An Unsupervised Approach of Denoising Time Series Data and Detecting Non-Continuous Peaks

Shafkat Waheed[1], B. M. Raihanul Haque[1], Abir Roy[1], Mst. Shapna Akter[1], Sumit Ranjan Chakraborty[2], Shakran Hayat[2], M.R.C. Mahdy[1,2]

Department of Electrical & Computer Engineering, North South University, Dhaka Bangladesh[1]

Pi Labs Bangladesh Ltd., Eden Center, 2/1/E, Toyenbee Rd, Dhaka 1000, Bangladesh[2]

shafkat.waheed@northsouth.edu; raihanul.haque@northsouth.edu; abir.roy@northsouth.edu; shapna.akter@northsouth.edu; sumitranjan612@gmail.com; shakranhayat@gmail.com; mahdy.chowdhury@northsouth.edu

Abstract: A dataset, collected under an industrial setting; often contains a significant portion of noises. In many cases, using trivial filters is not enough to retrieve useful information i.e. accurate value without the noise. One such data is time-series sensor readings collected from moving vehicles containing fuel information. Due to the noisy dynamics and mobile environment, the sensor readings can be very noisy. Denoising of such a dataset is a prerequisite for any useful application and it has led us to develop a system that can remove noise and keep the original value and help vehicle industry, fuel station and power-plant station that require fuel. In this work, we have only considered the value of fuel level and we have come up with a unique solution to filter out the noise of high magnitudes using several algorithms such as interpolation, extrapolation, spectral clustering, agglomerative clustering, wavelet analysis and median filtering. We have also employed peak detection and peak validation algorithms to detect fuel refill and consumption in charge-discharge cycles. We have used the R-squared metric to evaluate our model and it is 98%. In most of the cases, the difference between detected value and real value remains within the range of ±1L.

1. Introduction

Data denoising or removing noise from a dataset is a challenging and fascinating topic for the researchers. Each dataset is different in terms of data distribution and noise. Dataset might be a set of images or audio signals or any other generic data. These noises are generated due to device malfunctioning, human limitation, machine limitation, improper approaches of collecting and preserving data etc.

However, in this work, the focus revolves around industrial dataset that has issues like the severe noisy environment and this dataset contains fuel level information found in a fuel tank. Whether this is a vehicle industry or fuel station or a power plant, which is being operated using fuel, they have almost similar problems. Since this is a time-series data, any given point, within a certain time frame either indicates charging or discharging; and noise present in the dataset gives a false representation of peak or consumption. Here, false representation stands for noisy data that plagues the actual value. For instance, at any given time, the value obtained from sensors could be 40 liters; whereas the actual value could be of 60 or 30 liters. Using a typical filter, while cleaning the dataset, causes data loss and fails to remove noise properly. In an attempt to reduce noise from time-series data representing the fuel volume of a vehicle, some related works have been discussed below

So far, most of the researches have been conducted on medical images [1-5]. In all these papers, the dataset comprises MRI images, X-ray images, CT scan images, PET scan images and so on. Some of the proposed methods incorporate NN (neural network), SVM (support vector machine), denoising autoencoder etc. The problem with these methods is that: in the case of a complete unsupervised scenario; all the proposed models, especially NN, perform poorly. In our case, implementing a device to a car to get the accurate real-time values has not been feasible. Therefore, NN has not been an option to deduce real value. For similar reasons, the use of LSTM (long short term memory), GAN (generative adversarial network), RNN (recurrent neural network) and other classifiers, which have been proposed by some authors [6-8], have not been possible either.

Another notable area, where denoising techniques have been implemented, is audio signals [9-12]. Audio signals are usually coming from speech, background noises or from any other automated source (electronic, electromagnetic, acoustic). The suggested methodology is heavily dependent on filters such as Kalman filter, Butter Worth, Chebyshev, Elliptical Filters and so on. In the medical sector, ECG (electrocardiogram) signals are also cleaned using filters such as low pass filter [13]. Apart from these, deep learning methodologies [9] and non-linear diffusion [12] have also been considered. We have already ruled out a deep learning technique due to the unsupervised nature of the data, which we have dealt in this work. The other problem is that: relying just on filters produces poor results due to high regularization. In contrast, our model tends to be generic, which means it will work efficiently on any vehicle where there is a presence of ambiguous noise patterns.

It is evident that: the contemporary techniques are focused on denoising images or audio signals. Although denoising techniques are applied in the realm of IoT (internet of things) by using ANN (artificial neural network) [14], ToF (Time-of-Flight) data by using GAN [15], and vibration sensor data by using TFM (time-frequency manifold) [16], none of the solutions seems suited for our dataset due to unsupervised nature of the data. In our case, we had to tackle two parallel problems; one is to minimize data loss, and another is to get the actual value in a completely unsupervised manner.

Among few works in literature, Manmohan et al. [17] have thoroughly studied and implemented the process of wavelet analysis. They have also proposed a method which includes the median filter [17]. They have suggested that: ordinary median filter indicates an improvement against other filtering techniques i.e. the mean filter. Also, two thresholding methods along with wavelet are allowed them to retrieve original features of the image effectively. Additionally, implementing median filtering is suggested in reference [18]. This inspire us to implement wavelet analysis and the median filter in our dataset.

Among many other techniques, clustering has been implemented in amplicon sequencing by Kaisa Koskinen et al. [19]. The impact of clustering is huge on the number of operational taxonomic units (OTUs). Priyam Chatterjee et al. have suggested the positive impact of clustering in data denoising as well [20]. Therefore, clustering has been considered as an imperative option for our dataset. For time series and other type of data, such as for predicting time-averaged force [21], forecasting price of the market [22-23] or classify visual pollutant from an image [24], machine learning and deep learning can also serve as highly effective tools. But for our case, due to aforementioned reasons, a different approach has been implemented.

To test the efficiency of our model, we have been provided with a specific industrial dataset. This dataset contains the charging-discharging cycle value of the moving vehicle. All the vehicles have been operative on the street of Bangladesh. Due to the poor condition of few streets in Bangladesh, noise may have been prominent in the dataset. Since, barely any work has been done in denoising time-series data containing fuel information retrieved from the industrial domain; therefore, to accomplish this, we have tried many methodologies to find out the optimal solution. Our system has started with simple interpolation, extrapolation and white noise removal technique. Here, by white noise we are referring to random noise present in a dataset whose mean value is zero and has a finite variance [25]. Then two clustering methods have been brought to isolate noise from real value. Wavelet analysis has cleaned it further through wavelet transformation and inverse transformation. Finally, median filtering has been used to give the final push. Furthermore, many custom algorithms have been used to arrange previously mentioned algorithms and compare different datasets obtained from those algorithms to narrow down the refill value.

Apart from the unorthodox noises caused by the reason mentioned above, there is also an unusual noise present in the dataset. Usually, when the engine or machine is switched off, there is no charging or discharging occurring within a certain period of time. This yields a stationary or constant set of values within that time frame. Our system has required to deal with this issue and for most of the parts, it has accomplished successfully.

At first, we have developed our model using a concatenated dataset containing information from multiple vehicles. Then, to test it further we have been provided with an additional dataset of various vehicles. Initial dataset contains over 100,000 data points, which have been narrowed down to 37 peak value. The R-squared value of the final result is 0.98 and RMSE is 1.49. Our system not only detects refill of fuel but also the consumption rate.

2. Methodology

All the data have been collected from Pi Labs Bangladesh Ltd. Initially, from 9 vehicles roughly 1,00,000 data points have been collected and used for our system. Data flow is shown in Fig.1, which demonstrates the cycle starting from raw data to processed data. It begins with employing industrial settings in a moving vehicle. Then some generic transmission occurs. Our algorithm can reduce noise from the time-series data while it is streamed from source to destination through a server.

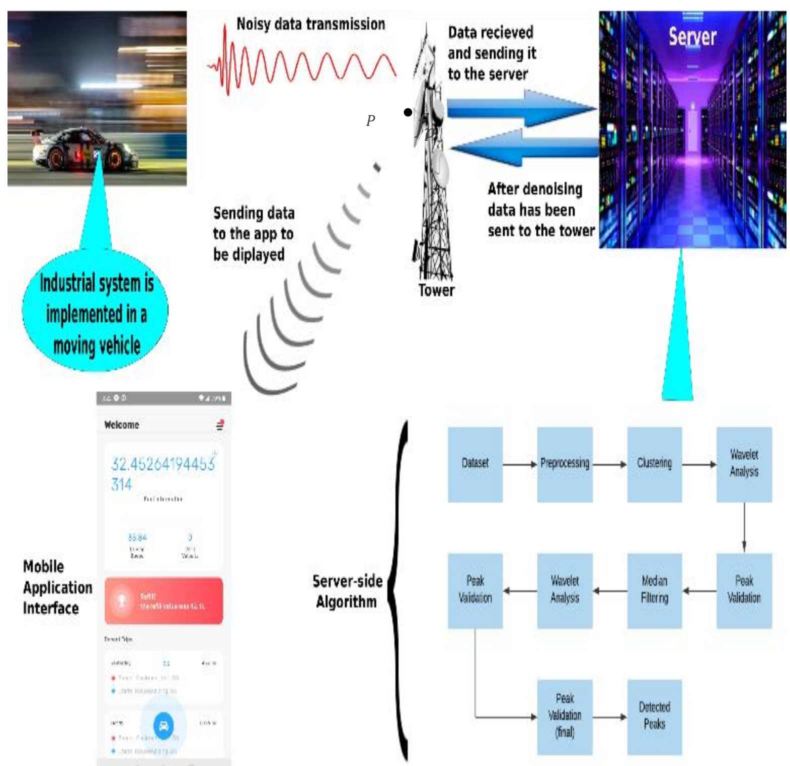

Fig. 1. *A general overview of the data cycle. The industrial setting is implanted in a vehicle and it collects and sends data to the server. Inside the server, our algorithms are implemented where the data traverse through each phase. After successful peak detection, the data is sent to the mobile application (beta version) from where the result can be displayed.*

