ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Machine Learning-Based Adaptive Equalization with Software-Defined Radio Experimental Setup

Annapurna H. S.[1] and S. Devi[2]

ABSTRACT

High-speed data transfer over the communication channel is now possible due to developments in wireless communication. However, as these data are transmitted over the channels, multiple elements' interference and interventions will disrupt the network's functionality, frequently causing data to be misinterpreted or distorted owing to overlap. Channel equalization is a notion that can be applied with the help of machine learning and artificial intelligence to counteract this kind of interference. The hybrid technique, which extracts features from an equalizer utilized in the channel in training and tracking modes, is the subject of current research efforts. Machine learning techniques are applied to distinguish between low, high, medium, and open space situations. Analysing the radio frequency signals that travel across the channel allows for distinction. The outcome and examination of multiple machine algorithms show that the suggested model functions well in a real-time setting. Multiple sample ratios and classifier models are used to train and test algorithms such as SVM, decision tree, random forest, KNN, logistic regression, and naive Bayes. Based on the parameters mapped by the confusion matrix, machine learning algorithms' performance and efficiency are estimated. When there are fewer samples overall, the random forest algorithm performs better than other algorithms. When there are more samples, the tree-based approach produces superior results. Decision trees can be implemented in real time since, in comparison, all environment types in high, low, and medium cluttered environments have produced better results.

Article information:

Keywords: Classification, Spectrum Sensing, Cognitive Radio, SVM, Machine Learning

Article history:
Received: June 20, 2024
Revised: October 3, 2024
Accepted: March 29, 2025
Published: April 19, 2025 (Online)

DOI: 10.37936/ecti-cit.2025192.256968

1. INTRODUCTION

The impact of time-related distortion within the channel significantly affects mobile communication [1]. This necessitates adaptive filters that can discern the diverse characteristics of the channel. A fundamental aspect of channel equalization involves the adjustment of tap weights. The evolution of virtual reality and the Internet of Things (loT) is poised to revolutionize cellular communication in the future. Accommodating the surge in data traffic demands a higher bandwidth. However, the intricate nature of these novel wireless systems presents challenges in their development [2]. Massive networks' low-latency requirements are becoming an issue. The lack of bandwidth usually causes the fading of channels, and the increasing number of devices will lead to this issue. In addition, phase and amplitude distortion can result in inter-symbol interference [3]. Due to the time-dependent nature of channel distortion, it can impact wireless communication. An adaptive filter should take into account the channel's attributes.

An adaptive equalizer can be implemented in a training sequence using an algorithm with specific criteria for adjusting coefficients [4]. This method ensures that the data transmission is reliable. Since the channel model of the transmission is unknown, an adaptive device is widely used in mobile applications. Non-linear structures can be utilized to create adaptive equalizers, which are widely used in various applications like image recognition and gaming.

[1]The author is with the Research scholar, PRIST Deemed to be University, Thanjavur, TN-612002, India , Email: anupankaj1@gmail.com
[2]The author is with the Research Supervisor, Associate Professor, Department of ECE, PRIST Deemed To be University, Thanjavur, TN-612002, India, Email: devi.bharath@gmail.com

ML can learn complex relationships among multiple factors, which is challenging to model with standard mathematical frameworks. Its utility in developing wireless systems is immense.

Machine learning has been widely utilized in the design and development of 5G cellular network infrastructure. The Bayes conjecture is an ideal solution to the classification invariance issue in the equalization procedure. The channel theory's non-obvious nature suggests that the states can't be found near the receiver's location, and a few algorithms can yield approximations. The K-means algorithm [5] can be hard to implement when dealing with the different time channels. However, deep learning techniques have shown an improved result for channel allocation solutions than the conventional methods for wireless communication [6]. Also, it has been reported that multi-level learning algorithms are more adept at handling adaptive channel allocation. Machine learning models are easy to learn, while deep learning requires going through multiple ayers to get the most out of the collected information. ML models are more sophisticated over time, so they can continue learning. ML can play an essential role in developing effective channel equalizers for wireless networks [7]. It is widely used to create of accurate and efficient channel models for various applications. This paper aims to find a solution for enhancing the capabilities of ML-equipped channel models in physical frameworks.

