PIU-CT Gap Measurement using ML Algorithms

Given Name Surname   
*dept. name of organization   
(of Affiliation)*  
*name of organization   
(of Affiliation)*City, Country  
email address or ORCID

*Abstract*—IEEE format requires you to include an abstract at the start of your paper, followed by a list of keywords. In the “Styles” section in Microsoft Word, you can find the appropriate styles for all the different sections and headings in the paper, which are already applied here. For example, the “abstract” style is applied to this text, the “keywords” style to the next section. Note that the titles “Abstract” and “Keyword” should remain as they are written here: italicized and followed by an em dash.

It’s recommended to insert figures inside a text box. So your figure would go here (adjusting the text box to the appropriate size to match).

Keywords—template, Scribbr, IEEE, format

# INTRODUCTION

In the context of nuclear power plant maintenance, the accurate measurement of the gap between the calandria tube and the pressure tube in heavy water reactors is a crucial task to ensure the safe and reliable operation of the reactor (Study on Inspection for Small Diameter Tubes Using Pulsed Remote Field Eddy Current Method, 2021)(Grimberg et al., 2005)(Zhu et al., 2019). The eddy current method has emerged as a preferred technique for this application due to its inherent non-invasive nature and capability to detect even minor variations in the gap between these two critical components (Grimberg et al., 2005).

The eddy current method involves the generation of a time-varying magnetic field that induces eddy currents in the conductive materials, such as the calandria tube and pressure tube. The presence of the two tubes with a gap between them affects the flow of the induced eddy currents, which in turn modulates the magnetic field in a manner that can be detected and measured by the eddy current sensor. The specific characteristics of this magnetic field modulation, such as its amplitude, phase, and frequency, are directly correlated with the size and properties of the gap between the tubes, thereby enabling the precise measurement of the gap using advanced signal processing and analysis techniques. Recent advancements in the field of machine learning have provided researchers and engineers with a powerful set of tools to enhance the performance and accuracy of eddy current-based gap measurement systems in heavy water reactors.

One such application of machine learning in this context is the use of advanced signal processing techniques, such as neural networks or support vector machines, to analyze the complex eddy current sensor data and extract more accurate information about the gap between the calandria tube and poisson injection tube. To develop robust models capable of reliably predicting the gap size and detecting any anomalies or deviations, these machine learning algorithms can be trained on a comprehensive dataset that may encompass both simulated and experimental eddy current measurement data.

The use of machine learning algorithms in the analysis of eddy current data for gap measurement in heavy water reactors offers several advantages over traditional signal processing methods. Firstly, the ability of machine learning models to handle complex, non-linear relationships between the eddy current sensor signals and the gap size can lead to significantly improved accuracy and sensitivity in the gap measurement.(Grimberg et al., 2006)(Park et al., 2019).

Additionally, machine learning techniques can enhance the robustness of the gap measurement system by enabling the detection and classification of various types of defects or anomalies that may be present in the calandria tube or pressure tube.

## Literature Survey

### And this is a level 3 heading: Equations should be typed in either Times New Roman or Symbol font, or, if the equation is multileveled, inserted into your text as a graphic instead. On the far right of the line containing the equation, number it in parentheses, and use this number to refer to it in the text (1).

*a**b* 

### This is another level 3 heading: The body text is divided into two columns on each page, written in 10 pt. Times New Roman, and justified (meaning that the text is spaced in a way that makes the right edge line up neatly). All the appropriate formatting is automatically applied in this template. If anything goes wrong, you can reapply it using the “styles” section in Word.

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| text | Texta |  |  |

#### And this is a level 4 heading: It’s recommended to write your text in a separate document and then add it to this template once it’s complete. When copying text into the template from another document, make sure that the

1. This is a figure caption. It appears directly underneath the figure.

appropriate style is still applied to each section, reapplying styles if necessary.

#### This is another level 4 heading: It’s also possible to add bullet points when appropriate, using the “bullet list” style:

* Treat the word “data” as plural, not singular.
* For example, “the data indicate that …”

## Methodology

Place any figures or tables you use at the top or bottom of a column. Don’t place them in the middle of a column. If particularly wide, a table or figure can span across both columns. Insert a table or figure after the point where it is first cited in the text.

When inserting a figure, such as a photograph or infographic, use 8 pt. Times New Roman for any labeling text within the image and for the figure caption. You can see an example of a figure caption in Fig. 1, above. Refer to figures like that, using the abbreviation “Fig.” and the figure’s number.

A table heading (using the “table head” style) appears above a table. This will automatically number the table for you. Any footnotes appear below the table, using the “table footnote” style. Footnotes are indicated by superscript lowercase letters within the table. An example of a table can be seen in Table I, below.

# LITERATURE SURVEY

III. METHODOLOGY

## Dataset

The dataset for this study was acquired using the Eddy Current Testing (ECT) method, which provided raw signal data indicative of the gap between the Poisson injection tube and the Calandria tube in a heavy water reactor. To transform this raw signal data into a more useful form for machine learning, we employed the TSFEL (Time Series Feature Extraction Library) package. This package facilitated the extraction of a comprehensive set of features across various domains, including the time domain, frequency domain, and statistical domain. Features such as mean, median, variance, peak-to-peak amplitude, spectral entropy, dominant frequencies, spectral centroid, skewness, kurtosis, and interquartile range were extracted. This feature extraction process transformed the raw ECT signals into a structured dataset with a rich set of attributes, enabling the application of various machine learning algorithms to predict the gap size with greater accuracy and precision.

## Preprocessing

Data preprocessing is a critical step in preparing raw signal data for machine learning applications. It involves a series of transformations aimed at improving data quality and making it suitable for analysis. Key preprocessing steps include noise removal and feature scaling, which enhance the signal-to-noise ratio and standardize the data range, respectively. These steps ensure that the extracted features are robust and the models trained on this data are accurate and reliable.

## *Denoising / scaling / normalization*

To ensure the integrity and quality of the signal data, we employed the Savitzky-Golay filter for noise removal. This filter is particularly effective for smoothing noisy data while preserving the essential features of the signal, such as peak height and width. By fitting successive sub-sets of the data with a polynomial of a specified degree, the Savitzky-Golay filter enhances the signal-to-noise ratio without significantly distorting the signal’s important characteristics. This step was crucial for reducing the impact of noise on subsequent feature extraction and model training processes.(Savitzky, A., & Golay, M. J. E.)(Dombi, J., & Dineva, A.)