The scope of this project revolves around the algorithms. We wanted to test whether real time data can be displayed, and this is why a mobile application prototype has been built. An enhanced overview of our algorithm is represented in Diagram.1. Our system begins cleaning the dataset with simple extrapolation and interpolation. This removes some trivial types of noises i.e. white noise. Then two set of clustering methodologies (spectral and agglomerative) isolates noise from actual value. After that, the dataset is stored and sent to the next phase to perform wavelet analysis. This generates two datasets to be compared and keep the common values. A similar strategy is undertaken for median filtering and wavelet analysis. Finally, when all prior operation is done, we have performed a final validation to get the peak values and consumption rate.

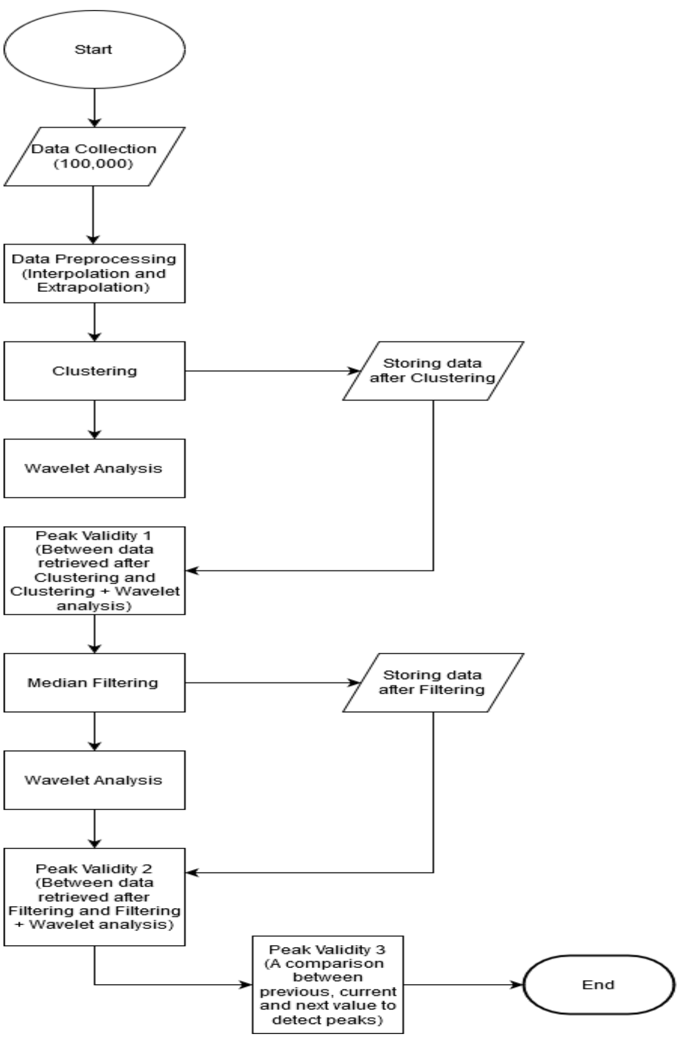

Diagram. 1. *After receiving the data it has been sent for preprocessing where we have used a few data wrangling techniques to deal with missing values and white noises. This data is then stored, and data is sent to the next phase for wavelet analysis. These two sets of data are being compared after, to eliminate redundant data points. A similar process can be observed for median filtering and wavelet transformation. Lastly, a final comparison is done to detect peaks*

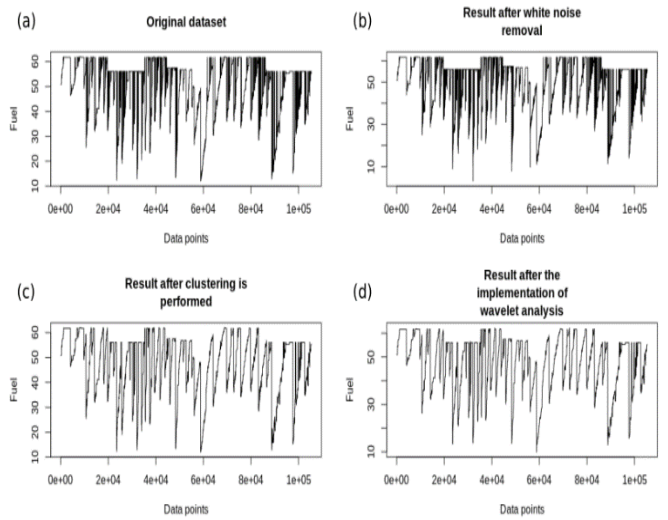

Fig. 2. *The outlook (first four steps) of our dataset after each step from initial to processed. The congested regions are the indication of data noise. (a)-(b) Raw data and data after white noise removal respectively. In both cases, heavy noise is present in the dataset. (c)-(d) Graphical representation of data after clustering, wavelet analysis (1st).*

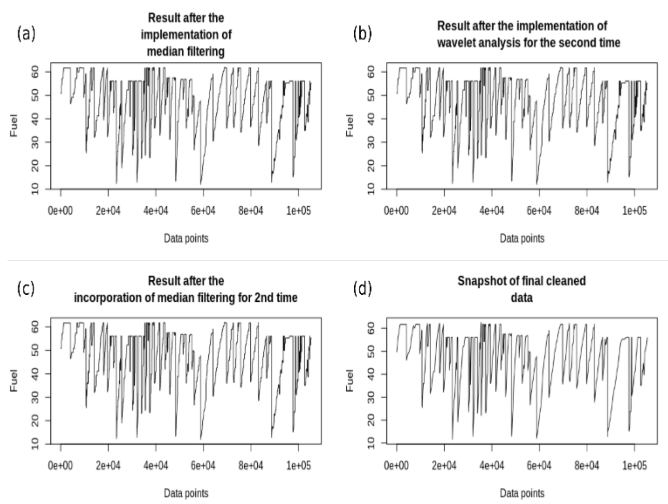

Fig. 3. *The outlook (last four steps) of our dataset after each step from initial to processed. (a)-(c) Graphical representation of data after median filtering (1st), wavelet analysis (2nd), and median filtering (2nd). A major portion of the noise is removed in each step. (d) Processed data, containing 37 peaks*

In this figure, from right to left, we see downward spikes. This slow decrease indicates the consumption rate over a certain time period. The sharp jump from the spike represents refuel. Each segment in this graph illustrates the individual vehicle's fuel level.

2.1. Data characteristics

Before designing a model, it is important to know some characteristics of the data so that similar work can be replicated, or the designed module can be implemented on a similar data set. The goal was to plot a histogram within a certain frequency.

As seen in Fig.4, there are multiple intervals and corresponding frequency where data points are grouped within a certain range.

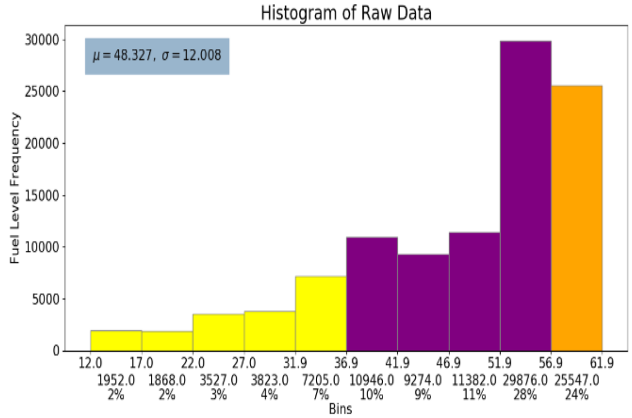

Fig. 4. *Histogram shows the distribution of the dataset*

To understand the characteristics better, we have calculated the mean and median of the data along with 1st and 3rd quartile range. Our objective was to find out whether our median is closed to 1st or 3rd quartile range. Our mean = 48.327, standard deviation = 12.008, median = 52.316, 1st quartile range = 40.790 and 3rd quartile range = 56.837. It is obvious that the median is close to 3rd quartile range and mean is less than the median. We have calculated the moments of the dataset to identify the distribution. But we could not find any specific distribution that relates to the dataset, and this is why we have undertaken an unsupervised approach to denoise the dataset.

2.2. Data Preprocessing

Initially, some rows contained zero values. This could contaminate the results of further processes and therefore data needed to be tackled in an efficient manner where data replacement conforms to the pattern of non-zero values. Preprocessing involves white noise removal using various data wrangling techniques and applying interpolation-extrapolation to recover the lost points. Also, we have differentiated the dataset to create a new feature.