The paper presents an overview of the numerous steps involved in developing and implementing an ML framework for wireless communication. It also covers the algorithm's implementation and architecture with appropriate results.

1.1 Multipath Interference

When the receiver is well- equipped, all the symbols within the message are independent [8]. However, if one of the symbols gets in the way, its value may be corrupted, preventing the message from being interpreted properly. This type of interference is known as inter-symbol interference (ISI). The recovery of a message can affect a system's response if the source of energy isn't interfering with it [9]. This phenomenon can happen when a loss of transmission signals occurs. A spike in the channel will then be repeated for each path. The number of impulse terms will also increase.

A linear filter response can be utilized to create a channel, and its most critical aspect is the delay spread. This design can also be created by using an equalizer, which counteracts the effects of the signal. The value of the signal that is received and transmitted is respectively x and y [10].

$$
\begin{aligned}
y(t) = \; & r_1 x(t-T_1) + r_2 x(t-T_2) + \cdots + \\
& r_n x(t-T_N) + w(t)
\end{aligned}
\tag{1}
$$

An impulse-response filter considers the delay in the spread over a medium where the response is non-zero, and it is modelled digitally. The Eq. (1) standard can be generalized to fit an approximation assuming that a sampling interval is fixed [11].

$$
\begin{aligned}
(kT_s) = \; & r_1 x(kT_s) + r_2 x((k-1)T_s) + \cdots + \\
& r_n x((k-n)T_s) + w(kT_s)
\end{aligned}
\tag{2}
$$

1.2 Adaptive Equalization

his method can be utilized to alter the channel's variables regurly and notify the receiver [12]. It can also be executed through a training sequence or a short preamble. During the adjustment phase, the symbols of the equalizer are considered. This technique can be utilized in multiple decision-making exercises.

(A) Adaptive Equalization

When a receiver is trained, it triggers an equalizer based on the sequence of events that occurred during the training. The receiver will then receive the user's information once the program has ended.

(B) Tracking mode

After taking into account the adjustable coefficients, the output of Eq. (3) is calculated

$$
y(n) = \sum L - 1 \; h(n)x(k+T-n) \tag{3}
$$

The signal is sent through a filter, and then it is processed by the system. The training algorithm then passes the information contained in the $T(n)$ data set to the next generation [13].

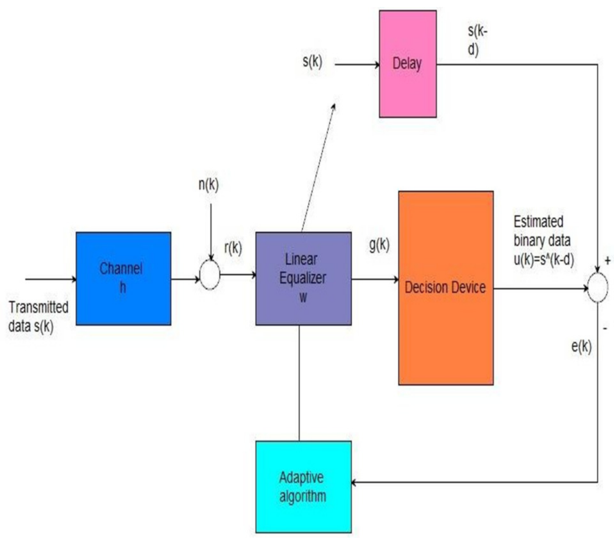

Fig.1: *Adaptive Channel Equalization.*

The coefficients of a filter are optimized by generating an error $e(n)$ in the $T(n)$ value.

