold is a test set once. An average of the outcomes in each fold is taken to give more accurate determination of the performance of the model. This will reduce the biasness and all the data will be utilized when training and testing.

3.2 Model Selection and Definition

Model selection process plays an important role in predictive analytics especially the studies involving complex systems like shell-and-tube heat exchangers (STHEs). Four machine learning models were chosen in the research due to their capability of modeling non-linear relationship and non-linearity utilization of data at high dimensions. These models are developed in order to fore-see the behavior of heat exchangers especially at outlet temperature (cold and hot) and pressure drop according to some of the following parameter attributes like flow rates, temperature differences, and other competing parameters during operations of the heat exchanger.

Such machine learning algorithms as Random Vector Functional Link (RVFL) or a Support Vector Machine (SVM), K-Nearest Neighbor algorithms (KNN), Random Forest (RF) have been selected to be a part of this research. Each of the discussed models is discussed in successive sections on the basis of which knowledge about their mathematical formulations, strengths, and claims with regard to their use in the frameworks of this research could be acquired.

3.2.1 Machine Learning Models

Each machine learning model to be employed in this work is explained in the following sections in the following order Random Vector Functional Link (RVFL), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest (RF). Every of the subsections elaborates the theory of the model, mathematical formulation, strengths of the model, and applicability of the model to the given problem of heat exchanger performance prediction.

3.2.1.1 Random Vector Functional Link (RVFL)

Random Vector Functional Link (RVFL) is another kind of a neural network that has drawn some interest due to simplicity and capacity to capture the non-linear and complex relationships. As contrasted to the classical neural networks, RVFL operates on the principle of the weights randomization of the internal layer and is computer-efficient. The parameters of the hidden layer are set randomly and is not updated in training. The model is dedicated to determining the output weights solving the task of the least-squares optimization.

Working Principle of RVFL:

- **Random Initialization:** Though the weights of the hidden layer a_j in RVFL system are randomly initialized, the biases β_j are also random in nature. The most significant feature of RVFL is the fact that the parameters of the hidden layers are not trained in any way and this makes computation very simple. The data range within which these weights lie is $[-S, S]$ where S is a scaling factor determined as per the nature of data.

$$a_j = -S + 2S \cdot \text{rand}(d, h), \quad \beta_j = S \cdot \text{rand}(1, h) \quad (7)$$

Here:

- a_j is the weights between the first layer and the second layer,
- Bias of j is denoted by β_j .

- d is the number of features used to make the inputs,
- h is the hidden neurons.
- **Hidden Layer Activation:** The neurons of the hidden layer would be activated according to computed activation function. In RVFL a Triangular Basis Factor (tribas) is employed. The result of this function is a piece-wise-linear output which is computationally straightforward and which is good at capturing non-linearities. The formula of the activation is:

$$H_j = \max(1 - |X \cdot a_j + \beta_j|, 0) \quad (8)$$

Non-linearity can be done differently by the use of sigmoid:

$$H_j = \frac{1}{1 + \exp(-(X \cdot a_j + \beta_j))} \quad (9)$$

In this case, X is an input data matrix and the activation function brings non-linear relationship between the input and output.

- **Output Weights Computation:** With the calculation of the outputs of the hidden layer H , the weights of the output layer W using the Moore-Penrose Pseudoinverse is computed. The approach reduces the minimum squares error between the estimated and the actual outputs. The weight vectors of the output W is determined as:

$$W = (F^T F)^{-1} F^T Y \quad (10)$$

Where $F = [X, H]$ is the augmented input matrix, and Y is the matrix of target values (the output variables). The matrix $(F^T F)^{-1} F^T$ is the Moore-Penrose pseudoinverse.

Where $F = [X, H]$ is the extended input matrix and Y is a target value (output variables) matrix. The Moore-Penrose pseudoinverse is the matrix $(F^T F)^{-1} F^T$.

- **Prediction:** The output $\hat{Y}$ to be expected as an output to a particular input X is:

$$\hat{Y} = F \cdot W \quad (11)$$

This generates the expected outputs values that will be used in comparing with the actual target values in the evaluation of the model.

3.2.1.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is undoubtedly one of the most potent and popular machine learning models, which is particularly applicable with classification and regression activities. The goal of SVM is to determine the optimal hyperplane that achieves the largest margin between different classes in case of the role of the classifier or best fit the data in the case of regression. The version of the SVM applied to prediction of continuous outcome is Support Vector Regression (SVR). The idea of working of SVM is as follows:

- **Kernel Trick:** Among the main characteristics of SVM, one may mention the application of the kernel trick that provides the mapping of the input features to a higher-dimensional space where a linear separation is achievable. In SVR, the radial basis function (RBF) kernel is by far the most popular kernel, as follows:

$$K(x, x') = \exp(-\gamma |x - x'|^2) \quad (12)$$

Where:

- γ is a positive constant that regulates the impact of each sample in training.
- Data points in input are x and x' .