Interpolation is a method that can be achieved using discrete data set and by forming general formula within a certain range. In our case, linear interpolation was used, which can be expressed as, $y = y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x - x_1)$. Here, unknown points are denoted as y for a certain value of x.

Extrapolation operates outside of the range of the observation. It is subjected to higher uncertainty, as opposed to interpolation since it fills out data based on the relation of other data points. In this research, midpoint extrapolation was incorporated, which is denoted as, $(x, y) = (\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2})$ and which is operative for one unknown point.

Interpolating and extrapolating data points were achieved through an algorithm as shown in Diagram-2. At the beginning, it is checked whether the data points contain 0 values or not. If it does so, all those values are set to NULL. Finally, through interpolation and extrapolation, new data has been generated and is being stored.

Apart from that, a general form of noise also known as random noise, has been removed from the data set. The main characteristic of the noise includes equal intensity of the signal at different frequencies. If mean equals to zero, then data is considered as white noise; as data is spread across negative and positive potion of the graph in an equal manner. After successful completion of preprocessing, we have gotten a visual representation of the data set as presented in Fig.2 (b).

2.3. Clustering

Clustering is an unsupervised learning method. It is used to group data according to their respective characteristics. Any data in a single cluster possesses a similar characteristic as every other data point in the same cluster. Right after the preprocessing is done, two clustering techniques have been incorporated. These clustering methods work together to separate noisy data from correct data.

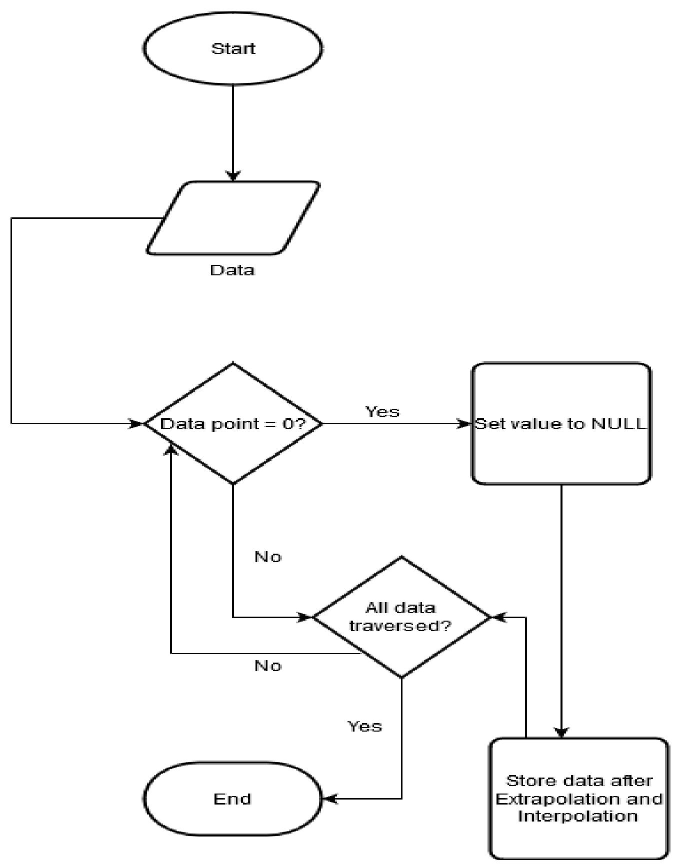

Diagram. 2. *Algorithm for data preprocessing*

Spectral clustering [26] is best suited when the variation of data is not much. It clusters the data based on the density of data points. The state of being close proximity to each other is called affinity and this phenomenon can be described by affinity matrix. Different vectors from this matrix can be extracted using principal component analysis (PCA), which later leads Eigenvectors to be formed. These vectors are referred to as feature vectors of each object of affinity or Laplacian matrix.

Hierarchical clustering [27], also known as Agglomerative clustering, generates a tree where roots are the lowest point of the data-set and leaf nodes consist of values that are greater than the values of the root node. It uses the bottom-up approach to group the data points in a hierarchical manner. Every data point groups together in its own cluster. This goes on for all data points and these clusters are then joined using a greedy approach. The greedy approach involves merging two most similar clusters together.

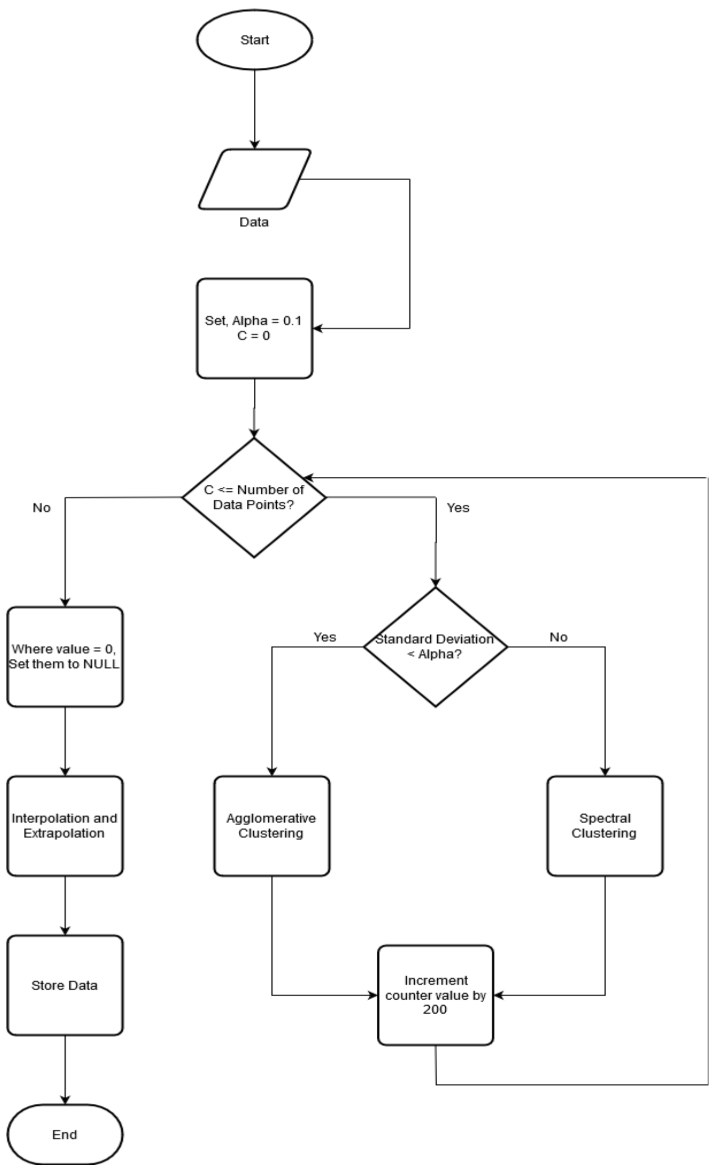

Diagram. 3. *Mechanism of clustering. Two different clustering techniques are included in this whole process*

At the start of clustering, we set a constant value denoted as alpha whose value is 0.1. We also set the counter to 0. Then standard deviation is being calculated based on all the data points. As agglomerative clustering uses a dendrogram to cluster that data it works well on the dataset having higher deviational value. As opposed to that, spectral clustering checks on the affinity of the data points to group them that is why it is better suited for dataset having lower deviational value. Due to variation of standard deviation after each iteration we used two clustering. If alpha is greater than standard deviation then, data points are grouped using Agglomerative clustering and if it is less then, data are grouped together using spectral clustering. Each iterative step is incremented by 200. After clustering, many data points get eliminated as part of the noise removal technique. Interpolation and extrapolation are used to fill out those data points. Diagram-3 illustrates the mechanism of clustering. Fig.2 (c) is a visual representation of the effect of clustering on the data set.