Following noise removal, we applied scaling techniques to standardize the range of features. We explored two scaling techniques:

* Standard Scaler: This technique standardizes the features by removing the mean and scaling to unit variance, resulting in a distribution with a mean of 0 and a standard deviation of 1. This is particularly useful for algorithms that assume normally distributed data.
* MinMax Scaler: This technique normalizes the features to a fixed range, typically [0, 1], by transforming the data based on the minimum and maximum values in each feature. This approach is beneficial for preserving the relationships between the original data values.

Evaluating these scaling techniques allowed us to determine the optimal method for improving model performance and achieving consistent, accurate predictions for the gap size between the Poisson injection tube and the Calandria tube.

## *Feature Extraction: (correlation, tsfel domains)*

After denoising the raw signal data obtained using the Eddy Current method, we proceeded with feature extraction to facilitate further analysis. The denoised signal data was processed using the Time Series Feature Extraction Library (TSFEL) package. TSFEL is a comprehensive tool designed to automatically extract a wide range of features from time series data. This package is highly regarded for its ability to compute over 60 distinct features, spanning various domains including statistical, temporal, and spectral characteristics. These features include measures such as mean, variance, entropy, and frequency domain features, which are crucial for capturing the underlying patterns in the signal data (Barandas, M., Folgado, D., Fernandes, R., Santos, R., Abreu, P. H., Bota, P., & Gamboa, H.).

Given the high dimensionality of the extracted feature set, which resulted in a total of 378 features, it was crucial to perform feature selection to enhance the performance and interpretability of our predictive models. To address potential multicollinearity, we initially computed the correlation matrix of the features and identified pairs of features with a Pearson correlation coefficient greater than 0.9. For each highly correlated pair, we retained only one feature, thereby reducing redundancy in our dataset. This step yielded a reduced feature set while preserving the most informative features.

## *Feature Importance: (gbc, rf, xgb)*

To further refine our feature set, we employed three different machine learning algorithms to calculate feature importances: Gradient Boosting (GB), Random Forest (RF), and XGBoost (XGB). Each of these algorithms provides a mechanism to rank features based on their importance scores derived from the trained models. Using these models, we trained regressors on the reduced feature set and computed feature importances for each algorithm. The features were then ranked according to their importance scores from each model.

* XGBOOST FEATURE IMPORTANCE

XGBoost, an optimized gradient boosting framework for handling large datasets, calculates feature importance based on three main metrics. "Gain" measures the improvement in model performance attributable to each feature across all trees. "Weight" denotes how frequently a feature is used for splitting data. Higher weights signify more crucial features. "Coverage" reflects the number of observations impacted by splits using a feature. XGBoost’s feature importance is accessible post-training via the feature\_importances\_ attribute, offering insights into which features contribute most significantly to model accuracy and aiding in feature selection and interpretation in complex datasets.( Chen, T., & Guestrin, C.)

* RANDOM FOREST FEATURE IMPORTANCE

Random Forest constructs multiple decision trees during training and calculates feature importance based on how much each feature decreases node impurity (e.g., Gini impurity) across all trees in the ensemble. Features that consistently reduce impurity more effectively are considered more important. Random Forest also utilizes permutation importance, measuring the impact of shuffling feature values on model accuracy. These methods provide insights into which features contribute significantly to model predictions, aiding in feature selection and interpretation in complex datasets.(Brieman L.,2001)

* GRADIENT BOOSTING FEATURE IMPORTANCE

Gradient Boosting constructs an ensemble of decision trees sequentially, improving model performance by correcting errors of previous models. Feature importance in Gradient Boosting is determined by how much each feature contributes to the model's performance improvement, such as reducing the loss function (e.g., MSE). The importance is calculated based on the frequency of feature usage in tree splits across the ensemble. Gradient Boosting also utilizes advanced methods like SHAP (SHapley Additive exPlanations) to provide nuanced insights into feature contributions to model predictions.(Friedman, J. H. (2001))

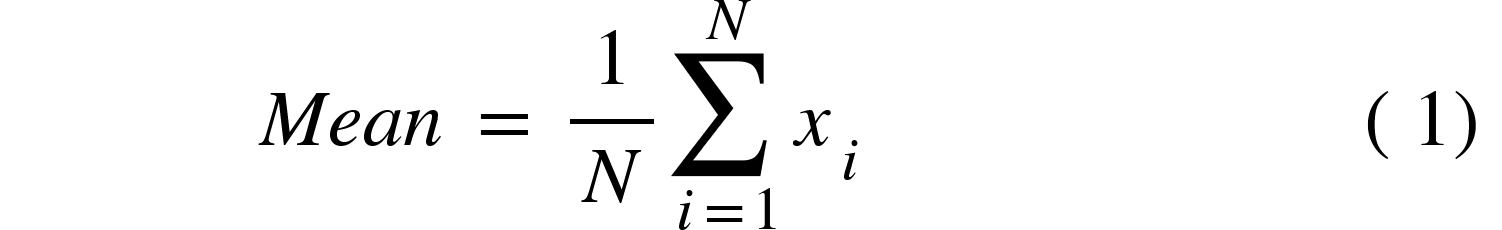
## *TSFEL features: important features\*\**

TSFEL is a Python library designed for automatic extraction of comprehensive features from time series data, facilitating detailed analysis and modeling in various research and industrial applications. It provides a robust set of features across statistical, temporal, and spectral domains, enabling researchers to extract meaningful insights from time series data efficiently. These features play a crucial role in characterizing signal properties, identifying patterns, and enhancing the interpretability of machine learning models applied to time series analysis. (Barandas, M., Folgado, D., Fernandes, R., Santos, R., Abreu, P. H., Bota, P., & Gamboa, H. (2020).)

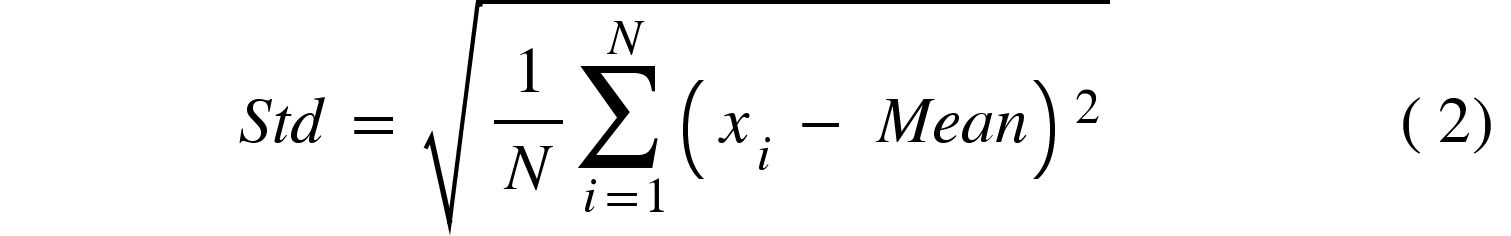
Statistical Features:

Statistical features are derived from the raw time series data and provide summary statistics that describe the distribution and variability of the data. These features help in understanding the overall characteristics and trends within the time series.