SVM has the kernel feature that allows it to represent non-linear as well as complex relations amid the features of the input and the target variable.

- **Optimization Problem:** SVM regression is aimed at locating a deviation-minimizing function, the difference between which and the true values of the targets would not exceed a predetermined value of ϵ , and at the same time should be a smooth one. SVM regression optimization problem is formulated as following minimization:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \epsilon_i \quad (13)$$

Where:

- The vector weight is w .
- C is a regularization coefficient which sets their trade-off between the margin and the error value.
- ϵ_i are the slack variables, they permit deviations in the ideal regression function.



- **Prediction:** At a new test point x the predicted value is:

$$f(x) = \sum_{i=1}^n \alpha_i K(x_i, x) + b \quad (14)$$

Where:

- Lagrange multipliers associated with the support vectors are known as α_i .
- b is the term for bias.

3.2.1.3 K-Nearest Neighbors (KNN)

One of the easiest and also useful machine learning algorithms is k-nearest neighbors (KNN). It is done with reference to the fact that like data values will have like outputs. KNN is performed by computing the difference between the test sample and all the training samples, and choosing the k nearest neighbors, an event that is followed by calculating the target value by using the majority classes rule or the average of the k -nearest training samples. KNN working Principle:

- **Distance Metric:** Euclidean distance is normally used to calculate the difference between data points. Defining two data points to be x and x' we obtain the Euclidean distance as:

$$d(x, x') = \sum_{i=1}^n (x_i - x'_i)^2 \quad (15)$$

The values of features of the data points x and x' are denoted by identifiers x_i and x'_i , respectively, and the value of n is the number of features.

- **Selecting Neighbors:** Once distance are calculated, then the algorithm picks the k nearest neighbors to the test point. The cross-validation would normally be used to select a value of k .
- **Prediction:** The test sample in regression is predicted as a mean of target values of the k -nearest neighbors:

$$\hat{y}(x) = \frac{1}{k} \sum_{i=1}^k y_i \quad (16)$$

Where, y_i is the intended value of the nearest neighbor i .

3.2.1.4 Random Forest (RF)

Random forest (RF) is an ensemble method which constructs a number of decision trees and combines their prediction to make a final decision. It uses the weights of the results of individual decision trees to enhance more accuracy and helps to overcome overfitting. Random forest is also mainly applicable in the case of regression problems dealing with large sets of data. RF Working Principle is as below:

Bootstrapping: RF employs bootstrapping, in which at each decision tree a random subset of the training data set is employed to train it. Such a replacement sampling technique also makes each of the trees gain access to a dissimilar number of data sets.

Building Decision Trees: Every decision tree is generated by recursively partitioning the data into subsets by choosing the feature that optimally partitions the data with respect to some pre-selected measure of impurity (e.g. Gini impurity in classification or mean squared error in regression). Gini impurity of a decision node t is given as:

$$G(t) = 1 - \sum_{i=1}^k p_i^2 \quad (17)$$

When p_i is the weight of samples of class i at node t and k is the number of classes.

Prediction: In regression, predictions made by all the trees are averaged to come up with prediction of test sample:

$$\hat{y}(x) = \frac{1}{N} \sum_{i=1}^N f_i(x) \quad (18)$$

Where, $f_i(x)$ is the prediction of the i^{th} tree and the total number of trees in the forest is denoted by N

3.3 Model Training and Prediction

Machine learning is based on training and prediction processes in building successful predictive models. The process of these steps entails feeding the model with data so that it can learn the relationship that exists between the input or features as well as target or outputs and then utilizing this knowledge to make predictions on new data. Within this study, the aim is achieved as the performance of shell-and-tube heat exchangers (STHEs) is to be determined as different operating conditions are reached. Examples of input parameters include the flow rates, temperature conditions among other operating parameters which affect the output variables which in this case are the outlet temperatures, and the pressure drop. This segment gives an even and thorough examination of the whole method of training models, after which there is the prediction phase in which trained models are tested and their accuracy is estimated. The described processes concentrate on the four machine learning models, which are applied in this study: Random Vector Functional Link (RVFL), Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Random Forest (RF).

3.3.1 Training the Models

To train a machine learning model, historical data will be supplied to it such that it learns the relationship between input features and output variables. The idea of the training process is to optimize the internal parameters (weights, biases, and others) of the model and achieve the ability to generalize the model on unseen data. The training phase is one of the aims of the study, which will empower the models to estimate key performance indicators (KPIs) like outlet temperatures and the pressure drops of the STHEs, given a list of input features (flow rates and temperature differentials).

3.3.1.1 Training Set Selection and Data Splitting

The initial activity of the training phase is to divide the available data into two parts that include training set and test set. It is this division that enables the model to be trained using part of the data and the rest of the data can be used to discover how well the model performs. In the current study, the data will be equally divided whereby 16 experiments will be taken as the training data and 4 experiments will be reserved as the test set.

