IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 6.252
Volume 2, Issue 8, May 2022

Development of Data Analysis Software to Identify Leakage from Radiotracer Data

Chinmay Walinjkar[1], Pushkar Sutar[1], B.G Avhad[2] and P.B Walinjkar[2]
Department of Electronics, Sardar Patel Institute of Technology, Mumbai, India[1]
Bhabha Atomic Research Centre, Mumbai, India[2]

Abstract: *The radiotracer method is used to identify the leakage in the heat exchanger from a bank of heat exchangers of the Diesel Hydrotreating (DHDT) unit in the refinery. The data of the radiotracer study is analyzed using peak detection techniques to identify the leakage. This paper introduces a software solution to analyze the radiotracer data. The paper also discusses various data smoothing and peak detection techniques used in the software. This software tool was tested successfully using simulated data and verified using actual data from previous studies. The software tool correctly differentiated false peaks and leakage peaks. This result was helpful for correctly identifying the leaking heat exchanger.*

Keywords: Data Wrangling, Continuous Wavelet Transform, radiotracer, leaky heat exchanger.

I. INTRODUCTION

A heat exchanger is one of the most commonly used equipment in Diesel Hydrotreating (DHDT) units [1]. Generally, the DHDT unit consists of shell and tube heat exchangers connected in series [2]. The occurrence of leakage in heat exchangers is a common problem faced during the operation [3]. Confirming the leak and identifying the exact location of the defect is essential to maintain the quality of the final product, avoid loss of efficiency and maintain the safe and hazard-free functioning of the system. When connected to a DHDT system, it is difficult and time-consuming to inspect every heat exchanger separately; hence it is crucial to identify the source of the leak before shutting down the part of the system and hiring a special agency for repair. Furthermore, it is difficult to pinpoint the leaking heat exchanger of the system using traditional leak detection methods [2]. Hence the radiotracer method is used to identify the exact leaking heat exchanger from the bank [1].

In a shell and tube type heat exchanger, the shell carries hot liquid at high pressure, and the tube is at a lower pressure. In the case of a leak, the liquid moves from the high-pressure side to the low-pressure side. Hence a radiotracer injected into the higher pressure side(shell) leaks into the low-pressure side(tube) [3,6]. So the radiation detectors are placed at the tube outlets, as shown in figure 1, to monitor the possible leakage [1]. Data is continuously collected by these detectors and analyzed.

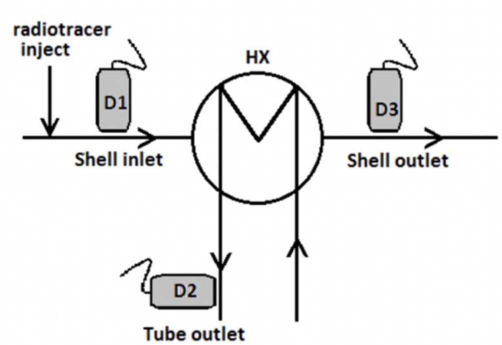

Figure 1: Principle of leakage detection

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 8, May 2022

Traditionally the data obtained from the detectors placed at the shell inlet to monitor injection, tube outlet of each heat exchanger to monitor leakage, and shell outlet to monitor exit of radiotracer, is plotted and analyzed manually by an expert. Then, the collected data is plotted graphically as radiation counts against time for analysis, as shown in Fig. 2. This process is time-consuming, and the results rely heavily on the skill and experience of an expert interpreter.