2.4. Wavelet Analysis

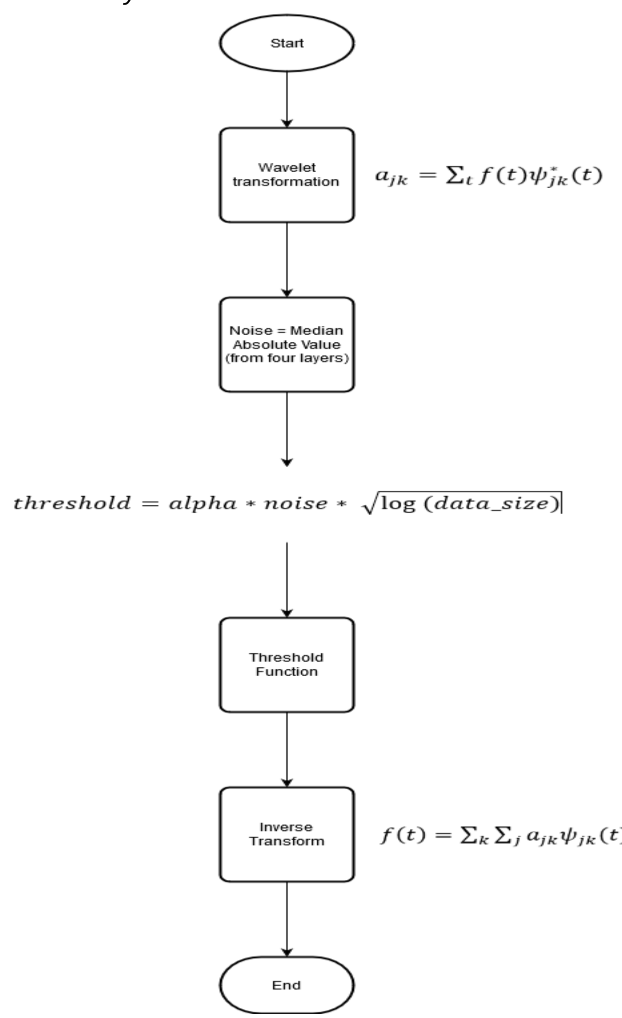

Diagram. 4. *An illustration of wavelet analysis containing two transformations*

Wavelet transformation, unlike any other Fourier transformation methodology, is used for transforming data represented in the time domain to the frequency domain [28-30]. In previous work, we have found that it has the ability to compress an image efficiently. By managing factors like shifting and scaling, it can decompose an image to multiple lower resolution images. The waves have features like varying frequency, limited duration and zero average value. This is also eligible to remove high-frequency noise from any time-series data with non-continuous peaks. The implementation of wavelets revolves around implementing two different transformation and incorporating one threshold function. These transformations are wavelet transformation and inverse wavelet transformation. The wavelet transformation is achieved by the following formula, $C(\tau, s) = \frac{1}{\sqrt{s}} \int_t f(t)\psi^*(\frac{t-\tau}{s})dt$.

Above is the formula for continuous wavelet transformation where τ and s are transition parameter and scale parameter respectively, $\frac{1}{\sqrt{s}}$ is normalization constant and $\int_t f(t)\psi^*(\frac{t-\tau}{s})dt$ is the mother wavelet. Inverse operation is carried out by the following function: $f(t) = \frac{1}{\sqrt{s}} \int_\tau \int_s C(\tau, s)\psi(\frac{t-\tau}{s})d\tau ds$.

Discrete wavelet transformation is a bit straight-forward than this and, to state the obvious, free from integral operation. The formula for discrete wavelet transformation is, $a_{jk} = \sum_t f(t)\psi_{jk}^*(t)$ and inverse discrete wavelet transformation can be written as such, $f(t) = \sum_k \sum_j a_{jk}\psi_{jk}(t)$. These formulas provide simultaneous localization in time and scale, sparsity, adaptability and linear time complexity; which allow noise filtering, image compression, image fusion, recognition, image matching and retrieval efficiently. Finally, an additional threshold function can be represented as such to improve the proposed model, $threshold = alpha * noise * \sqrt{data\_size}$ and in this formula, alpha is a constant and noise = absolute median value. We incorporated the discrete approach in our model.

This reserves the original signal but without the noise. After the implementation, the peak value was shifted by 5200 points. Diagram-4 provides the mechanism and Fig.2 (d) and Fig.3 (b) show the result after wavelet analysis.

2.5. Median filtering

Median filtering is a non-linear filtering technique and it removes noise from the data set while preserving the valuable data [31]. In our case, the median filter is to traverse through the signal one by one. It generates a window to slide through the data entry and this window is generated based on the pattern of neighboring window. This replaces each entry with the median of neighboring entries and what remains is the peak. Fig.3 (a) and (b) show results after median filtering applied for the first time on the data set. Our data has one dimension. A window of the median filter contains the first few preceding and following entries. Diagram-5 shows the algorithm of this traversal and filtering technique.

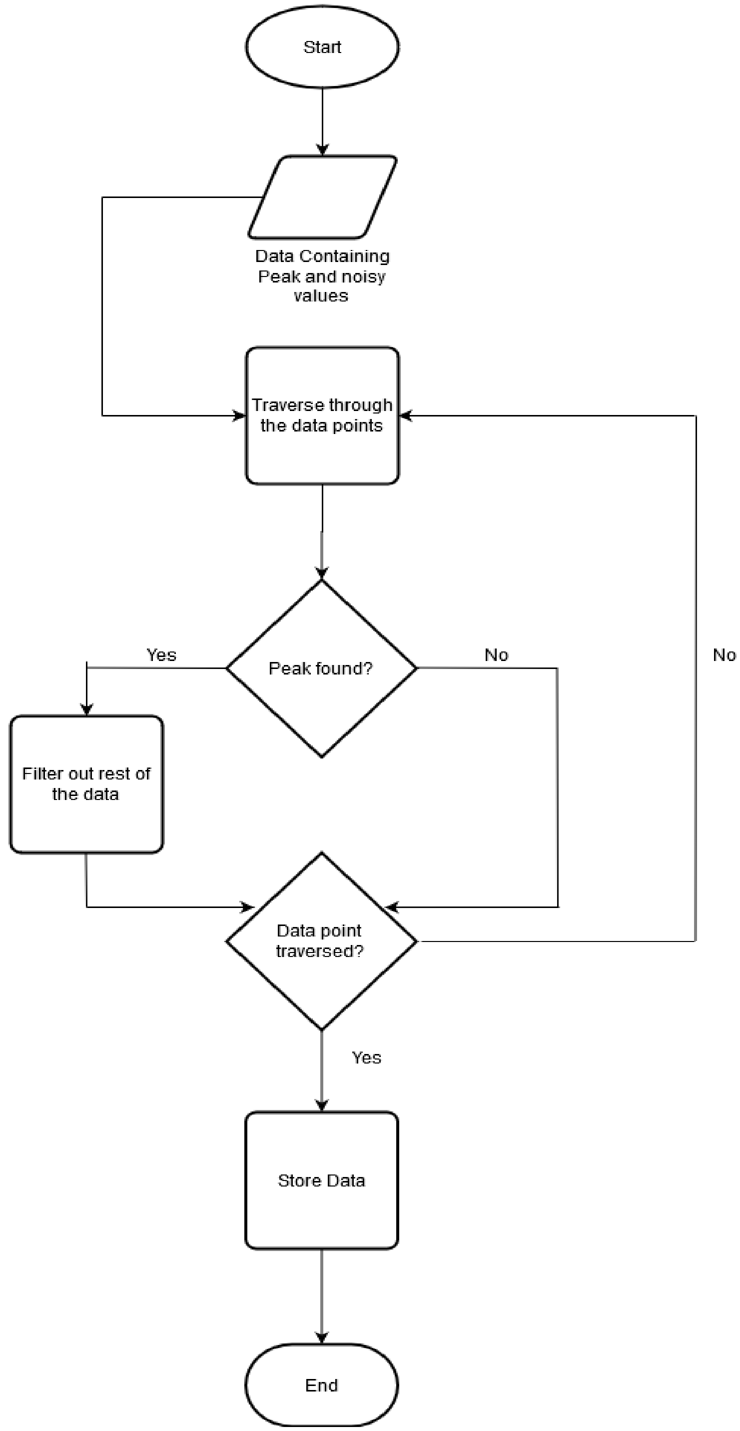

Diagram. 5. *A median fileting approach for our dataset*

Once what algorithm to use is sorted out, we have been required to write our own custom algorithms where we can use clustering, filtering methodologies efficiently. It is necessary to choose data window and iterative steps carefully so that we can minimize the data loss. Our custom algorithms include one peak detection and three peak validation methods.

2.6. Peak detection

This is the part where peaks are detected and sorted. The algorithm works as follows, as defined in Diagram-6 while traversing through the data points it checks the difference of two immediate data points. If the difference is less than 3; then they are grouped together, within the range of 100 data points by their index value. Then it evaluates the highest and lowest points surrounding the peak and ignores the rest of the data points. This allows the system to pluck out the peaks easily.

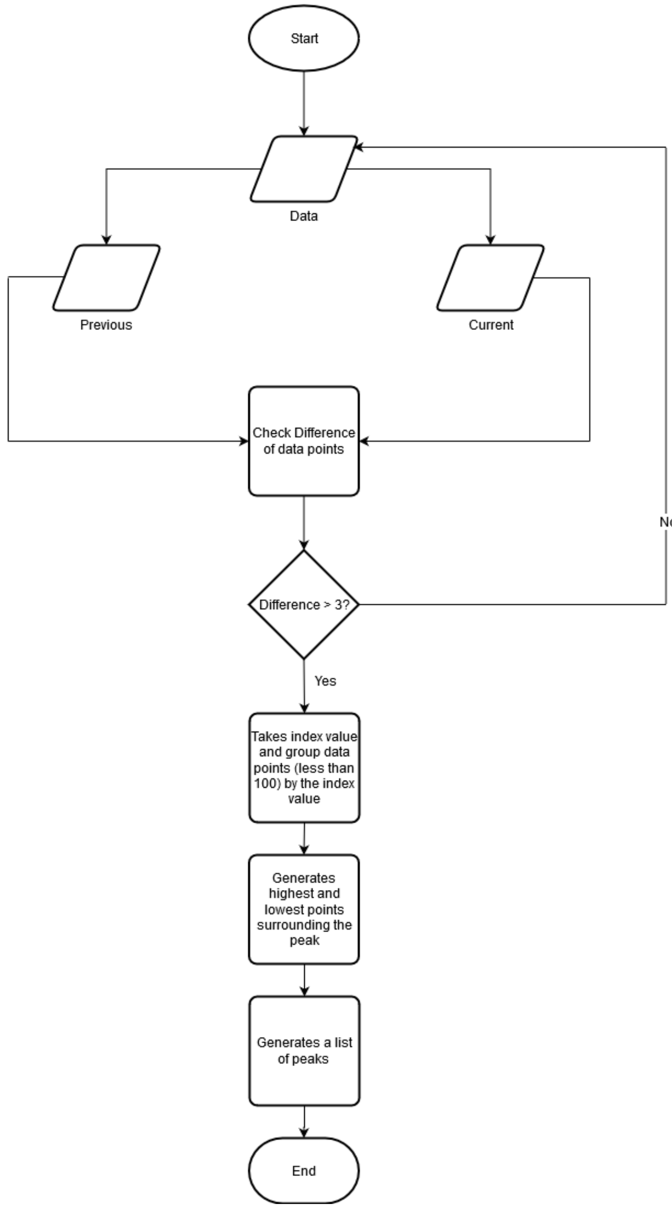

Diagram. 6. *Peak detection algorithm which checks differences between data points and based on this decision it chooses peak values*

There are three peak validation methods in this system among which two are almost identical. The prior two peak validation methods include the following:

2.7. Peak validation (first)

At the start, peak validation data is sent to two different directions. One part is operative using clustering and peak detection algorithm. Another part is operative using a combination of clustering, wavelet analysis and peak detection algorithm.