This separation is essential to prevent overfitting, which is a typical issue in machine learning on which a model may work well on known data but it does not work well with novel data. The test set can be used to ensure that the model can predict accurately on unobserved data which is the same scenario that will occur in the real world where the model will be applied in actual practice.

3.3.1.2 Data Preprocessing: Normalization, and Scaling

Prior to the training stage, one should preprocess input data so that not to waste time on achieving the best performance of these models. Normalization is the first preprocessing that will be done to the data. More precisely, Z-score normalization is applied, and all the input features are normalized so that they have the average of 0 and standard deviation of 1. This matters since there are models which are sensitive with regard to the scale of the input feature such as Support Vector Machine (SVM) and K-Nearest Neighbor (KNN).

The formula to normalize is the Z- score:

$$X_{norm} = \frac{X - \mu}{\sigma} \quad (19)$$

Where:

- X is the original data point,
- μ is the mean of the feature,
- σ is the standard deviation of the feature.

The training process is made more efficient since turning every feature into a standardized scale, and the models will not be inclined to provide disproportionate weight to variables that have higher numeric ranges.

Besides normalization, the other kinds of preprocessing steps can include the management of missing values (in case there are any), outlier detection and elimination, and a proper representation of categorical values (e.g., one-hot encoding). But in this research, the most awaited consideration on is the numerical input features, which are normalized in person by means of Z-score method.



3.3.1.3 Model Hyperparameter Tuning and Configuration

All the four machine learning models applied in this research experiment have to be configured and hyperparameterized accordingly to be able to work best. These hyperparameters regulate the learning process such as model complexity, learning rate and regularization among others. The proper choice of these parameters is necessary to make of accurate predictions.

The hyperparameters that are used in each model are as follows:

- **RVFL:** h (number of hidden nodes), S (the scaling factor) and the type of activation function tribas or sigmoid are fixed beforehand. The number of hidden nodes is important to define how well the model can be used to capture non-linear relationship.
- **SVM:** Kernel function is selected, the box constraint C and the kernel parameter γ are selected. These parameters regulate the trade of bias and variance and influence generalization skills of the model.
- **KNN:** Number of neighbors k is fixed, it ascertains the number of neighbors the model will take into account during a prediction. Having a bigger k will allow one to make predictions easier; however, excessive k can smooth out critical patterns.
- **RF:** N is the number of trees, and the minimal size of leaves (that determines the maximum depth of a tree) are specified. It is done by using number of trees which enhances the model accuracy as variance is less since the predictions of different trees are averaged.

3.3.1.4 Training Algorithm Execution

After preprocessing and configuring all the hyperparameters is done, the process of training can start. In this stage the models are trained on the training data in order to learn the association between the features and the targets. The way each of them learns is explained below:

- **RVFL Training:** The RVFL model initializes the weights in the hidden layer randomly and only the output weights get learned using the Moore-Penrose pseudoinverse. With the model, it calculates the scores of the activation of the hidden layers based on input features after which the least squares optimization problem is solved in order to obtain the optimal output weights. This enables RVFL to capture nonlinear relationships which are complex but computationally efficient.
- **SVM Training:** SVM is projected by a measuring function, which blows the data to higher dimension. It then optimizes the quadratic form on the given data in order to come up with the best hyperplane which separates the data points. In case of regression, model tries to minimize the margin error and checks overfitting by means of the regularization parameter C .
- **KNN Training:** KNN need not be trained as in the typical sense since it maintains the complete training set. In its stage of prediction, it determines the distance between a test point and each training sample, choosing the k -nearest neighbors and predicting the result of them according to the target value they have.
- **RF Training:** Random Forest applies bootstrapping to train a number of decision trees using dissimilar pieces of the data. The trees are built through the recursive partitioning of the data with reference to feature values that result in the lowest impurities values (Gini impurity in case of classification and Mean Squared Error in case of regression). This last result is a mean of predictions of all trees.

3.3.2 Model Prediction

1. Input Features for Prediction

The test set should be provided in order to do predictions, and it is passed through each of the trained models (this is preprocessed data, i.e., normalized). The test data points are real-life situation in which the models are supposed to give the output variables using new input conditions. The model produces an output at every test data point and this value can be seen as the value of the target variable (outlet temperature or pressure drop) predicted by the model.

2. Prediction Process for Each Model

Every machine learning model has a different prediction mechanism as follows:

- **RVFL Prediction:** During prediction, once RVFL model is trained, the hidden layer activation of the test input is calculated and the values are multiplied with the weights of the output that are acquired during the training process. The last prediction is as follows:

$$\hat{Y} = F \cdot W \quad (20)$$

Here, F is the augmented matrix of input feature and hidden layer activation and W symbolizes the matrix of output weights.



- **SVM Prediction:** Kernel function is used to map the test data into the space of features that are higher-dimensional. The dot product of the test data with the support vectors is calculated and the bias term is added giving the predicted output:

$$f(x) = \sum_{i=1}^n \alpha_i K(x_i, x) + b \quad (21)$$

Where α_i are the support vector weights, and b symbolizes the bias term.