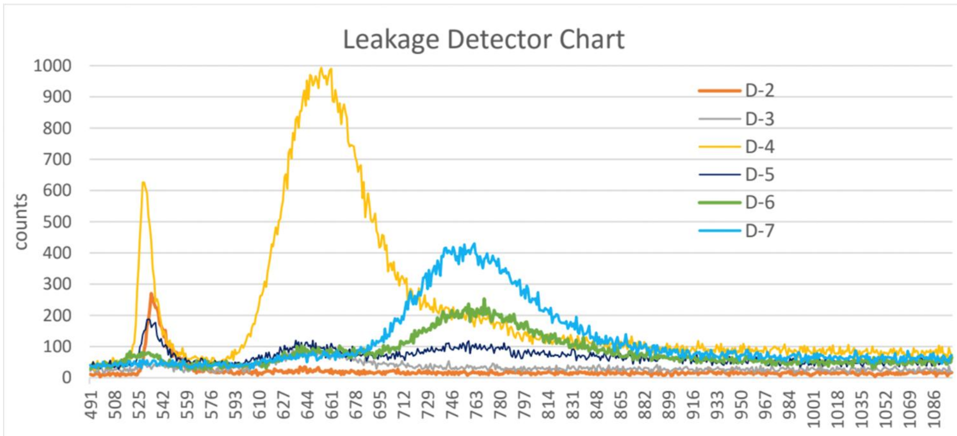

Figure 2: Sample graphical plot of radiotracer data

The main goal of this work is to develop user-friendly software to automate the data analysis and report the genuine leakage peaks from radiotracer data to help identify a leaking heat exchanger from a bank of heat exchangers. To automate and make this process more reliable, a simple software solution was developed using different peak detection algorithms to quickly report all the leakage peaks present, if any, in the respective radiotracer data.

II. Software Development

The leak detection process depends on a robust peak detection algorithm, proper data wrangling, and thresholding. The performance of various peak detection techniques depends on filtering factors such as prominence, minimum vertical and horizontal distance, peak width, and threshold [ ]. Leaks are identified as local maxima in the radiotracer counts graph. However, detecting these peaks remains challenging due to factors like background radiation, simultaneous peaks in the graph, identification of injection, and exit times of the radiotracer. The basic pipeline followed in developing software for data analysis is presented in Fig. 3.

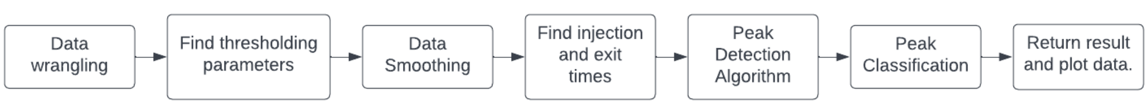

Figure 3: Basic pipeline of the software code for data analysis in leak detection

The first step is cleaning the raw data and making it usable for analysis. The raw text data obtained from the data acquisition system (MIDAS) is converted to a suitable format(.csv). Appropriate column names are given, and the null values are dropped. This file is later used as a data frame for analysis. The user can also save the file into the local system for further reference. The file is opened using the "Open File" button provided on GUI as shown in Fig. 4. The GUI is divided into three sections; the first is for data smoothing, the second is for peak detection technique, and the third section gives results.

Data is smoothened to remove low-resolution peaks and avoid false peaks. Therefore, this step is taken before any peak detection algorithm is used. Moving Averages, exponentially weighted averages, Savitzky Golay Filter, and Local Regression are the Smoothing techniques provided to the user in the software, as shown in figure 4. The performances

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 8, May 2022

of these smoothing techniques were analyzed, and the results are given in Table 1 below. After data smoothening, the data is plotted as a graph and displayed in the bottom portion of the GUI. It shows all the peaks present in the dataset.