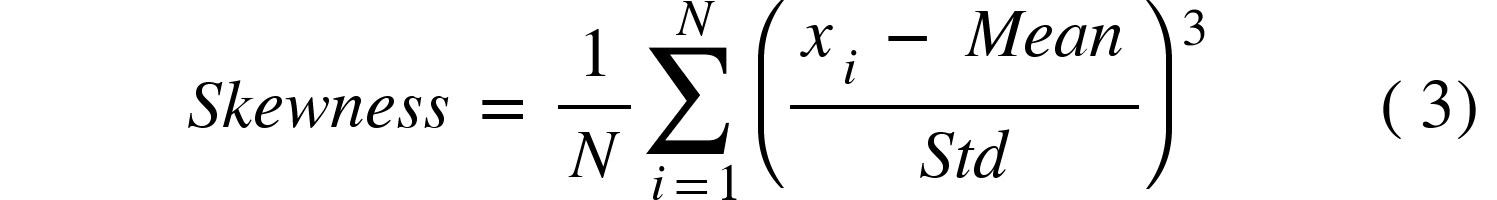
* Mean: Represents the average signal value over the entire series. It provides a measure of the central tendency of the data.



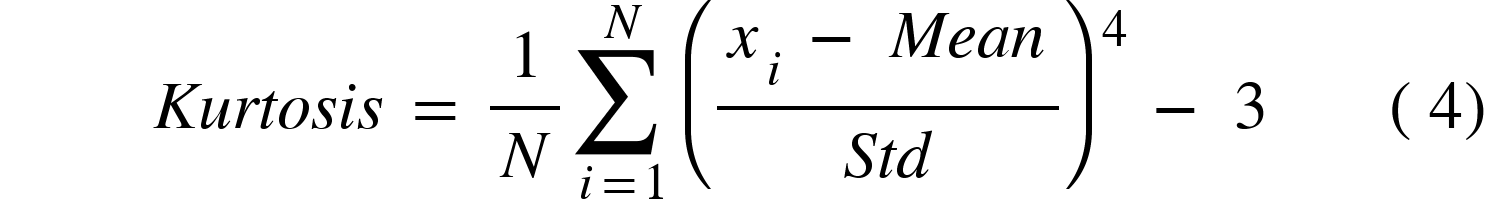
* Standard Deviation: Measures the dispersion of signal values around the mean. A higher standard deviation indicates greater variability in the data.



* Skewness: Indicates the asymmetry of the distribution of signal values. Positive skewness means the distribution tail is longer on the right side, while negative skewness means the tail is longer on the left.



* Kurtosis: Measures the "tailedness" or degree of outlier presence in the signal. High kurtosis indicates heavy tails and potential outliers.

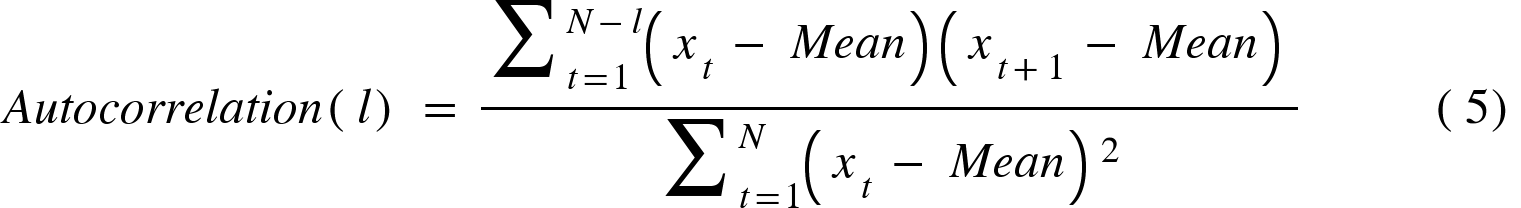


Statistical features provide a foundational understanding of the data’s distribution and variability, which is crucial for identifying patterns and anomalies in time series analysis.

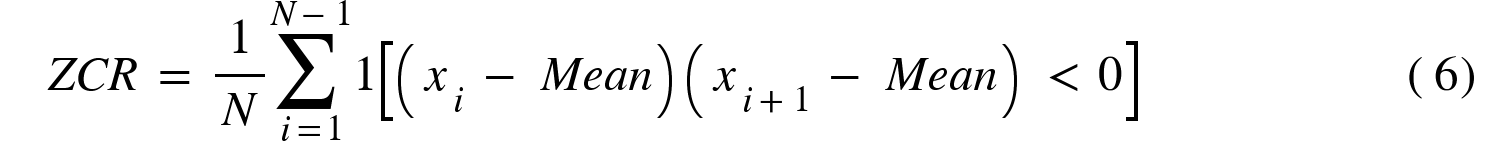
Temporal Features:

Temporal features capture the time-dependent characteristics of the signal, focusing on the structure and behavior over time. These features help in understanding the dynamics and temporal dependencies within the data.

* Autocorrelation: Quantifies the similarity between a signal and a delayed version of itself. It provides insights into the repeating patterns or periodicity in the data.



* Zero Crossing Rate: Counts the number of times the signal crosses the mean value. It is useful for identifying the frequency of oscillations in the time series.



* Peak-to-Peak Distance: Measures the distance between the highest and lowest values in the signal within a given window. It provides a measure of the signal's amplitude variability.

Temporal features capture the sequential and temporal relationships in the data, which are essential for understanding the time-dependent behaviors and trends in time series analysis.

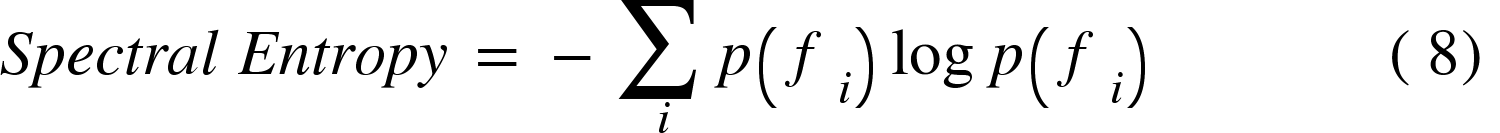
Spectral Features:

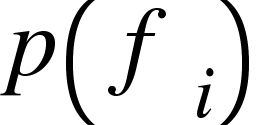
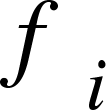
Spectral features are derived from the frequency domain representation of the signal. These features help in understanding the signal’s frequency content and identifying periodicities and frequency-based patterns.