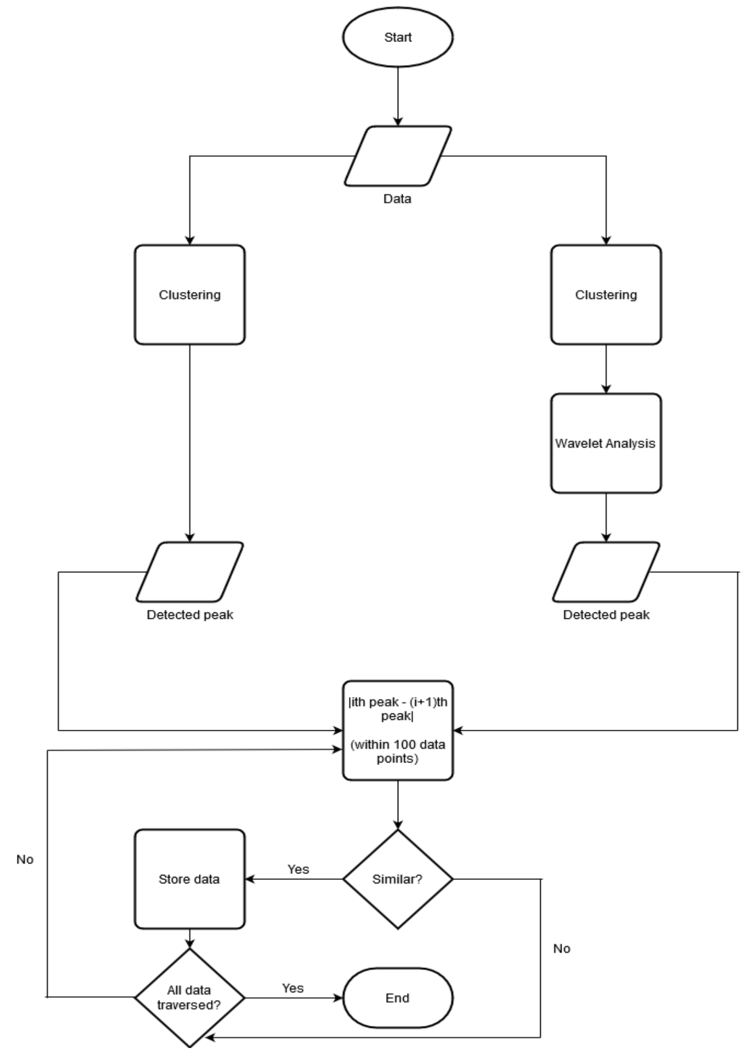

Diagram. 7. *First peak validation algorithm It is a comparison between values obtained by only after clustering and a combination of clustering and wavelet. Data is sorted based on similarity*

This produces two different datasets. After that, each entry of one dataset is evaluated against the corresponding entry of other data set. If there is a similarity, we validate the peaks otherwise we discard it. This parallel processing gets the work done fast without interrupting each other. Diagram-7 demonstrates the algorithm.

2.8. Peak validation (second)

As mentioned earlier both validation methods work similarly. The only difference is that, in second peak validation, median filtering has been used instead of clustering. This is explained in the Diagram-8.

As part of the peak validation, the effect of wavelet transformation and median filtering, which is implemented for the second time, can be found in Fig.3 (b) and Fig.3 (c) respectively.

2.9. Peak validation (final)

This peak validation method is operative on the final data set before producing the final result. Three consecutive data points, previous = i-1, current = i and next = i+1, are considered for data traversal.

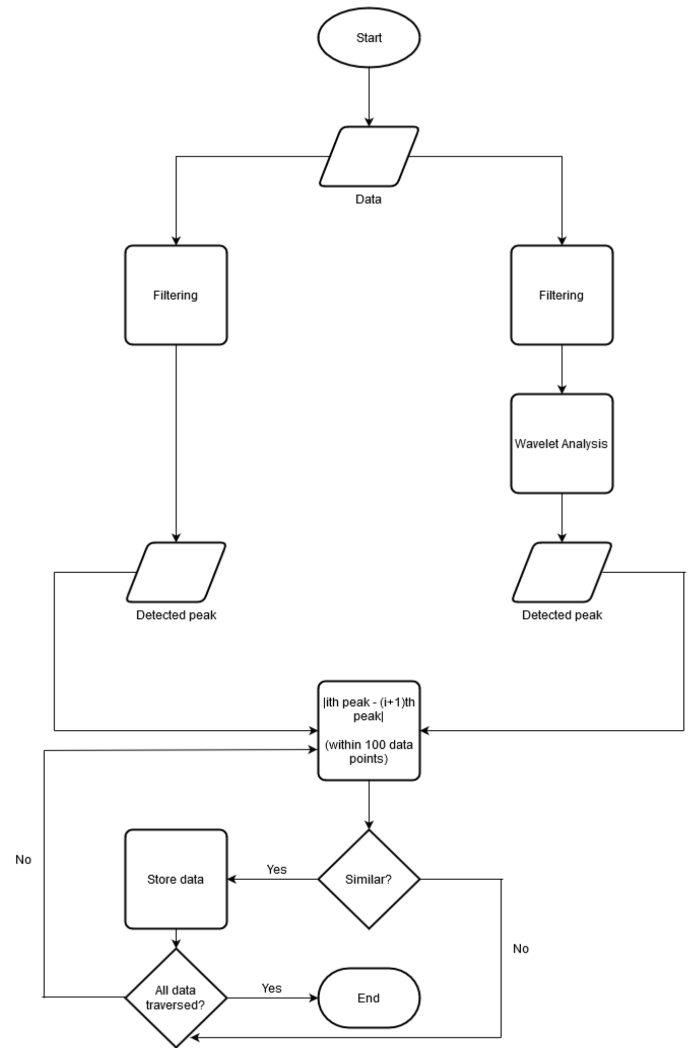

Diagram. 8. *Second peak validation algorithm It is a comparison between two datasets. One is obtained after performing just filtering and another is acquired through implementing a combined system of filtering and wavelet. Data is sorted based on similarity.*

We have defined two variables such as, A = previous – current and B = current – next, to check the distance and difference of A and B. Distance refers to the position between A and B and difference is the value obtained by subtracting A and B. If the distance is less than 30 and the difference is less than 5, then the value is removed from the peak. This final procedure isolates the peak from rest of the data to have proper evaluation on peak detection. After this peak validation, we obtain 37 peak points. After getting the peaks, we can easily calculate the consumption rate by finding the difference between peaks.

3. Result and Analysis

Due to hardware malfunction or some similar reason, at different time periods, the obtained data showed bizarre patterns; which has caused discrepancy in finding the correct result. In the highway, due to the high speed, the fuel