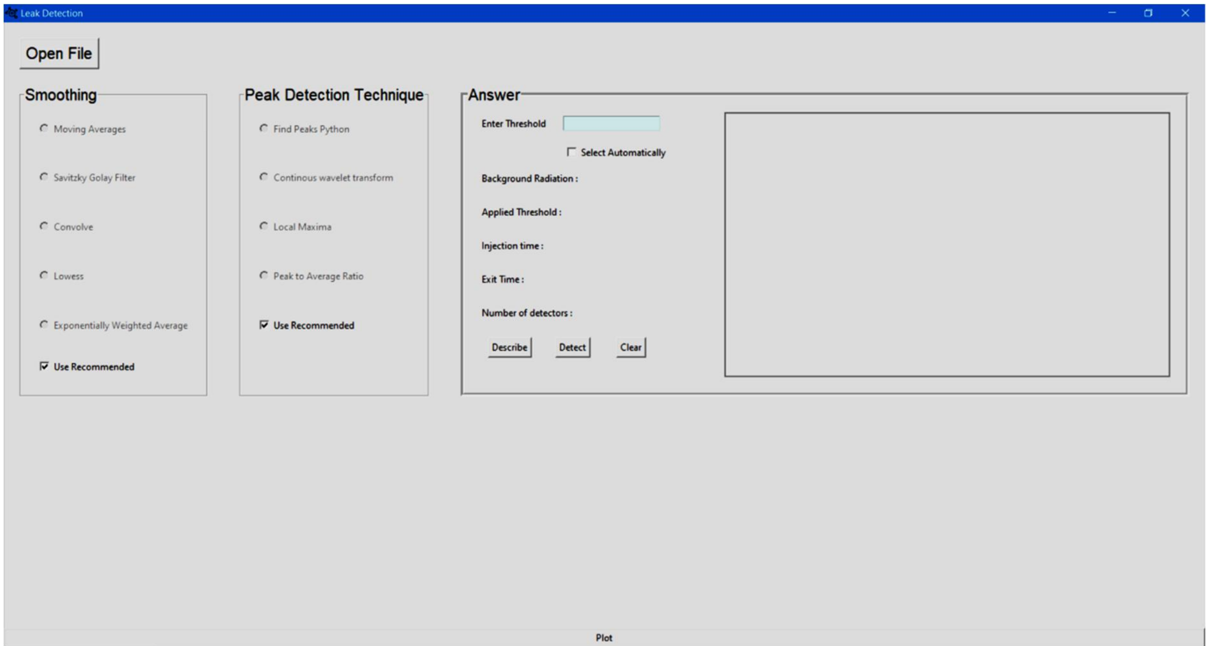

Figure 4: GUI of the software

The second section gives the user options to select the appropriate peak detection technique, as shown in Fig. 4. The peak detection algorithm is implemented using four techniques that take into account minimum vertical and horizontal distance between peaks, a threshold to identify real peaks from false peaks, peak prominence, and peak amplitude to discard false-positive results. In Continuous Wavelet Transformation, wavelets of different widths are used to identify all the true positive peaks correctly. This method works well even if the data is not smoothened. However, this method works comparatively slower as mathematical computations are performed over the vector multiple times using different wavelet widths. In the Peak to Average Ratio method, the peak is identified in three steps, first by finding the root mean square value of the signal, then by calculating the peak to average ratio, and finally, discarding small peaks based on the threshold value. The time at which a particular peak occurs is obtained by simply iterating over the data. In the Conditional Local Maxima technique, a peak is defined as the local maximum of N neighboring points. Thus, local maxima is found over a one-dimensional data array by comparing N neighboring points. ‘Find peaks’ technique is also based on the same principle. An open-source Python library was used to implement the Find peaks technique.

Finally, the third section gives the results. The left-hand side shows the calculated threshold value, injection and exit time, and the number of leakage detectors used. It also gives the average value of background radiation calculated from the data. A user can enter a threshold value manually if not satisfied with the automatically calculated threshold value. The leakage peaks found after analyzing the data are shown in the right-hand top corner. In the right-hand bottom corner, the status of each peak detected with its peak time is displayed. It shows all those peaks detected by the algorithm as false peaks and discarded. A simple methodology is applied to identify the false peak; the peaks appearing before injection peak time and after the exit peak time are considered false peaks. Further peaks appearing simultaneously and peaks with amplitude less than the threshold value are also identified as false peaks. The ‘Describe’ button calculates some statistical data like percentile, mean, standard deviation, minimum and maximum values for each column and displays it in a new window.

Multiple datasets were statistically generated to test the developed software, covering the cases as mentioned previously. These datasets were tested on the software, and the results were obtained. The software is tested using one such dataset with all the corner cases taken here for discussion. After satisfactory test results, the software was validated using actual data of the previously-formed radiotracer study.

III. Results and Discussion

3.1 Performance Test

The results of the performances of the smoothing techniques are given in the table 1 below. For checking the performance, the signal-to-noise ratio (SNR) is defined as the ratio of the mean value (μ) to its standard deviation (σ). It was observed that SNR is maximum for Locally-Weighted Scatterplot Smoothing (LOWESS) technique; therefore, it is used as a default choice. In this method, data is fitted as a curve on a scatter plot where all the original data points are present.