- **KNN Prediction:** In KNN, prediction is made with relation to the k nearest neighbors. The same is available in the model that calculates the Euclidean distance among the test point and every single training point, picks the closest neighbors, and averages their target:

$$\hat{y}(x) = \frac{1}{k} \sum_{i=1}^k y_i \quad (22)$$

- **RF Prediction:** In Random forest, prediction is computed as a mean of predictions made by all separate decision trees. The output based on each tree is equally weighted and the result as:

$$\hat{y}(x) = \frac{1}{N} \sum_{i=1}^N f_i(x) \quad (23)$$

Where N is the total number of trees in the forest, and i^{th} tree prediction is shown by $f_i(x)$.

3.4 Model Evaluation

This section describes the measures of evaluation that are applied to ascertain the performance of the models and how the cross-validation method was adopted to ascertaining how the models are satisfactory and how they are robust.

3.4.1 Evaluation Metrics

Evaluation metrics are important core aspects in determining the effectiveness of a model. In tasks involving regression such as the case with the current study which aims at predicting continuous measures of outlet temperatures and pressure drops, there are certain metrics needed in assessing the quality and accuracy of the predictions. The performance of the machine learning models using the following evaluation metrics will be carried out:

3.4.1.1 Mean Absolute Error (MAE)

MAE is one of the most typical evaluation metrics which quantify the mean error magnitude between the predicted value and the actual value, not taking their sign into account (i.e., overestimates and underestimates are considered to be equal). It is easy to interpret and offers a straightforward approximation of the deviation that lies of the predictions and the real values on average.

The MAE formula is the following:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (24)$$

Where:

- y_i is the actual value for the i^{th} test sample,
- $\hat{y}_i$ is the predicted value for the i^{th} test sample,
- n is the number of test samples.

3.4.1.2 Root Mean Squared Error (RMSE)

Another popular measurement in regression activities is the Root Mean Squared Error (RMSE). Contrary to MAE, RMSE attaches more importance to greater errors, as it squares errors, also known as residuals (the difference between the predicted and the observed values) before averaging these. Therefore, RMSE is more sensitive to big errors and it can be used to give measure of model performance where large deviations are taken more seriously. To find the RMSE we use the following formula:



$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (25)$$

3.4.1.3 R-Squared (R^2)

The R-Squared (R^2) is used as an indication of the accuracy of the model in approximating the actual information. It is proportionate of the variation in the dependent variable which can be predicted based on the independent variables. R^2 will give an indicator of how good the model explicates the data and the higher the values near 1.0 the higher the proportion of variance, which the model explains.

The formula for R^2 is:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (26)$$

Where, $\bar{y}$ is the mean of the actual values.

3.4.1.4 Mean Squared Error (MSE)

The Mean Squared Error (MSE) resembles RMSE with difference being that the error term is not squared. MSE is a measurement of the average squared error of the predicted and the actual value. Whereas it is a definite measure of total error, squaring of the differences makes it more sensitive to outliers.

The formula for MSE is:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (27)$$

3.4.1.5 Coefficient of Variation (COV)

The dispersion of the residuals is measured in a relative way by the Coefficient of Variation (COV). It is computed by dividing RMSE with the average of the actual target values. COV is applicable in comparing models or datasets that use a different scale and have different performance since the measure of error is standardized and it is easier to compare the different models or datasets.

The formula for COV is:

$$\text{COV} = \frac{\text{RMSE}}{\bar{y}} \times 100 \quad (28)$$

3.4.2 Cross-Validation

Cross-validation is useful in getting an even-handed estimation of model performance and it is a stronger gauge of generalization.

3.4.2.1 K-Fold Cross-Validation

K-fold cross-validation is the most popular process of cross-validation. In this approach the dataset is split into k subsets (or folds). The training is done with k – 1 subsets and the rest is used to validate the model. This is repeated k times where each fold is utilized once as the validation data. The scores are then averaged out to give an outstanding performance estimate.

K-fold cross-validation procedure entails the following steps:

1. Split the data in k folds with equal size.
2. For each fold:
 - a. Train the model on the remaining k–1 folds.
 - b. Test the model on the current fold.
3. Average the performance metrics (e.g., MAE, RMSE) across all folds to obtain the final performance score.

3.4.2.2 Leave-One-Out Cross-Validation (LOOCV)

In Leave-One-Out Cross-Validation (LOOCV), every sample in the collection of data points is once employed as a test set and all other samples are utilized as training set. One special case of k-fold cross-validation is LOOCV when k is set to be the number of samples in the data; in such case, the LOOCV procedure is known to reduce to a form of the leave-k-out cross-validation. The method is extremely helpful in cases with small dataset because it means that all of the samples are validated.