The quantity can be reduced by seazlecting the coefficients $h(n)''$ if the criterion for the least squares is followed again [14].

$$\boldsymbol{\Psi_L = \sum\nolimits_{n=0}^{N}[T(n) - y(n)]^2 = \sum\nolimits_{n=0}^{N}[T(n)\sum\nolimits_{n=0}^{N} h(n)x(k+T-n)]^2} \quad (4)$$

The results of the optimization are shown in the following linear equations.

$$\sum_{n=0}^{L-1} h(n)r_{xx}(l-n) = r_{dx}(l-t), l = 0, 1, 2, \ldots, (L-1) \quad (5)$$

While Eq. (5) is usually utilized recursively, the coefficients can be utilized when the adjustment is required. The sequence x is last transmitted once the training period ends. Coefficients are adjusted to ensure that they can track the changes in channel time. This is a very effective technique if errors happen frequently.

2. EXPERIMENTAL SETUP- SOFTWARE-DEFINED RADIO

Current literature catalogs the classification techniques that can are used to classify environments according to their surrounding structure [15]. These techniques are used to define areas as semi-outdoor, indoor, and outdoor. Generally, a region's classification is determined by looking at the data that was collected by the cell towers [16]. This method can be very laborious and requires many informations to analyse. Other techniques, such as identity maps and sound waves, can also be utilized. A classification system is then developed that takes into account the different parts of the environment.

This paper intends to create a machine learning based system [17], that is supported by multiple software and hardware, such as GNU Radio and USRP N210 [18]. The system's classification process can be carried out in real time. The system takes into account multiple parameters when evaluating and inputs according to the cluttered environment's signal characteristics, taking into account the noise or signal from the user. The outputs are then split into a training and testing group using ML algorithms such as KNN, Logistic Regression, Random Forest, and Naive Bayes. The training set's requirements are then considered to evaluate the models' precision, accuracy, sensitivity, F1_score, and specificity. The decision devices then pick the best algorithm-based on their environment. The equalizer considers the Information it has received and adjusts the weights accordingly. It then passes the signal to its receiver section for decoding and demodulation.

The signals such as their frequency, amplitude, and phase difference. A sample dataset containing three noise outputs and seven user signals is presented in Fig.1[19] The signal frequency and amplitude are adjusted according to the SNR value. The system is then evaluated by different ML platforms [20-22].

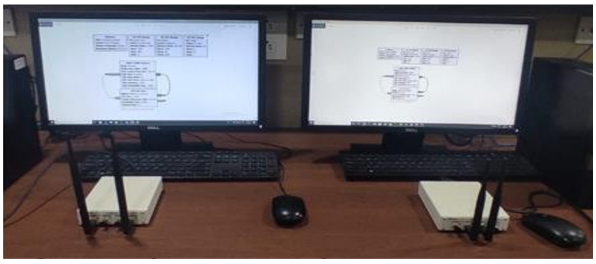

Fig.2: *Experimental setup.*

3. PROPOSED SYSTEM MODEL

The use of linear equalizations is shown in Fig. 3. They can be used in the areas of linear prediction and regression analysis. The $N+1$ parameter can are used to define the model. Outputs are sent to the decision machine depending on the distance from the circle and its nearest symbol. The algorithm is based on a structure that is shown in Fig 3, and it seeks to replace the filter procedure for calculating the FIR with a method that can predict its outputs. Although its input parameters are identical to those used in N samples, the program can be programmed to utilize classification or regression techniques.

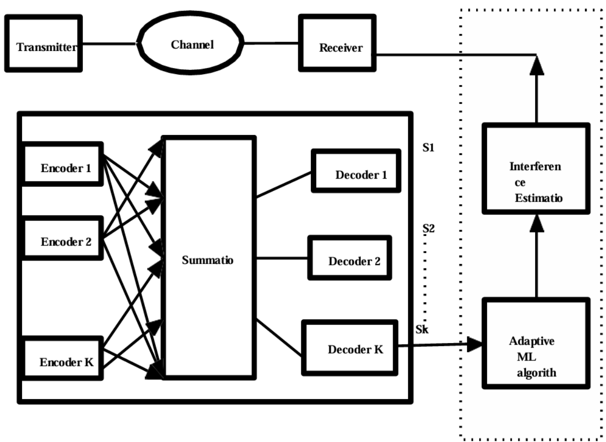

Fig.3: *Conventional equalizer and ML.*

The program shown in Fig. 4 has the suggested system's workflow, which includes the signals and characteristics. It imports the resulting files. The generated attributes are then categorized into outputs.