Table 1: Analysis of Smoothing Algorithms

Smoothing Technique	Runtime (in seconds)	SNR (μ/σ)
Moving Averages	1.078	31.892
Averages using Numpy convolution	0.001	20.425
Exponentially weighted averages	0.038	22.351
Savitzky Golay filter	0.584	33.871
LOWESS	1.318	44.009
Raw data	0.00	5.015

3.2 Test Results

The result of one of the simulated datasets in which all three corner cases were simulated is shown in figure 5.

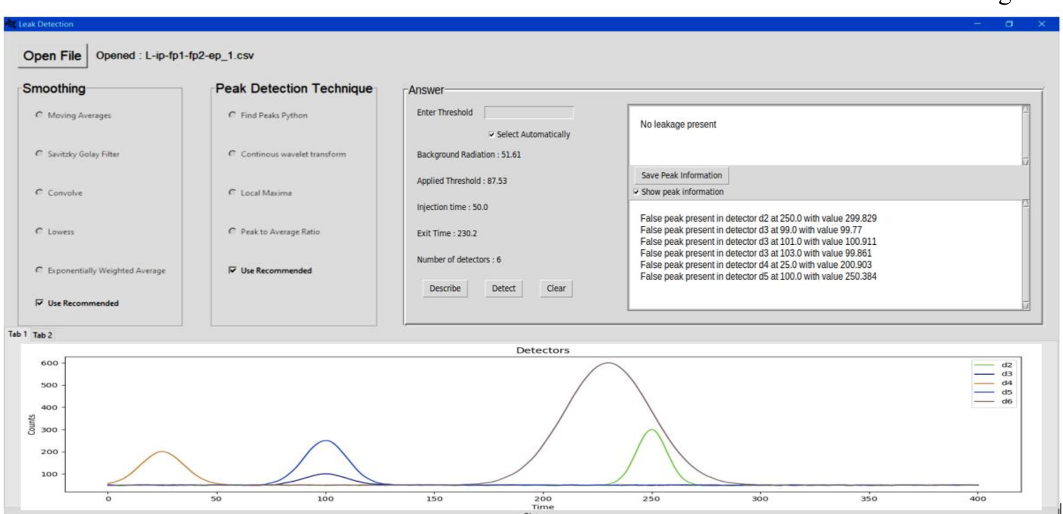

Figure 5: Software output of the test data

DOI: 10.48175/IJARSCT-4462

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 8, May 2022

In the graph section, of the Fig. 5, the graph of the injection detector is not included as the y-axis value(counts) is considerably larger. The injection peak appears at 50 seconds. In the graph, five peaks are observed. It is analyzed that the peak of d6 represents the exit peak, d2 peak appears after exit peak hence is a false peak, d3 and d5 peak occur simultaneously, hence are considered false peaks and d4 occurs before the injection peak at 25 seconds hence is considered a false peak as well. Thus, no genuine leakage peak is seen in the graph. These results are shown in the answer window of the software, as shown in Fig. 5.

The result shows that no leakage is present. The right-hand bottom corner shows the status of peaks detected with its peak time. It shows that all the peaks detected by the algorithm are false and discarded. The peaks before injection and after exit peak are discarded as false peaks. Further peaks with amplitude less than the threshold value are also identified as false peaks.

3.3 Validation Results

The software validation in automatic mode of data interpretation for peak detection using actual data of a radiotracer experiment that was carried out earlier. Its results derived using traditional methods were published in 2020 [1]. The output of the data analysis using a software tool is shown in Fig. 6. The graph shows four peaks of detectors d4, d5, d6, and d7. However, the leakage prediction result shows that leakage is only in d4, and peaks of d5, d6, and d7 are false peaks. The result is verified with the result published. It was found that both the results matched. It verifies that the software output has correct results and validates its use to analyze the experimental data for leakage identification. As the accuracy of the result depends on the application of proper smoothening and peak detection techniques on the data, the best possible is selected by default. However, options are given to the user to select various techniques to get satisfactory results.