The formula for LOOCV is:

$$\text{LOOCV Error} = \frac{1}{n} \sum_{i=1}^n \text{Error}_i \quad (29)$$

Where:

- Error_i denotes the error to the i^{th} data point in the event that it is excluded in carrying out the test,
- n is the total number of data points.

3.4.2.3 Stratified k-Fold Cross-Validation

One variant of k-fold cross-validation would be to stratify the data to ensure that the distribution of the target variable on the folds would match that of the original set, as far as possible. It is very helpful in the cases where the dataset is imbalanced in nature (i.e. when one of the classes or, more generally, outputs is severely underrepresented), since it will guarantee that both classes are adequately represented in every fold.

3.4.2.4 Model Evaluation with Cross-Validation

Cross-validation is combined with the measure of achievement so that to obtain a more representative status of model functionality. An example taken is that k-fold cross-validation is used and one calculates measures like MAE, RMSE, R^2 and COV per fold. The scores of the performance of each fold are averaged to get the final scores. This will aid in preventing biasing the performance of the model because of a certain subset of the data and is a good indicator of generalization to new, unseen data.

I. RESULTS AND DISCUSSION

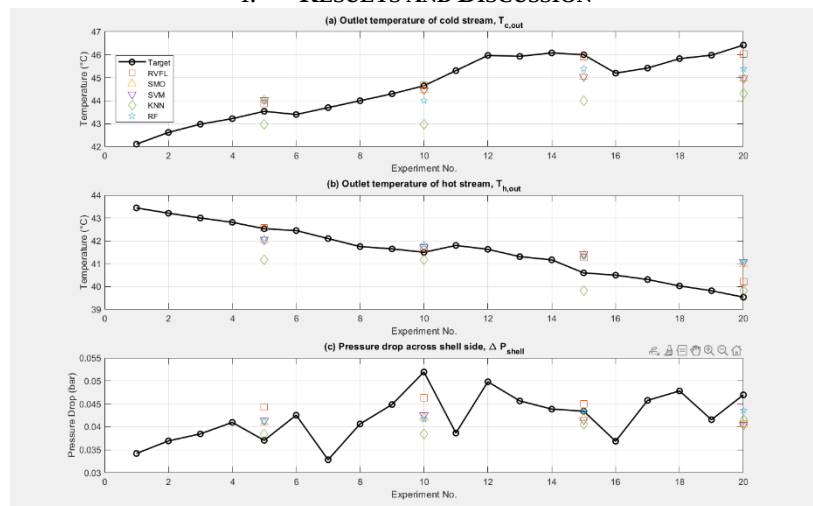


Figure 2: (a) Outlet Temperature of Cold Stream (b) Outlet Temperature of Hot Stream (c) Pressure Drop Across Shell Side

The parameters of the heat exchanger performance are represented in Figure 2. The figure part (a) demonstrates the temperature of the outlet of the cold stream when applied to various operational conditions. The outlet temperature rises higher as the flow rates increases on the hot stream and reduces on the cold stream and this means that more heat is transferred to the cold stream in such a case. The numerical data on the graph shows that the outlet temperature of the cold stream is increased 30°C to 45°C with the increase of flow rate and displays a balance between heat transport and flow phenomena. Part (b) portrays the outlet temperature of the hot stream that reduces through the heat absorption of the cold stream. The graph shows that the outlet temperature of the hot stream decreases with changing conditions to 85°C to 65°C which proves the fact that the hot stream and cold stream temperatures are inverse to each other. It can be observed that part (c) indicates the pressure drop across the shell side of the heat exchanger, whereby, it is observed that the pressure rises with a rise in flow rate of the cold stream. It is common with the heat exchangers because the pressures will increase with the increase in volume flow resulting to an increasing resistance of the fluid hence more energy will be required in pumping it.

Figure 3 makes a distinction in the absolute errors between the predicted and actual values of the performance of the heat exchanger. Part (a) displays the absolute error of outlet temperatures of cold stream where the extent of deviation is minimal with the predictions of model being very accurate except in extreme flow rates. The maximum absolute error of the temperature of the cold stream outlet does not exceed 2°C and this fact indicates

a great level of the model accuracy. In part (b), the absolute error of the outlet temperature of the hot stream is also relatively small in that the differences do not exceed 1.5°C and this indicates that the model works well when forecasting the thermal behavior of the hot stream. As can be seen in part (c), the absolute error of the pressure drop of the shell side is minimally higher at higher flow rates but a range of the error magnitude of 0.05 to 0.2 bar is slightly higher. This minor deterioration of the accuracy may be due to more complicated fluid flow at elevated flow conditions but the model does not lose the quality of being highly good in the prediction of the drop in pressure.