Table 1: *Sample dataset for 7 number of user signals and 3 noise signals in terms of frequency,amplitude ,phase from different environments.*

UserID	Highly cluttered (0-20)	Medium cluttered (21-45)	Low cluttered (46-97)	Open -space (98-154)	Operating frequency	Amplitude	Phase	Presence of signal or Noise
1	21	46	98	154	2400001000	92	36	1
2	16	42	95	152	2400166666	85	321	1
3	15	39	87	148	2400333333	82	68	1
4	13	36	85	139	2400500000	83	45	1
5	11	34	83	133	2400666667	74	52	0
6	9	30	81	129	2400833333	71	24	0
7	7	27	71	118	2401000000	68	18	1
8	5	23	65	114	2401166667	62	27	0
9	3	22	55	107	2401333332	56	60	1
10	2	21	46	98	2401500000	54	13	1

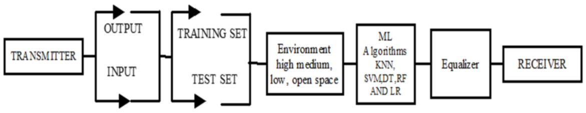

Fig.4: *The proposed workflow.*

4. MACHINE LEARNING ALGORITHMS

4.1 Logistic Regression

Linear regression analysis evaluates the association between a single dependent variable or a group of variables. It can also explain the dataset's traits. Unfortunately, this approach tends to predict values outside of a particular region. One of the most common problems that can be encountered with linear regression is the fit of the line. This can be solved by converting the ideal fit into a curve through the sigmoid function. The logit transformation considers the path of the x function. We'll get the value between 0 and 1, using this procedure.

$$\log p(x)/1 - p(x) = \beta 0 + \beta.x \tag{6}$$

After solving for $p(x)$

$$p(x) = e^{\beta 0+\beta}/e^{\beta 0+\beta+1} \tag{7}$$

By adjusting the threshold of a logistic regression Model, a linear classifier can be established to categorize a given dataset automatically. For instance, if the prediction score surpasses 0.5, it can be interpreted as belonging to the positive class, thereby altering the classification outcome.

4.2 KNN CLASSIFIER

The KNN framework considers the different types of data that are gathered and stored by each of its neighbors. It performs a learning process based on its similarity. Another crucial part of the system is its ability to include its neighbors in the voting procedure.

A KNN framework can perform a search for nearby data points if its value is zero. It then ranks its results and assigns the best possible locations.

The input (x) is associated with the class that has the highest probability among the given classes.

$$d(a,\grave{a}) = \text{square root}\{(a_1 - \grave{a})^2 + \ldots + (a_n - a_n\grave{})^2\}(8)$$

The input (x) is associated with the class that has the highest probability among the given classes.

$$P(B = j|x = a) = (1/k)\sum I(B^i = j) \tag{9}$$

Regression will be the same as before, except that it considers the target's value instead of the neighbor's classes. This method can be used to find the datapoint's unseen value.

4.3 SVM classifier

The SVM is a standard method for solving nonlinear problems that involve the use of the kernel method. It aims to find a hyperplane with enough space to accommodate the Model's two components. The variables can be dealt with in multiple ways. For instance, the Rn information Model employs the values x_k and y_k. The training data for the hyperplane Hp can be obtained by referring to the given equation.

$$T = \{(x_k, y_k) \in R^n, \quad k = 1, \ldots n\} \tag{10}$$

$$H_p : w_g.x_i + \tau \tag{11}$$

The y_k value can be computed by the boundary between the classes of information represented by the vectors and the scales, which respectively represent SVM (Support Vector Machine) can be categorized into two types: linear SVM and nonlinear SVM.