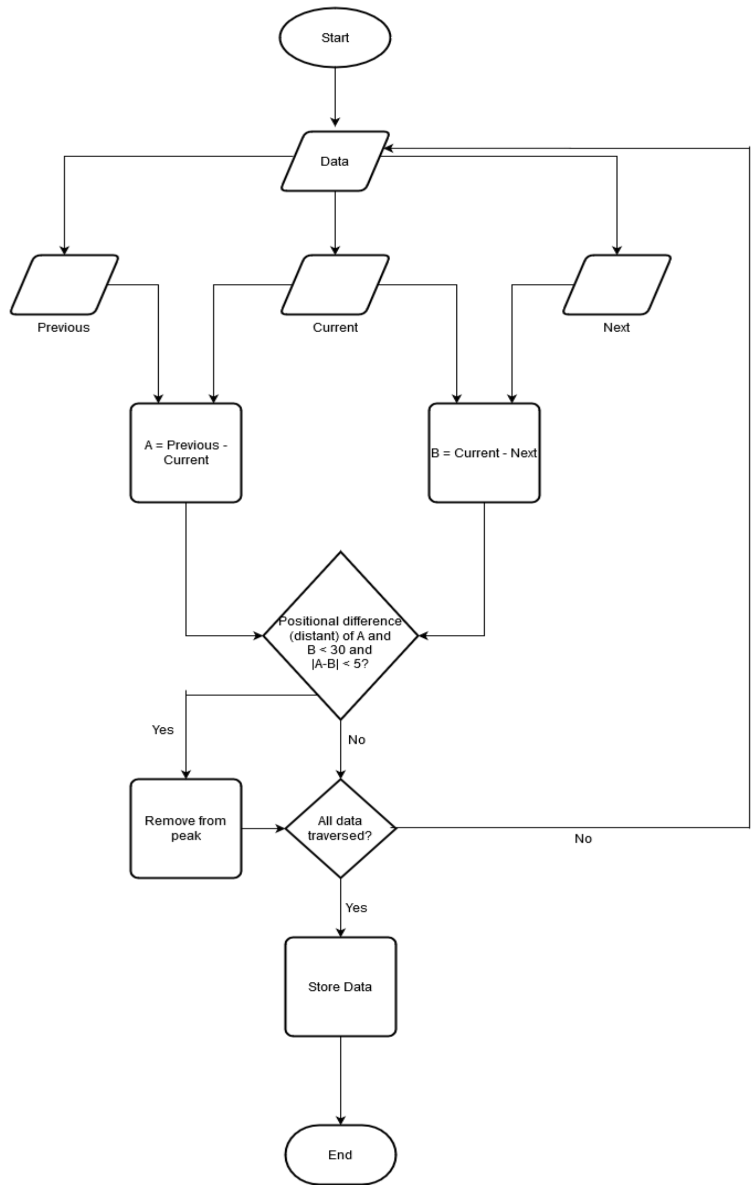

Diagram. 9. *Third and final peak validation method. It is a comparison between three adjacent data points. This exerts the final 37 data points, which correspond to the peak value*

consumption rate remains constant, as the device has its limitation. As a result, there is a huge data loss during this time period. If the dataset collected have no loss of data, then the result can be significantly improved where the accuracy may exceed ours. This final procedure isolates the peak from rest of the data to have a proper evaluation on peak detection as illustrated in Diagram-9.

As seen in Table-1, for the most part, our module can detect peaks with an error of $\pm 1L$. For those values, whose error rate is higher than 5 liters is due to the reason mentioned above. However, the R-squared score is 0.98 and RMSE is 1.49, which indicates that our module is highly efficient. The graphical representation of cleaned data after all noise removal can be seen in Fig.3 (d).

Table 1 List of detected peaks as compared to the real value. For the most part, the error is minimum.

Start_index_refill	Stop_index_refill	Deduced_value	Real_value	Error	Percentage_error
4041	4042	17.75168437	17.77	0.018315629965759	0.103070511906353
9709	9710	14.64695647	14.65	0.003043528614741	0.02077493934977
10663	10664	35.29107497	36.89	1.59892502522143	4.33430475798708
12938	12939	10.87529252	9.8	1.07529251780108	10.9723726306232
14083	14084	31.3662363	33.25	1.88376369997919	5.66545473677951
18052	18163	23.66448841	23.54	0.124488411240215	0.528837770774064
19687	19783	29.39111195	29.202177	0.188934954874462	0.646989280540495
22121	22122	14.69663734	14.66	0.036637341635856	0.24991365372344
23412	23413	46.69013676	47.14	0.449863239609741	0.954313193911202
25657	25658	35.89892004	36.8	0.901079957266589	2.44858684039834
30112	30113	32.85478607	32.79	0.064786069970658	0.197578743429882
31964	31990	52.91355821	45.71	7.20355820989442	15.7592610148642
33898	33899	37.63575852	37.915	0.279241484810619	0.736493432178871
35525	35551	39.18674741	38.46	0.726747413687697	1.88961886034243
36694	36695	8.34281661	8.342817	3.86526206597182E-07	4.63304189217122E-06
37283	37309	39.57391374	40.62	1.04608626490735	2.57529853497623
39543	39544	28.36952385	28.98	0.610476145318319	2.10654294450766
41653	41654	31.27637457	31.79	0.51362542510611	1.61568236900318
43770	43771	19.8699458	19.8	0.069945799087723	0.353261611554156
44859	44860	8.93436836	8.65	0.28436836419594	3.28749553983745
45817	45818	26.51803476	26.31	0.208034755485475	0.790706026170562
48269	48337	46.49246463	46.35	0.142464625747117	0.307367045840597
51489	51490	8.39503321	8.84	0.444966794162927	5.03356101994261
52715	52716	20.71302187	20.86	0.14697812776361	0.70459313405374
55116	55117	14.84071357	14.94	0.099286428987975	0.664567797777611
56075	56086	24.55695611	24.56	0.003043893387808	0.012393702719088
58861	58862	36.03948936	37.47	1.43051064281764	3.81774924691123
64186	64187	28.85588498	28	0.855884980066733	3.0567320716669
69723	69749	27.06560695	28	0.9343930504497	3.33711803732036
72642	72643	23.80248598	27.27	3.46751402431882	12.7154896381328
75681	75682	27.38444797	28.79	1.40555203108102	4.8820841649219
79819	79820	27.16472555	27	0.164725548989559	0.610094625887254
83096	83097	30.65766458	34.85	4.19233542465138	12.0296568856568
86449	86543	18.95143924	15.045849	3.90559023839059	25.9579252615827
88718	88744	44.74483446	44.7	0.044834460714711	0.100300806968033
97673	97674	41.87936796	41.97	0.090632040213478	0.215944818235592
102601	102602	21.01643509	24.07	3.05356491175409	12.6861857571836

Our module narrowed down 37 peaks from a dataset of 100000 data points as shown in Fig.5. The original dataset is represented using yellow color and cleaned one is represented using black.

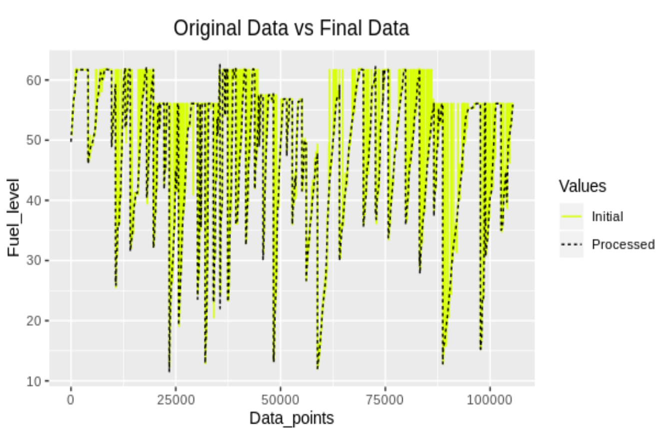

Fig. 5. *A comparison between our initial dataset and processed dataset. The data points are narrowed down, eliminating all the noises that were present at the beginning*

Additionally, we have tested our module with an additional four datasets. In this case, for each of the datasets, we have compared two consecutive stages for all the phases. The stages are Initial, Cluster, 1st Wavelet, 1st Median, 2nd Wavelet, 2nd Median and Final. We represented the data by superimposing the plots on top of one another. Each of the six graphs represents a comparison between the two phases of data denoising. For any graph the comparisons are between (a) raw data and data after clustering, (b) data after clustering and data after first wavelet analysis, (c) data after first wavelet analysis and data after first filtering, (d) data after first filtering vs data after second wavelet analysis, (e) data after second wavelet analysis and data after second filtering and (f) data after second filtering and final value. Also, unlike in our initial test where we have plotted a data points vs fuel graph, we have drawn a graph where X-axis corresponds to date and time and Y-axis corresponds to fuel value. Their graphical representations are presented from Fig.6 to Fig.13.

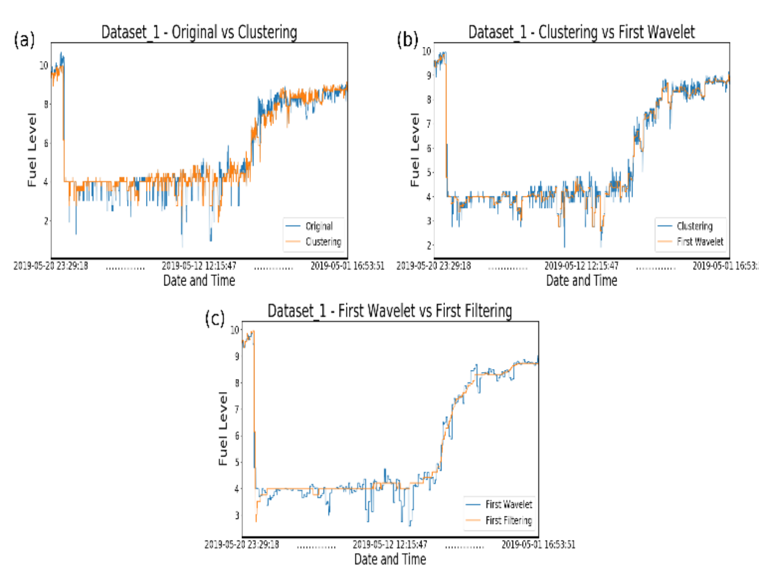

Fig. 6. *(a)-(c) One notable point here is most of the data is cleaned in the first three phases.*