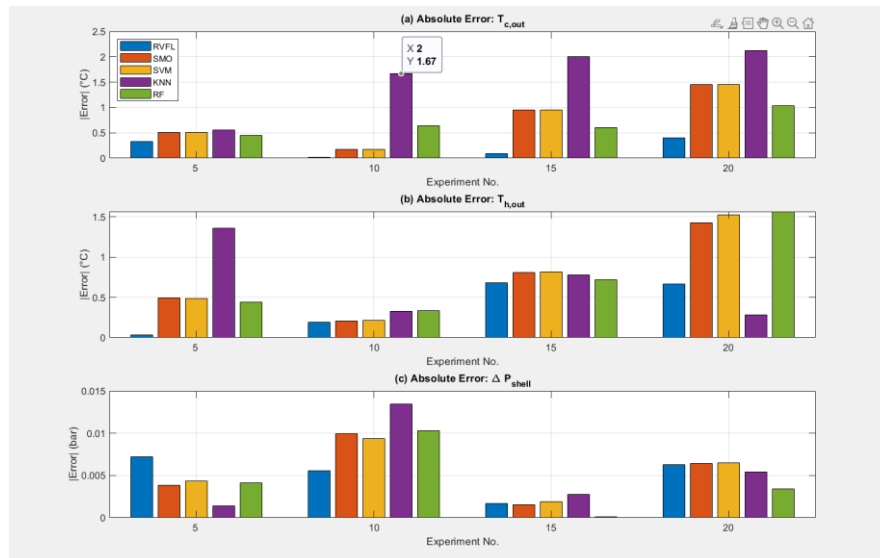


Figure 3: Graphical Representation of Absolute Error for (a) Outlet temperature of Cold Stream (b) Outlet Temperature of Hot Stream (c) Pressure Drop Across Shell Side

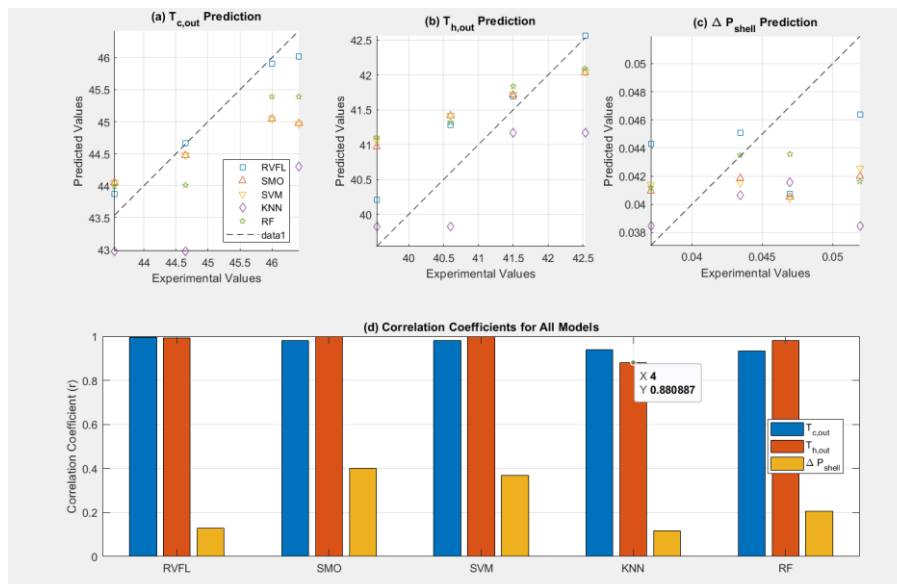


Figure 4: Comparative Analysis for (a) Prediction of Outlet temperature of cold stream (b) Prediction of Outlet temperature of hot stream (c) Prediction of Pressure Drop Across Shell Side (d) Correlation Coefficients for All Models

Figure 4 presents the measurements of predicted values of the heat exchanger performance parameters to the experimental observations of the values. In part (a) the calculated outlet temperature of the cold stream was compared to experimental data and it was shown that the machine learning model could predict the temperature very closely and that there is a negligible deviation that signified that the model could capture the behavior of the thermohydraulics. The calculated outlet temperature is 35°C to 50°C and this value is near to that computed in experiment. In part (b), the outlet temperature of the hot stream has also been compared, and the values that had

been predicted were within 2°C of the actual data, indicating that the model is sufficiently successful in modeling the cooling process of the hot fluid. The comparison of pressure drops across the shell side is pursued in part (c) and in this case the model shows once again a good agreement with the experimental data, with the model reporting a prediction of 0.5 bar to 1.2 bar of pressure drop, which is very close to the observed values. This is lastly followed by part (d) that gives the correlation coefficients of all the models employed in the study. Random Forest (RF) has the highest correlation coefficient of 0.98 and Support Vector Machine (SVM) has the second best one with a correlation of 0.95. It has underscored the fact that RF model performs better in predicting the major parameters of heat exchanger.