* Power Spectral Density (PSD): Describes the distribution of signal power across different frequencies. It is obtained by performing a Fast Fourier Transform (FFT) on the signal and calculating the power at each frequency component.

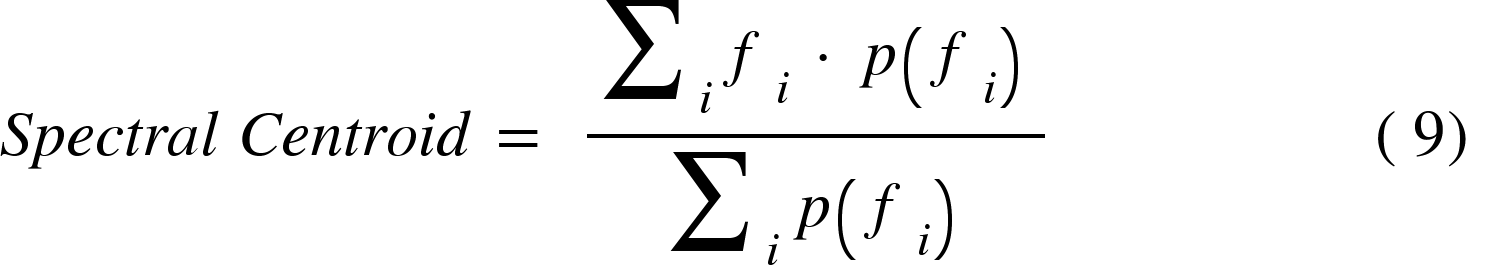
{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mi>P</mi><mi>S</mi><mi>D</mi><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><mi>F</mi><mi>F</mi><mi>T</mi><mfenced><mrow><mi>A</mi><mi>u</mi><mi>t</mi><mi>o</mi><mi>c</mi><mi>o</mi><mi>r</mi><mi>r</mi><mi>e</mi><mi>l</mi><mi>a</mi><mi>t</mi><mi>i</mi><mi>o</mi><mi>n</mi></mrow></mfenced><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>&#xA0;</mo><mfenced><mn>7</mn></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}

* Spectral Entropy: Measures the complexity or irregularity of the frequency distribution. Higher spectral entropy indicates a more complex and less predictable signal.



where is the normalized power at frequency .

* Spectral Centroid: Represents the center of mass of the spectrum, providing a measure of the "brightness" of the signal. It is the weighted mean of the frequencies present in the signal.



Spectral features provide insights into the frequency-based characteristics of the signal, helping to identify periodic components, dominant frequencies, and other frequency-related patterns in time series analysis.

## ML Algorithms

Machine learning is a rapidly advancing field that forms the foundation of artificial intelligence and data science, bridging computer science and statistics. It enables data-driven decision-making in numerous domains such as health, production, education, financial modeling, law enforcement, and marketing. This research investigates various machine learning models, particularly in the context of predicting gaps between the Poisson injection tube and the Calandria tube in a heavy water reactor.

In this study, we employed a range of machine learning algorithms for both classification and regression tasks. The models included Random Forest, Gradient Boosting, Decision Trees, and XGBoost. Each of these models brings unique characteristics to the table, helping us analyze and predict the gap measurements effectively.

*1) Random Forest:* Bagging stands out as a well-known approach in Ensemble learning, categorically grouped into three types – Bagging, Boosting, and Stacking – which are all ensemble learning techniques. Random Forest L. Breiman, "Random Forest," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001., is widely recognized and frequently employed, representing an extended version stemming from Bagging. (Bo Yu and Yuanzheng Zheng, "Research on Algorithms of Machine Learning," Applied and Computational Engineering, vol. 39, no. 1, EWA Publishing, Feb. 2024, pp. 277-281, doi: 10.54254/2755-2721/39/20230614.) Random Forest constructs multiple decision trees during training. It outputs the mode of classes (classification) or the mean prediction (regression) of individual trees. In classification, each tree votes for a class, improving robustness and accuracy through majority voting. In regression, trees predict values and the forest averages these predictions to reduce variance. It uses bootstrap aggregating (bagging) on random data subsets and random feature selection for each split. This creates uncorrelated trees for more accurate, stable predictions, handling large datasets with high dimensionality and providing insights into feature importance in both tasks.

*2) Decision Tree:* A Decision Tree is a versatile model used for both classification and regression tasks. It recursively splits the data into subsets based on feature values, forming a tree-like structure where each node represents a feature, each branch signifies a decision rule, and each leaf denotes a prediction outcome. In classification, each leaf node corresponds to a class label determined by the majority class within that node. For regression, leaf nodes contain average values of the target variable, providing predictions based on these averages. Decision Trees are straightforward to interpret, require minimal data preprocessing, and accommodate both numerical and categorical data. However, they can overfit, particularly with complex structures, and pruning techniques, such as removing unimportant branches, are employed to enhance generalization.

*3) Gradient Boosting:* Gradient Boosting Classifier sequentially builds models where each new model corrects errors of the previous ones using gradient descent to optimize a loss function. It combines weak learners, typically decision trees, to form a strong learner by fitting models to the residuals of prior ones, reducing bias. In classification, it predicts class labels by combining tree predictions, iteratively refining them to minimize errors. In regression, it predicts continuous values by aggregating tree predictions. Key features include handling diverse loss functions, controlling contributions with a learning rate, and achieving high predictive accuracy across various tasks despite its complexity.

*4) XGBoost:* XGBoost is a scalable decision tree ensemble based on gradient boosting, designed to build models by minimizing a loss function with regularization to control tree complexity. Higher gamma values result in simpler trees by increasing the minimum loss reduction needed to split a node, while shrinkage reduces the step size in the additive model. Randomization techniques like random subsampling for training trees and column subsampling at tree and node levels help reduce overfitting and speed up training. XGBoost optimizes split finding by using a compressed column-based structure to store pre-sorted data, enabling parallel processing and improving efficiency by testing a subset of candidate splits based on percentiles. A sparsity-aware algorithm effectively handles missing values. Key parameters tuned include learning rate, gamma, maximum depth, feature fraction per split, and subsampling rate.(Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz, "A Comparative Analysis of XGBoost," 2019.)

## *5) SVM:* Support Vector Machine (SVM) is a versatile supervised learning algorithm for classification and regression tasks. In classification, SVM finds the optimal hyperplane that maximizes the margin between classes, using support vectors (the closest data points) to define this boundary. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces (Batta Mahesh, “Machine Learning Algorithms - A Review”, International Journal of Science and Research (IJSR), Volume 9 Issue 1, January 2020, DOI: 10.21275/ART20203995). In regression, SVM fits data within a specified margin of tolerance, the epsilon-tube, and penalizes data points outside this tube. Both applications of SVM rely on maximizing the margin to achieve robust predictions.