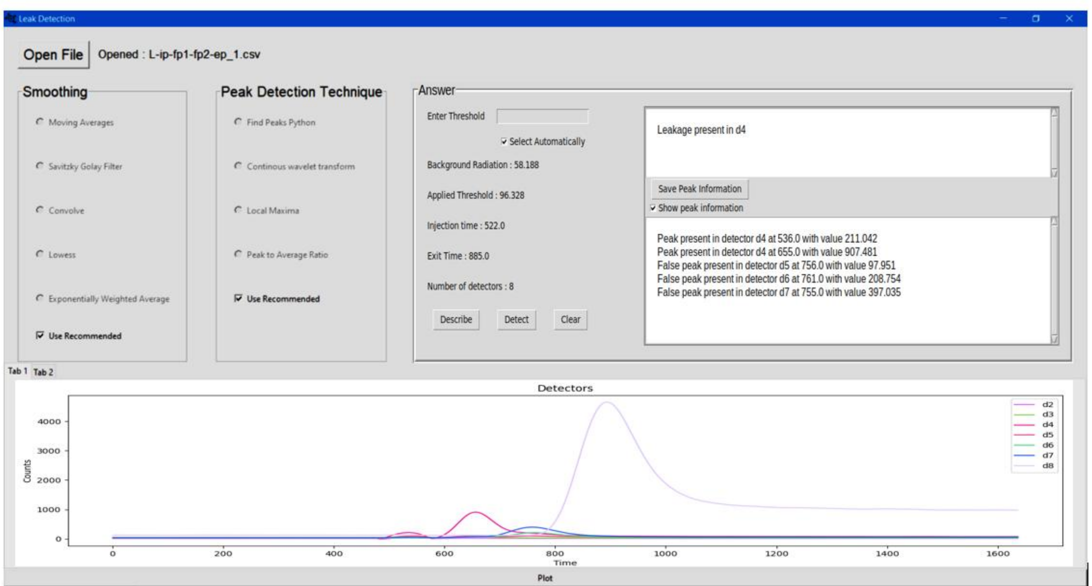

Figure 6: Result showing the genuine leakage and false peaks reported of actual data

IV. CONCLUSION

The test and validation results show that the software is successfully developed to analyze the radiotracer data to identify the leaky heat exchanger. It has met the objectives of development. It has automated the process of reporting genuine leakage and false peaks. The software enables the users to identify genuine leakage peaks. This has simplified the analysis of radiotracer data and can improve the success rate of identifying leaky heat exchangers.

V. ACKNOWLEDGMENT

The authors gratefully acknowledge Director, RC&IG, BARC for giving the opportunity to carry out the project work in BARC. The authors thank B.N Chaudhari, Principal, SPIT, Prof Deepak Karia, HoD, SPIT for providing support and motivation to carry out the developmental work.

REFERENCES

[1] P. B. Walinjkar, and V. N. Yelgaonkar, “Radiation Based Advanced Leak Detection Technique for Indian Refinery”, in IJRES 2020, vol. 8, issue 11, pp. 33-37.

[2] H. J. Pant, V.K.Sharma, Sunil Goswami, J.S.Samantray, and Gursharan Singh, “Development and Application of Radiotracer Technique for Online Leak Detection in High Pressure Heat Exchangers”, in BARC Newsletter, issue 330, Jan-Feb 2013.

[3] “Radiotracer Methods for Leak Detection in Heat Exchangers,” in Leak detection in heat exchangers and underground pipelines using radiotracers, Vienna: IAEA, 2009.

[4] C. Yang, Z. He, and W. Yu, “Comparison of public peak detection algorithms for Maldi Mass Spectrometry Data Analysis,” BMC Bioinformatics, vol. 10, no. 1, 2009.

[5] McKinney, W. & others, 2010. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference. pp. 51–56.

[6] Virtanen, P. et al., 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17, pp.261–272.

[7]Y. Tournade, “Simple, distributed software.,” Atom, 01-Nov-2015. [Online]. Available: https://blog.ytotech.com/2015/11/01/findpeaks-in-python/. [Accessed: 25-Mar-2022].