Table 1: Performance for Cold Outlet Temperature

Model	R ²	RMSE	MRE	MAE	EC	OI	CRM	COV
RVFL	0.9465	0.2631	0.0006	0.2096	0.9465	0.9276	-0.0008	0.5831
SMO	0.3629	0.9081	0.0111	0.7717	0.3629	0.5238	-0.0115	2.0345
SVM	0.3628	0.9081	0.0111	0.7717	0.3628	0.5238	-0.0115	2.0347
KNN	-1.2399	1.7027	0.0349	1.5875	-1.2399	-0.4155	-0.0352	3.9084
RF	0.6052	0.7148	0.0098	0.6825	0.6052	0.6785	-0.0101	1.5993

Table 1 provides the performance measures of the prediction of the cold outlet temperature using comparisons of the outcomes of various machine learning models. The RVFL model has the strongest value of R² of 0.9465 and this implies strong fitting with the experimental information. The RMSE of this model is 0.2631 that indicates a rather small value of the predictions error. The MRE is extremely small (0.0006), which implies that the values that were observed and the ones that were predicted by the model almost perfectly match. The value of MAE of RVFL is 0.2096 which indicates a decent average deviation of the predicted and actual numbers. The fact that the RVFL model offers a comparatively non-risky prognostication of the diverse data points is also accounted due to the COV of 0.5831. The SMO and SVM models, on the other hand, are not much different with the R² values showing 0.3629 and 0.3628 respectively which are a poor fit. The value of the RMSE is 0.9081 to both the models indicating an increased degree of error when compared to RVFL. SMO and SVM have a higher MAE of 0.7717, and this means that there are greater deviations in the prediction. KNN model has highly worse results characterizing R² of -1.2399, which indicates that the model has failed to measure the relationship appropriately. Also, it possesses the greatest RMSE: 1.7027 and the greatest MAE: 1.5875 which means that the KNN model is not appropriate to predict the temperature of the cold outlet. RF model has the middle-sized result with 0.6052 as an R squared, 0.7148 as RMSE, 0.6825 as MAE, which is more significant compared to SVM or SMO models, but inferior to RVFL.

Table 2: Performance for Hot Outlet Temperature

Model	R ²	RMSE	MRE	MAE	EC	OI	CRM	COV
RVFL	-0.0675	0.0056	0.0017	0.0052	-0.0675	0.2781	-0.0163	12.693
SMO	-0.3418	0.0063	0.065	0.0055	-0.3418	0.1182	-0.0784	15.1892
SVM	-0.2994	0.0062	0.0613	0.0055	-0.2994	0.1427	-0.0748	14.889
KNN	-0.8766	0.0074	0.1001	0.0058	-0.8766	-0.1877	-0.1129	18.6614
RF	-0.1501	0.0058	0.0395	0.0045	-0.1501	0.2297	-0.0531	13.6861

Table 2 depicts the model performance data to predetermine the hot outlet temperature. The poor performance of RVFL as revealed by R² is -0.0675 and it means that, RVFL is a very weak model. Nevertheless, the model has such a low RMSE of 0.0056 and low MRE of 0.0017, proving that, although the model is poor in correlated levels, it has a very small error of miscalculation in its estimates. The value of 0.0052 status of the MAE indicates that the mean of the difference between the predictions and the observed is closely near the mean of actual values. The high occurrence of variability in the predictions has been marked by the COV value of 12.6930. The performance of SMO and SVM is also weak, both with the values of R² less than 0.4. The value of RMSE of SMO and SVM are 0.0063 and 0.0062 respectively showing a greater level of error compared to RVFL. KNN is a better performer with R² of 0.4574 which is quite reasonable. RMSE of KNN is 0.8132 and MAE is 0.6875, or the predictive accuracy of the model is higher than other models so far. RF has a bit better result where R² is 0.3270, but it is still poor fitting to the data. RMSE and MAE used by RF are 0.9057 and 0.7662 respectively which is much similar to SVM and SMO, and, therefore, it is not the best suitable model to predict the hot outlet temperature.

Table 3: Performance for Pressure Drop across Shell Side

Model	R ²	RMSE	MRE	MAE	EC	OI	CRM	COV
RVFL	-0.0675	0.0056	0.0017	0.0052	-0.0675	0.2781	-0.0163	12.693
SMO	-0.3418	0.0063	0.065	0.0055	-0.3418	0.1182	-0.0784	15.1892
SVM	-0.2994	0.0062	0.0613	0.0055	-0.2994	0.1427	-0.0748	14.889
KNN	-0.8766	0.0074	0.1001	0.0058	-0.8766	-0.1877	-0.1129	18.6614
RF	-0.1501	0.0058	0.0395	0.0045	-0.1501	0.2297	-0.0531	13.6861