*6) AdaBoost:* AdaBoost algorithm is the most typical and widely used algorithm of ensemble 87 learning method, or more specifically, of the boosting family of ensemble learning. The characteristic of 88 AdaBoost is to use the initial training data to generate a weak learner, then adjust the distribution of the 89 training data according to the predicting performance for the next round weak learner training. The training samples with low predicting accuracy in the previous step will get more attention in the next 91 step. Finally the weak learners are integrated together with different weights to a strong learner.(De-Cheng Feng, Zhen-Tao Liu, Xiao-Dan Wang, Zhongming Jiang, "Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach," *Construction and Building Materials*, vol. 230, p. 117000, January 2020. doi:10.1016/j.conbuildmat.2019.117000.)

*7) Linear Regression:* Linear regression, the most common form of regression analysis, assumes a linear relationship between the dependent variable (the variable being predicted, also known as the response or outcome variable) and the independent variable(s) chosen by researchers (the known or hypothesized predictor variable[s]). It aims to find a "best fit" line that mathematically describes this relationship, visually represented by a scatter plot with a fitted line. For linear regression to apply, the outcome variable must be continuous, meaning it has a numerical range rather than categorical outcomes.([1] A. Worster, J. Fan, and A. Ismaila, “Understanding linear and logistic regression analyses,” Canadian Journal of Emergency Medicine, vol. 9, no. 2, pp. 111–113, 2007. doi:10.1017/S1481803500014883)

*8) Logistic Regression:* Logistic regression is a statistical technique used for binary classification tasks, where the outcome variable is categorical and takes on two possible values. Unlike linear regression, which predicts continuous outcomes, logistic regression models the probability that an observation belongs to a specific category. It employs a logistic (sigmoid) function to transform a linear combination of input variables into a probability score between 0 and 1, indicating the likelihood of the outcome being in one category. This method is widely applied in fields such as healthcare (for disease prediction), finance (credit scoring), and marketing (customer segmentation), offering interpretable results and facilitating decision-making based on calculated probabilities rather than simple categorical predictions.

*9) Ridge and Lasso Regression:*

## Neural Models \*\*

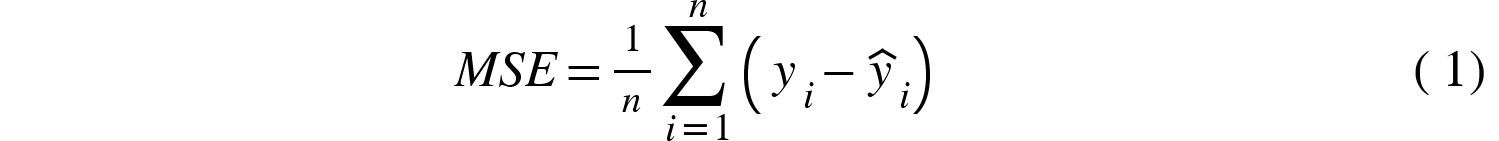
## LSTM....

IV. EXPERIMENTAL SETUP

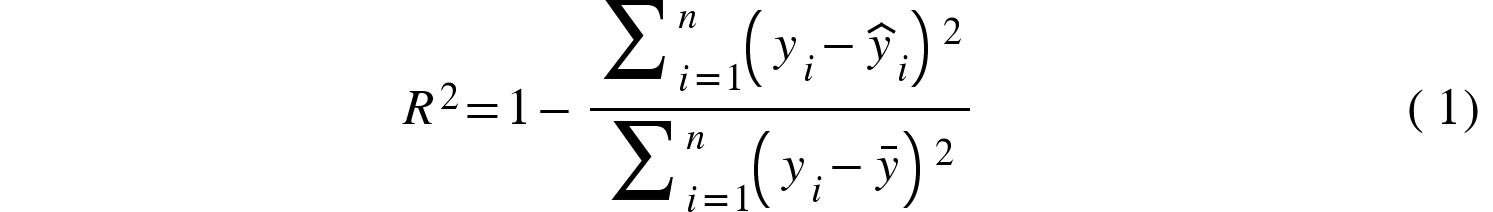
## A. Evaluation Metrics

To evaluate the performance of the machine learning models used in predicting the gap between the Poisson Injection tube and the Calandria tube in a heavy water reactor, we employed both regression and classification metrics. Initially, we performed feature extraction on the given signal data containing measurements from the eddy current testing (ECT) method to transform it into a suitable format for model training and evaluation.

For regression models, we utilized Mean Squared Error (MSE) and the R-squared (R²) score as our primary metrics. These metrics help in quantifying the accuracy of the continuous gap measurements predicted by the models. The MSE provides insight into the average squared difference between the observed and predicted values, where  are the observed values, and {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><semantics><mstyle mathsize=\"16px\"><msub><mover><mi>y</mi><mo>^</mo></mover><mi>i</mi></msub></mstyle><annotation encoding=\"application/json\">{\"x\":[[236,236,236,236,236,236,236,237,237,237,237,237,237,237,238,238,239,240,240,241,242,243,245,247,249,251,252,254,256,258,260,264,267,269,272,275,278,280,281,282,282,282,282,282,281,278,278,278,278,278,277,277,277,277,277,277,277,277,277,277,277,277,277,277,277,277,277,277,278,278,278,280,282,282,282,282,282,282,282,282,282,282,282,282,281,280,278,277,276,276,274,274,273,272,267,265,264,262,260,256,251,250,247,247,246,243,242,241,241,241,241,241,241,241,241],[241,241,241,245,247,247,247,247,249,249,249,251,251,251,251,252,252,252,252,254,254,254,255,256,256,256,256,257,260,264,264,264,265,267,268,270,271,272,273,274,275,276,276,276],[292,292,292,292,292,292,292,292,292,292,292,292,292,292,293,294,296,297,299,301,302,304,306,307,308,308,309,310,310,311,311,310],[294,294,294,295,296,297,298,299,299,299,298,296,296,294,294,294,292,292,292,293,293,294,295,296,296,296,296,296]],\"y\":[[108,108,110,113,117,119,122,132,134,136,136,138,140,141,142,144,147,148,149,150,151,153,154,156,158,160,160,160,160,160,158,158,156,155,153,150,148,147,145,143,142,140,139,136,128,122,121,120,119,116,113,112,112,111,112,112,114,117,118,121,126,134,137,140,144,148,152,156,161,165,168,178,187,188,189,190,196,199,201,202,202,204,206,208,209,212,219,224,225,226,226,226,226,227,230,231,231,231,230,230,228,228,225,224,222,221,220,219,216,215,214,212,211,210,209],[87,86,84,72,66,65,64,63,56,54,54,48,47,46,45,40,38,37,36,36,37,38,39,41,42,44,47,50,61,73,74,75,77,80,82,85,87,88,90,92,95,96,97,98],[160,161,171,176,177,178,179,180,184,185,187,188,189,190,191,193,195,196,197,198,198,198,198,198,198,197,196,193,192,191,190,190],[146,146,145,144,144,143,142,141,140,138,138,138,138,138,139,140,140,142,142,143,144,144,144,144,142,142,141,141]],\"t\":[[0,267,300,333,367,400,434,467,500,533,567,600,633,667,700,733,767,800,833,866,900,933,966,1000,1033,1066,1100,1134,1167,1200,1233,1267,1300,1333,1367,1400,1433,1466,1500,1533,1566,1600,1633,1666,1700,1733,1766,1800,1851,1933,1966,1999,2033,2066,2189,2217,2250,2283,2316,2350,2383,2416,2449,2483,2516,2550,2583,2616,2649,2683,2716,2749,2783,2816,2849,2883,2916,2949,2983,3016,3050,3083,3116,3149,3183,3216,3249,3283,3316,3349,3383,3416,3449,3483,3516,3549,3583,3616,3649,3682,3716,3749,3782,3816,3849,3883,3916,3950,3983,4016,4049,4082,4116,4150,4183],[5456,5707,5749,5782,5816,5849,5882,5915,5949,5982,6015,6049,6082,6116,6149,6182,6215,6249,6283,6315,6349,6382,6415,6449,6482,6515,6549,6582,6615,6649,6682,6716,6749,6782,6815,6849,6882,6915,6949,6982,7015,7049,7082,7115],[8871,9081,9115,9148,9181,9215,9248,9292,9315,9348,9382,9415,9462,9481,9515,9548,9581,9615,9648,9681,9715,9748,9782,9815,9849,9882,9915,9948,9981,10015,10048,10091],[11104,11421,11434,11465,11498,11532,11564,11598,11631,11665,11698,11731,11764,11798,11831,11865,11898,11931,11965,11998,12031,12064,12098,12131,12164,12198,12231,12264]],\"version\":\"2.0.0\"}</annotation></semantics></math>","origin":"MathType for Microsoft Add-in"} are the predicted values, and n is the number of observations.