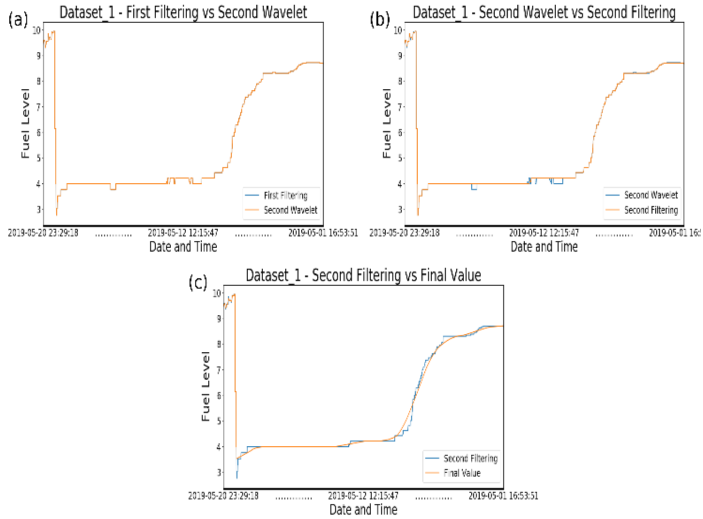

Fig. 7. *(a) There seems to be no effect of the second wavelet after the first filtering is performed. (b) Second filtering has caused small changes in the dataset. (c) Changes are significant as compared to immediately previous ones.*

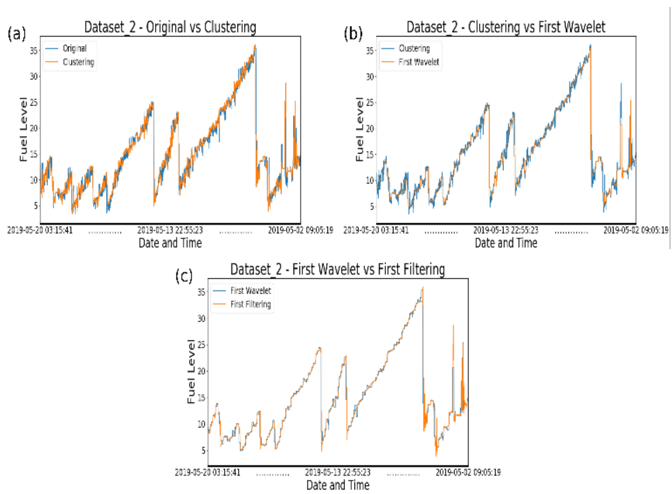

Fig. 8. *(a)-(c) Noise in prominent and heavy cleaning is performed.*

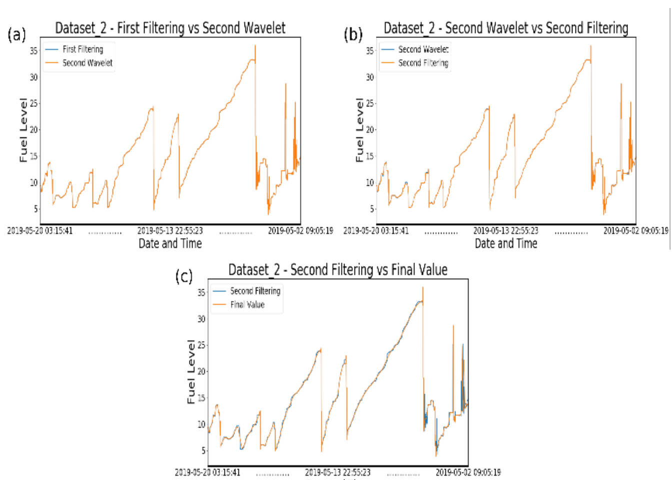

Fig. 9. *(a)-(b) Dissimilar to the previous graph, the state of the graph remains almost the same. Hence, there seems to be a small effect of second wavelet and second filtering after the first filtering is performed. (c) Final peak validation changes the graph by a small margin.*

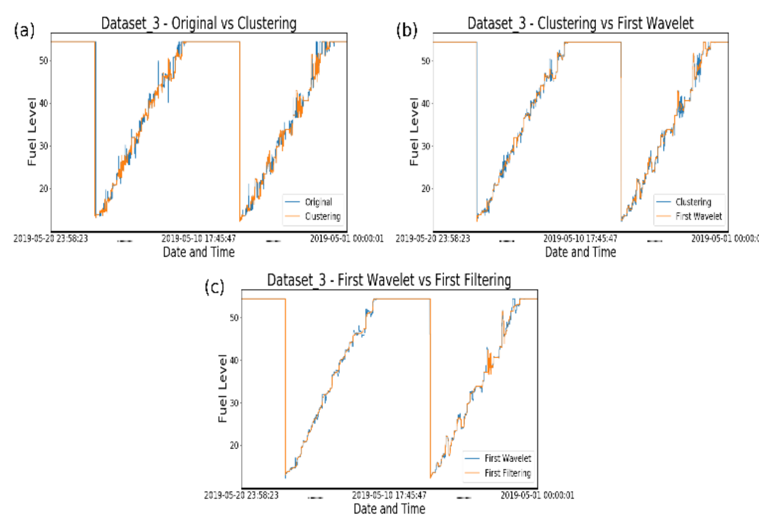

Fig. 10. *(a)-(c) Another notable point is that noise is more condensed along with the consumption rate which is shown in the graph by a downward trend.*

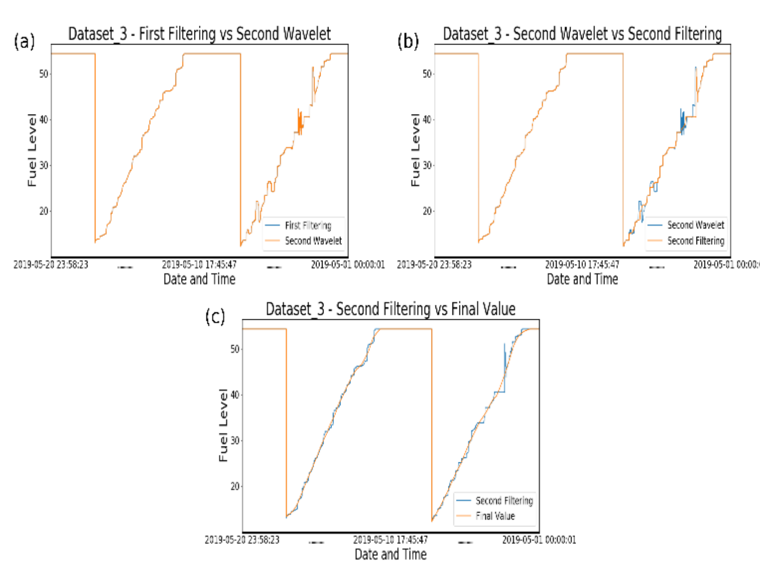

Fig. 11. *(a) Similar to Fig.8 and Fig.9, the difference is minimal after the implementation of wavelet for the second time. (b)-(c) Cleaning effect is visible in the downward trend.*

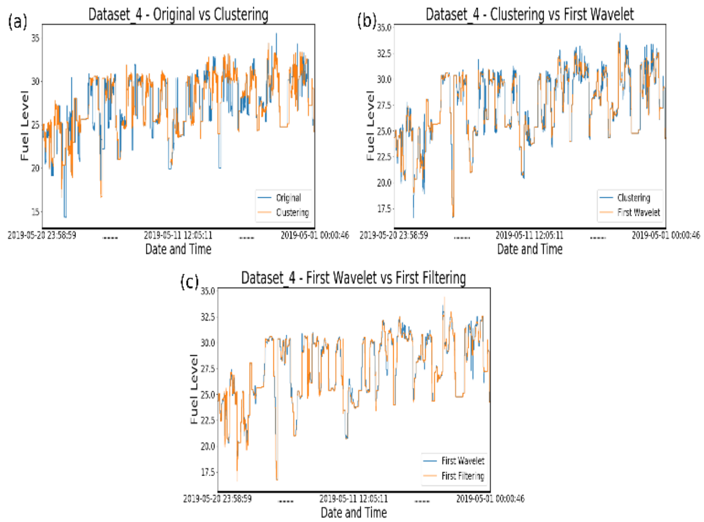

Fig. 12. *(a)-(c) Fluctuations in fuel level is much prominent as compared to the previous dataset.*

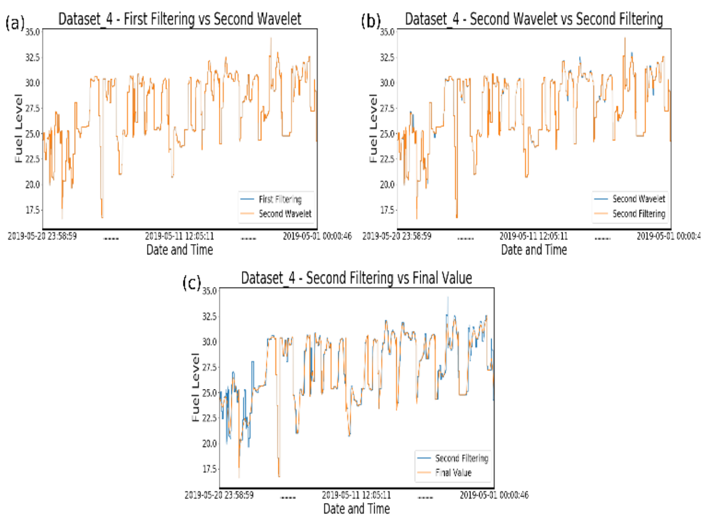

Fig. 13. *(a)-(b) Although, state of the graph remains almost same, (c) in the last stage due the peak validation method we see changes in the final dataset as compared to the previous adjacent graph.*