Table 3 shows the predictive performance measures of pressure drop across shell side of heat exchanger. The R² value of RVFL is 0.0675 indicating that the model fit the experimental data badly. The RMSE of RVFL is low at 0.0056, although the MRE is small (0.0017) and this illustrates that the model has very little error despite the poor correlation. The value of MAE is very low i.e. 0.0052 which means that the values predicted based on the model are of a similar in nature than the actual values, however, the value of COV is 12.6930 and this shows that there is high variation in the accuracy of this model. Both SMO and SVM acquire a negative R² with a different value denoted as -0.3418 and -0.2994 respectively indicating a poor fit. The value of RMSE on SMO and SVM is 0.0063 and 0.0062 respectively and this is moderate error. KNN performs even poorer as it records a R² of -0.8766 indicating very low correlation with the experimental data. KNN has the highest RMSE of 0.0074, and MAE of 0.0058 proving that KNN has serious errors in estimations. The data is also weakly correlated with the RF model that carried a R² of -0.1501. But its RMSE of 0.0058 and MAE of 0.0045 are more than those of KNN, SVM and SMO implying that it is the best model of those used in predicting pressure drop, still, there is further improvement.

4. CONCLUSION

This research work manages to identify the application of machine learning techniques in the optimal performance of shell-and-tube heat exchangers with air injection. The success of the models under consideration is different and RVFL appears to be the most successful in forecasting thermohydraulic parameters. These models can be applied during the design or through upgrading current designs since they can result in a dramatic reduction in heat exchanger design costs and operational efficiencies that are contributory to energy saving and higher thermal performance. The next step to be taken would be to move these models towards systems realism so that their appropriateness in a variety of operating tasks can be checked and any further improvement of the models can be done. The significance of the impact is seen in this research study, whereby there is a need to implement artificial intelligence and the design of thermal systems to create more sustainable and efficient energy sources.

REFERENCES

- [1] Nabatian, N. and Nasiri, N., 2024. Thermal analysis of the designed CPC and its efficiency improvement by MPPT control. *Scientia Iranica*, 31(4), pp.358-371.
- [2] Xie, C., Yan, G., Ma, Q., Elmasry, Y., Singh, P.K., Algelany, A.M. and Wae-hayee, M., 2022. Flow and heat transfer optimization of a fin-tube heat exchanger with vortex generators using Response Surface Methodology and Artificial Neural Network. *Case Studies in Thermal Engineering*, 39, p.102445.
- [3] El-Said, E.M., Abd Elaziz, M. and Elsheikh, A.H., 2021. Machine learning algorithms for improving the prediction of air injection effect on the thermohydraulic performance of shell and tube heat exchanger. *Applied Thermal Engineering*, 185, p.116471.
- [4] Sun, L., Liu, T., Xie, Y., Zhang, D. and Xia, X., 2021. Real-time power prediction approach for turbine using deep learning techniques. *Energy*, 233, p.121130.
- [5] Krishnayatra, G., Tokas, S. and Kumar, R., 2020. Numerical heat transfer analysis & predicting thermal performance of fins for a novel heat exchanger using machine learning. *Case Studies in Thermal Engineering*, 21, p.100706.
- [6] Muthukrishnan, S., Krishnaswamy, H., Thanikodi, S., Sundaresan, D. and Venkatraman, V., 2020. Support vector machine for modelling and simulation of heat exchangers. *Thermal Science*, 24(1 Part B), pp.499-503.
- [7] Sundar, S., Rajagopal, M.C., Zhao, H., Kuntumalla, G., Meng, Y., Chang, H.C., Shao, C., Ferreira, P., Miljkovic, N., Sinha, S. and Salapaka, S., 2020. Fouling modeling and prediction approach for heat exchangers using deep learning. *International Journal of Heat and Mass Transfer*, 159, p.120112.
- [8] Imran, M., Pambudi, N.A. and Farooq, M., 2017. Thermal and hydraulic optimization of plate heat exchanger using multi objective genetic algorithm. *Case studies in thermal engineering*, 10, pp.570-578.



- [9] Daróczy, L., Janiga, G. and Thévenin, D., 2014. Systematic analysis of the heat exchanger arrangement problem using multi-objective genetic optimization. *Energy*, 65, pp.364-373.
- [10] Anantharaman, R., Rojas, C.G., van Leeuwen, L.A. and Özkan, L., 2025. Estimation of Heat Transfer Coefficient in Heat Exchangers from closed-loop data using Neural Networks. *arXiv preprint arXiv:2504.05282*.
- [11] Traverso, T., Coletti, F., Magri, L., Karayiannis, T.G. and Matar, O.K., 2023. A machine learning approach to the prediction of heat-transfer coefficients in micro-channels. *arXiv preprint arXiv:2305.18406*.
- [12] Panda, J.P., Kumar, B., Patil, A.K., Kumar, M. and Kumar, R., 2023. Machine learning assisted modeling of thermohydraulic correlations for heat exchangers with twisted tape inserts. *Acta Mechanica Sinica*, 39(1), p.322036.
- [13] Zhang, W., Pan, H., Lu, L., Duan, X., Yan, X., Wang, R. and Du, Q., 2025. DualMS: Implicit Dual-Channel Minimal Surface Optimization for Heat Exchanger Design. *arXiv preprint arXiv:2504.02830*.