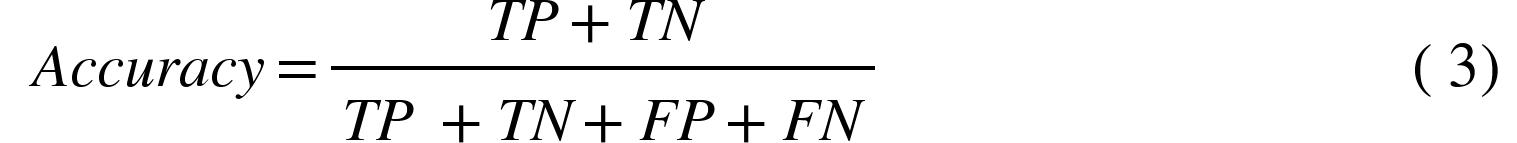


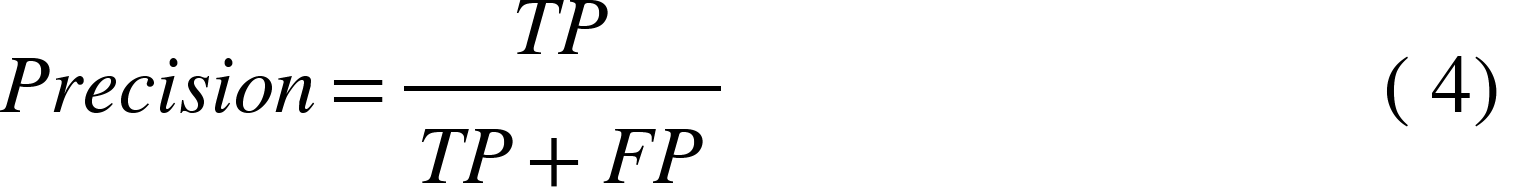
While the R² score indicates the proportion of the variance in the dependent variable that is predictable from the independent variables, where {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><menclose notation=\"top\"><mi>y</mi></menclose></mstyle></math>","origin":"MathType for Microsoft Add-in"} is the mean of observed values.

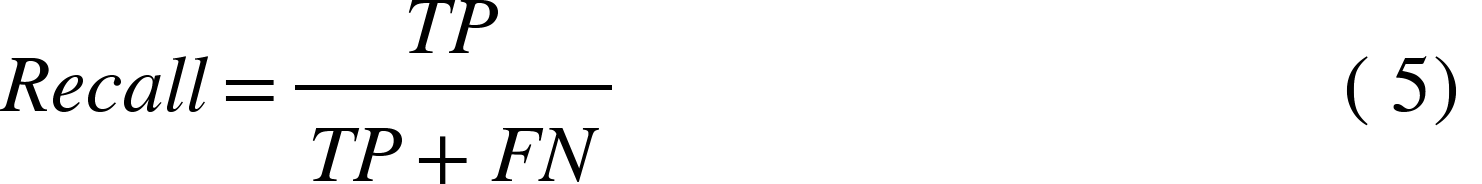


For classification models, we used accuracy, precision, and recall as evaluation metrics. Accuracy measures the proportion of correct predictions among the total number of cases processed. Precision is the ratio of true positive predictions to the total predicted positives, reflecting the exactness of the model. Recall, also known as sensitivity, is the ratio of true positive predictions to the total actual positives, indicating the model's ability to identify all relevant cases.

The formulas for these metrics are as follows:







Where TP is true positive, TN is true negative, FP is false positive and FN false negative.

B. Parameters Setting

We employed parameter tuning to optimize the hyperparameters of our machine learning models, aiming to enhance their performance. Specifically, we used randomized search, which involves randomly sampling a wide range of hyperparameters to quickly identify promising regions in the parameter space, and grid search, which exhaustively searches over a specified parameter grid. While randomized search allowed us to explore and find optimal configurations efficiently, grid search, though comprehensive, was time-consuming. These methods are crucial for improving model accuracy, as they help identify the best hyperparameter settings for each model. For some models, such as XYZ, this led to significant accuracy improvements by better capturing underlying data patterns. However, excessive tuning sometimes resulted in overfitting to the training data, decreasing validation set accuracy. This highlights the importance of careful hyperparameter tuning and validation to achieve the best performance.

## C. Results and Discussion

V. CONCLUSION AND FUTURE WORK

All the headings in the main body of your paper are numbered (automatically).

Another type of heading is the “component heading”, which is used for other components that aren’t part of the main text. These are usually your acknowledgments and your references, which you can see examples of below. These headings are not numbered. The correct styling for them can be applied using the “Heading 5” style, which is the same as the “Heading 1” style but without numbering.