4. Conclusion

In conclusion, the real dataset (on which we have worked) has contained severe noise, which may have occurred due to poor street condition of some roads of Bangladesh, noisy dynamics, mobile environment and so on. The proposed module in this work, regardless of the intensity of data noise, is capable of detecting peaks without removing the noise in the first place. The reason behind the capability of doing that: different methods have been arranged in such a way that it can store the data before sending it to the next phase or process. For instance, data generated from clustering is stored first before sending it to the following phase to perform wavelet transform. This strategy allows the system to compare the values of the datasets retrieved from two adjacent processes. Before removing the noise, this entire task is conducted using multiple peak validity and peak detection methods. Consequently, the system works more efficiently as it is independent of the noise percentage. So regardless of the percentage of noise present in the datasets, peaks have been detected. However, after successful peak detection, the noise has been gotten removed and the correct value of the consumption rate has also been generated. So, this system can isolate noise and peak first; and then remove noises from the rest of the datasets, which provides not only correct peak values but also consumption values as well. Since this is a two layer-based noise removal system, at first peaks have been separated from the noisy dataset. Then, the consumption rate is deduced using these separated peaks. This approach helps to maintain the integrity of the peaks. This type of design allows us to minimize data loss during noise removal, which has not been reported previously in the presence of bizarre data patterns i.e. constant values, accuracy subsides. Without such extreme cases, accuracy may increase a lot. Our work can be implemented in any of the industrial fields, where exist the complicated issues of fuel measurement i.e. vehicle industry, fuel station, power plant and so on. As long as any time-series data distribution resembles our dataset distribution, the module can work efficiently to reduce noise and produce correct values.

Acknowledgement

M.R.C. Mahdy acknowledges Dr. K.M. Masum Habib and Mr. Mahmudul Hasan Sohag of Pi-Labs Bangladesh Limited for several important discussions. M.R.C. Mahdy also acknowledges Mr. Saikat Chandra Das of North South University (ECE Dept.) for several important discussions. M.R.C. Mahdy also acknowledges the support of NSU internal fund (CTRG Research grant 2019-20) approved by the BOT members of NSU.

5. References

[1] Kollem, S., Reddy, K. R. L., & Rao, D. S. (2019). A Review of Image Denoising and Segmentation Methods Based on Medical Images. *International Journal of Machine Learning and Computing*, *9*(3).

[2] Gondara, L. (2016, December). Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)* (pp. 241-246). IEEE.

[3] Singh, A., & Verma, R. S. (2017). An efficient non-local approach for noise reduction in natural images. *International Journal of Advanced Research in Computer Science*, *8*(5).

[4] Kaur, C., & Bansal, N. (2016). De-Noising Medical Images Using Low Rank Matrix Decomposition NN and SVM. *(IJCSIT) International Journal of Computer Science and Information Technologies*, *7*(1), 45-48.

[5] Saihood, A. A. (2014). Image Denoising using Neural Network with SVM (Support Vector Machine) and LDA (Linear Discriminant Analysis). *Int. J. Comput. Sci. Inf. Technol.*, *5*(4), 5363-5367.

[6] Wu, J., Huang, D. Y., Xie, L., & Li, H. (2017, August). Denoising Recurrent Neural Network for Deep Bidirectional LSTM Based Voice Conversion. In *INTERSPEECH* (pp. 3379-3383).

[7] Yang, Q., Yan, P., Zhang, Y., Yu, H., Shi, Y., Mou, X., & Wang, G. (2018). Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss. *IEEE transactions on medical imaging*, *37*(6), 1348-1357.

[8] Miranda, A. L., Garcia, L. P. F., Carvalho, A. C., & Lorena, A. C. (2009, June). Use of classification algorithms in noise detection and elimination. In *International Conference on Hybrid Artificial Intelligence Systems* (pp. 417-424). Springer, Berlin, Heidelberg.

[9] Germain, F. G., Chen, Q., & Koltun, V. (2018). Speech denoising with deep feature losses. *arXiv preprint arXiv:1806.10522*.

[10] Singla, E. M., & Singh, M. H. (2015). Paper on frequency based audio noise reduction using butterworth, Chebyshev & Elliptical filters. *Int. J. Recent Innov. Trends Comput. Commun*, *3*, 5989-5995.

[11] Haque, M., & Bhattacharyya, K. (2018). Speech Background Noise Removal Using Different Linear Filtering Techniques. In *Advanced Computational and Communication Paradigms* (pp. 297-307). Springer, Singapore.

[12] Welk M., Bergmeister A., Weickert J. (2005) Denoising of Audio Data by Nonlinear Diffusion. In: Kimmel R., Sochen N.A., Weickert J. (eds) Scale Space and PDE Methods in Computer Vision. Scale-Space 2005. Lecture Notes in Computer Science.

[13] Magsi, H., Sodhro, A. H., Chachar, F. A., & Abro, S. A. K. (2018, March). Analysis of signal noise reduction by using filters. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-6). IEEE.

[14] Magondu, S. (2016). *Using Neural Networks to reduce noise in internet of things data streams* (Doctoral dissertation, University of Nairobi).

[15] Agresti, G., Schaefer, H., Sartor, P., & Zanuttigh, P. (2019). Unsupervised domain adaptation for ToF data denoising with adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5584-5593).

[16] He, Q., Wang, X., & Zhou, Q. (2014). Vibration sensor data denoising using a time-frequency manifold for machinery fault diagnosis. *Sensors*, *14*(1), 382-402.

[17] Manmohan, Gupta, S., & Kuldeep. (2015). An Novel Approach for image denoising using Wavelet Transforms. International Journal of Engineering Research and Management (IJERM), 02(1), 142-146.

[18] Mythili, C., & Kavitha, V. (2011). Efficient technique for color image noise reduction. *The research bulletin of Jordan, ACM*, *1*(11), 41-44.

[19] Koskinen, K., Auvinen, P., Björkroth, K. J., & Hultman, J. (2015). Inconsistent denoising and clustering algorithms for amplicon sequence data. *Journal of Computational Biology*, *22*(8), 743-751.

[20] Chatterjee, P., & Milanfar, P. (2009). Clustering-based denoising with locally learned dictionaries. *IEEE transactions on Image Processing*, *18*(7), 1438-1451.

[21] Mahdy, M. R. C., Rivy, H. M., Jony, Z. R., Alam, N. B., Masud, N., Al Quaderi, G. D., & Rahman, M. S. (2020). Dielectric or plasmonic Mie object at air–liquid interface: The transferred and the traveling momenta of photon. *Chinese Physics B*, *29*(1), 014211.

[22] Chowdhury, R., Mahdy, M. R. C., Alam, T. N., Al Quaderi, G. D., & Rahman, M. A. (2020). Predicting the stock price of frontier markets using modified Black–Scholes Option pricing model and machine learning. *Physica A: Statistical Mechanics and its Applications*, 124444.

[23] Chowdhury, R., Rahman, M. A., Rahman, M. S., & Mahdy, M. R. C. (2020). An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning. *Physica A: Statistical Mechanics and its Applications*, 124569.

[24] Ahmed, Nahian, M. Nazmul Islam, Ahmad Saraf Tuba, M. R. C. Mahdy, and Mohammad Sujauddin. "Solving visual pollution with deep learning: A new nexus in environmental management." *Journal of environmental management* 248 (2019): 109253.

[25] Hambal, A. M., Pei, Z., & Ishabailu, F. L. (2017). Image noise reduction and filtering techniques. *International Journal of Science and Research (IJSR)*, *6*(3), 2033-2038.

[26] Ding, S., Zhang, L., & Zhang, Y. (2010, April). Research on spectral clustering algorithms and prospects. In *2010 2nd International Conference on Computer Engineering and Technology* (Vol. 6, pp. V6-149). IEEE.

[27] Murtagh, F. (2011). Hierarchical Clustering. Computing Research Repository 633-635.

[28] Sudha, S., Suresh, G. R., & Sukanesh, R. (2009). Speckle noise reduction in ultrasound images by wavelet thresholding based on weighted variance. *International journal of computer theory and engineering*, *1*(1), 7.

[29] Aggarwal, R., Singh, J. K., Gupta, V. K., Rathore, S., Tiwari, M., & Khare, A. (2011). Noise reduction of speech signal using wavelet transform with modified universal threshold. *International Journal of Computer Applications*, *20*(5), 14-19.

[30] Üstündağ, M., Şengür, A., Gökbulut, M., & Ata, F. (2013). Performance comparison of wavelet thresholding techniques on weak ECG signal denoising. *Przegląd Elektrotechniczny*, *89*(5), 63-66.

[31] Boateng, K. O., Asubam, B. W., & Laar, D. S. (2012). Improving the effectiveness of the median filter.