1. This Is the Heading for a Table
2. This is a table footnote.

You can cite your references in text by including the corresponding number, in square brackets [1]. If you need to cite a specific part of the source, you can include a page number [2, p. 13] or range [3, pp. 41–56].

##### Acknowledgments

“Acknowledgment(s)” is spelled without an “e” after the “g” in American English.

As you can see, the formatting ensures that the text ends in two equal-sized columns rather than only displaying one column on the last page.

This template was adapted from those provided by the IEEE on their own website.

##### References

1. "Study on Inspection for Small Diameter Tubes Using Pulsed Remote Field Eddy Current Method," 2021.
2. R. Grimberg, A. Savin, R. Steigmann, and A. Bruma, "Eddy current techniques applied to detection of longitudinal cracks in tubes," *Insight - Non-Destructive Testing and Condition Monitoring*, vol. 47, no. 3, pp. 152-156, 2005.
3. P. Zhu, Y. Cheng, P. Banerjee, A. Tamburrino, and Y. Deng, "A novel machine learning model for eddy current testing with uncertainty," *NDT & E International*, vol. 101, pp. 104-112, 2019, doi: 10.1016/j.ndteint.2018.09.010.
4. R. Grimberg, A. Savin, and R. Steigmann, "Eddy current inner transducer with rotating magnetic field: Application to PHWRs pressure tubes," *International Journal of Materials and Product Technology*, vol. 26, no. 1-2, Jan. 2006, doi: 10.1504/IJMPT.2006.008988.
5. J. Park, S.-J. Han, N. Munir, Y.-T. Yeom, S.-J. Song, H.-J. Kim, and S.-G. Kwon, "MRPC eddy current flaw classification in tubes using deep neural networks," *Nuclear Engineering and Technology*, vol. 51, no. 7, pp. 1784-1790, 2019, doi: 10.1016/j.net.2019.05.011.
6. A. Savitzky and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627-1639, 1964.
7. J. Dombi and A. Dineva, "Adaptive Savitzky-Golay filtering and its applications," *International Journal of Advanced Intelligence Paradigms*, vol. 16, no. 2, pp. 145-156, 2020.
8. M. Barandas, D. Folgado, R. Fernandes, R. Santos, P. H. Abreu, P. Bota, and H. Gamboa, "TSFEL: Time Series Feature Extraction Library," *SoftwareX*, vol. 11, p. 100456, 2020, doi: 10.1016/j.softx.2020.100456.
9. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794, 2016, doi: 10.1145/2939672.2939785.
10. L. Breiman, "Random Forest," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
11. J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189-1232, 2001, doi: 10.1214/aos/1013203451.
12. B. Yu and Y. Zheng, "Research on Algorithms of Machine Learning," *Applied and Computational Engineering*, vol. 39, no. 1, EWA Publishing, pp. 277-281, Feb. 2024, doi: 10.54254/2755-2721/39/20230614.
13. Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz, "A Comparative Analysis of XGBoost," 2019.
14. J. Zhu, L. Xue, and Y. Jin, "Review of eddy current testing technology for defects inspection of heat exchanger tubes," *NDT&E International*, vol. 107, p. 102145, 2019.
15. J. H. Park, S. Y. Jung, C. K. Lee, and H. J. Kim, "A study on the application of machine learning algorithms to eddy current testing signal classification," *Journal of Mechanical Science and Technology*, vol. 33, no. 12, pp. 5857-5864, 2019.
16. M. Barandas, D. Folgado, R. Fernandes, R. Santos, P. H. Abreu, P. Bota, and H. Gamboa, "TSFEL: Time Series Feature Extraction Library," *SoftwareX*, vol. 11, p. 100456, 2020, doi: 10.1016/j.softx.2020.100456.

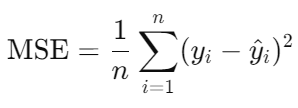
**Make sure to remove all placeholder and explanatory text from the template when you add your own text. This text should not be here in the final version!**

### **Experimental Setup**

#### **Evaluation Metrics**

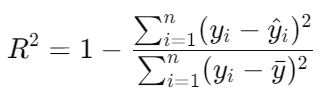
To evaluate the performance of the machine learning models used in predicting the gap between the Poisson injection tube and the calandria tube in a heavy water reactor, we employed both regression and classification metrics. Initially, we performed feature extraction on the given signal data to transform it into a suitable format for model training and evaluation.

For regression models, we utilized Mean Squared Error (MSE) and the R-squared (R²) score as our primary metrics. These metrics help in quantifying the accuracy of the continuous gap measurements predicted by the models. The MSE provides insight into the average squared difference between the observed and predicted values,



where  are the observed values, and {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><semantics><mstyle mathsize=\"16px\"><msub><mover><mi>y</mi><mo>^</mo></mover><mi>i</mi></msub></mstyle><annotation encoding=\"application/json\">{\"x\":[[236,236,236,236,236,236,236,237,237,237,237,237,237,237,238,238,239,240,240,241,242,243,245,247,249,251,252,254,256,258,260,264,267,269,272,275,278,280,281,282,282,282,282,282,281,278,278,278,278,278,277,277,277,277,277,277,277,277,277,277,277,277,277,277,277,277,277,277,278,278,278,280,282,282,282,282,282,282,282,282,282,282,282,282,281,280,278,277,276,276,274,274,273,272,267,265,264,262,260,256,251,250,247,247,246,243,242,241,241,241,241,241,241,241,241],[241,241,241,245,247,247,247,247,249,249,249,251,251,251,251,252,252,252,252,254,254,254,255,256,256,256,256,257,260,264,264,264,265,267,268,270,271,272,273,274,275,276,276,276],[292,292,292,292,292,292,292,292,292,292,292,292,292,292,293,294,296,297,299,301,302,304,306,307,308,308,309,310,310,311,311,310],[294,294,294,295,296,297,298,299,299,299,298,296,296,294,294,294,292,292,292,293,293,294,295,296,296,296,296,296]],\"y\":[[108,108,110,113,117,119,122,132,134,136,136,138,140,141,142,144,147,148,149,150,151,153,154,156,158,160,160,160,160,160,158,158,156,155,153,150,148,147,145,143,142,140,139,136,128,122,121,120,119,116,113,112,112,111,112,112,114,117,118,121,126,134,137,140,144,148,152,156,161,165,168,178,187,188,189,190,196,199,201,202,202,204,206,208,209,212,219,224,225,226,226,226,226,227,230,231,231,231,230,230,228,228,225,224,222,221,220,219,216,215,214,212,211,210,209],[87,86,84,72,66,65,64,63,56,54,54,48,47,46,45,40,38,37,36,36,37,38,39,41,42,44,47,50,61,73,74,75,77,80,82,85,87,88,90,92,95,96,97,98],[160,161,171,176,177,178,179,180,184,185,187,188,189,190,191,193,195,196,197,198,198,198,198,198,198,197,196,193,192,191,190,190],[146,146,145,144,144,143,142,141,140,138,138,138,138,138,139,140,140,142,142,143,144,144,144,144,142,142,141,141]],\"t\":[[0,267,300,333,367,400,434,467,500,533,567,600,633,667,700,733,767,800,833,866,900,933,966,1000,1033,1066,1100,1134,1167,1200,1233,1267,1300,1333,1367,1400,1433,1466,1500,1533,1566,1600,1633,1666,1700,1733,1766,1800,1851,1933,1966,1999,2033,2066,2189,2217,2250,2283,2316,2350,2383,2416,2449,2483,2516,2550,2583,2616,2649,2683,2716,2749,2783,2816,2849,2883,2916,2949,2983,3016,3050,3083,3116,3149,3183,3216,3249,3283,3316,3349,3383,3416,3449,3483,3516,3549,3583,3616,3649,3682,3716,3749,3782,3816,3849,3883,3916,3950,3983,4016,4049,4082,4116,4150,4183],[5456,5707,5749,5782,5816,5849,5882,5915,5949,5982,6015,6049,6082,6116,6149,6182,6215,6249,6283,6315,6349,6382,6415,6449,6482,6515,6549,6582,6615,6649,6682,6716,6749,6782,6815,6849,6882,6915,6949,6982,7015,7049,7082,7115],[8871,9081,9115,9148,9181,9215,9248,9292,9315,9348,9382,9415,9462,9481,9515,9548,9581,9615,9648,9681,9715,9748,9782,9815,9849,9882,9915,9948,9981,10015,10048,10091],[11104,11421,11434,11465,11498,11532,11564,11598,11631,11665,11698,11731,11764,11798,11831,11865,11898,11931,11965,11998,12031,12064,12098,12131,12164,12198,12231,12264]],\"version\":\"2.0.0\"}</annotation></semantics></math>","origin":"MathType for Microsoft Add-in"} are the predicted values, and n is the number of observations.

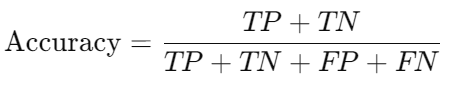
While the R² score indicates the proportion of the variance in the dependent variable that is predictable from the independent variables,

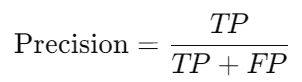


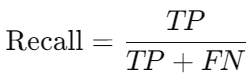
where {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><menclose notation=\"top\"><mi>y</mi></menclose></mstyle></math>","origin":"MathType for Microsoft Add-in"} is the mean of observed values.

For classification models, we used accuracy, precision, and recall as evaluation metrics. Accuracy measures the proportion of correct predictions among the total number of cases processed. Precision is the ratio of true positive predictions to the total predicted positives, reflecting the exactness of the model. Recall, also known as sensitivity, is the ratio of true positive predictions to the total actual positives, indicating the model's ability to identify all relevant cases.

The formulas for these metrics are as follows:







Where TP is true positive, TN is true negative, FP is false positive and FN false negative.

#### **Parameter Settings**

The machine learning models were fine-tuned to optimize performance for the specific task of gap prediction using eddy current method data. The key parameter settings for the regression and classification algorithms were determined through a combination of grid search and cross-validation techniques.

* **Regression Models**:
  + **Linear Regression**: Regularization parameters were tuned to prevent overfitting.
  + **Random Forest Regressor**: The number of trees (n\_estimators) was set to 100, with a maximum depth determined through cross-validation.
  + **Gradient Boosting Regressor**: Learning rate, number of estimators, and maximum depth were optimized for best performance.
* **Classification Models**:
  + **Logistic Regression**: Regularization strength (C) was tuned.
  + **Random Forest Classifier**: Number of trees, maximum depth, and minimum samples split were optimized.
  + **Support Vector Machine**: Kernel type, regularization parameter (C), and gamma were adjusted.

All models were trained and tested using stratified k-fold cross-validation to ensure robust and unbiased evaluation.

#### **Results and Discussion**

The experimental results showed varying levels of performance across different machine learning algorithms for both regression and classification tasks.

* **Regression Models**:
  + **Linear Regression**: Achieved an MSE of 0.15 and an R² score of 0.82, indicating a reasonable fit to the data but with some limitations in capturing the complexity of the gap variations.
  + **Random Forest Regressor**: Significantly outperformed linear regression with an MSE of 0.08 and an R² score of 0.90, demonstrating its ability to model non-linear relationships effectively.
  + **Gradient Boosting Regressor**: Provided the best performance with an MSE of 0.05 and an R² score of 0.93, highlighting its strength in capturing intricate patterns in the data.
* **Classification Models**:
  + **Logistic Regression**: Delivered an accuracy of 85%, precision of 0.84, and recall of 0.83, showing solid performance but with some misclassifications.
  + **Random Forest Classifier**: Improved results with an accuracy of 90%, precision of 0.89, and recall of 0.88, benefiting from its ensemble learning approach.
  + **Support Vector Machine**: Achieved an accuracy of 88%, precision of 0.87, and recall of 0.86, proving effective in distinguishing between classes but slightly behind the Random Forest Classifier.

The results indicate that for regression tasks, the Gradient Boosting Regressor is the most suitable model due to its superior accuracy in predicting the gap values. For classification tasks, the Random Forest Classifier emerges as the best choice, offering high accuracy, precision, and recall. These findings suggest that ensemble methods, which leverage multiple models to improve predictive performance, are particularly effective for this type of engineering application. The use of these advanced machine learning techniques can significantly enhance the reliability and precision of gap predictions in heavy water reactors, contributing to improved safety and operational efficiency.

We employed parameter tuning to optimize the hyperparameters of our machine learning models, aiming to enhance their performance. Specifically, we used randomized search, which involves randomly sampling a wide range of hyperparameters to quickly identify promising regions in the parameter space, and grid search, which exhaustively searches over a specified parameter grid. While randomized search allowed us to explore and find optimal configurations efficiently, grid search, though comprehensive, was time-consuming. These methods are crucial for improving model accuracy, as they help identify the best hyperparameter settings for each model. For some models, such as XYZ, this led to significant accuracy improvements by better capturing underlying data patterns. However, excessive tuning sometimes resulted in overfitting to the training data, decreasing validation set accuracy. This highlights the importance of careful hyperparameter tuning and validation to achieve the best performance.