4.4 Naïve Bayes Algorithm

The Bayes Theorem is considered the basis for the Naive Bayes algorithm, a classification method. It is mainly utilized in the analysis of natural language. The Naive Bayes method considers account the fact that the features that are used in determining a target do not coordinate with each other. While the Bayes Theorem may not always yield absolute correctness, it efficient and valuable in multiple applications. This theorem asserts that the probability of an event occurring can be inferred based on our understanding of factors that might influence it.

4.5 Decision Tree classifier

Decision trees are the representations that show how a single variable affects various rules and decisions. It iterates through its level to produce a label indicating the leaf node, and the information within it is considered pure.

4.5.1 Entropy

Entropy is a type of information that describes a set of data. On the other hand, a homogeneous set has

Entropy is a type of information that describes a set of data. On the other hand, a homogeneous set has zero elements, while one which is in a similar region has all similar elements.

$$Entropy = -\sum^{n} p_x * \log(p_x) \tag{12}$$

4.5.2 Gini index/Gini impurity

The Gini Index is a tool that are used to measure the impurities in a node. While some homogeneous samples have similar elements, their Gini index values are unequal.

$$\text{Gini Index} = 1 - \sum Px^2 \quad \text{with } i = 1 \text{ to } n \tag{13}$$

The impurities' function serves to gauge the uniformity of data. When the data is homogeneous, meaning it predominantly belongs to the same class or tree, the impurity measure will below.

4.6 Random Forest

The goal of the impurities is to determine the uniformity of the data. If there's a homogeneous sample, then the tree or class associated with it will be the same. The Random Forest classifiers are designed to improve a Model's effectiveness and efficiency. The name of the framework refers to how it takes into consideration the dataset's decision trees.

5. RESULTS AND DISCUSSION

The performance of different advanced Models, such as SVC and KNN, in three different environments was evaluated. The first was a low-density area with few people, the second was a large population with many machines, and the third was a cluttered environment with many machines. The results of this research are presented in three tables. The algorithms' precision, sensitivity, specificity, and F1_score is then analyzed.

A confusion matrix (see Fig. 5), typically represented as a 2×2 matrix, juxtaposes the predicted values of an algorithm against the exact values. The rows represent the exact classes, while the columns are the predicted ones. This matrix facilitates the evaluation of classification quality. In an ideal scenario, the matrix would display both false negatives and false positives, indicating that the algorithm makes mistakes in both directions. This evaluation is conducted within a training setting where 75% of the dataset is used for training, while the remaining 25% constitutes the test set.

If the differences between false positives and false negatives are not the same, then F1 is probably the best choice. This can be used in situations where there is an uneven distribution of classes. The accuracy of F1 is computed based on the degree to which its false positives are true. The recall is computed on the confidence that its false positives are still intact. To recall false alarms instead of genuine ones, choose sensitivity. Doing so makes sense since covering all true ones would be ideal, discarding.

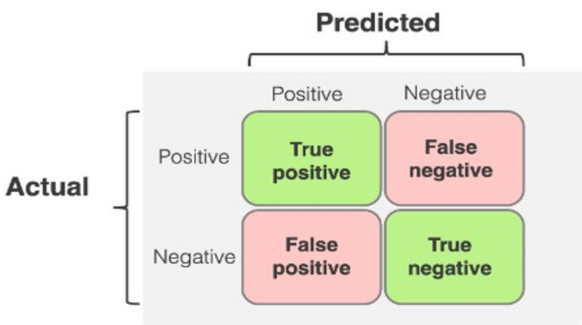

Fig.5: *Confusion Matrix.*

Table 2 shows the results of the various algorithms that are used in the study. They all performed well in all four. The Random Forest classifier is ideal for detecting anomalies since it has a 99% accuracy. The table below shows the various algorithms' performance regarding specificity, precision, and accuracy. The Random Forest is the best performer, with an accuracy of 99.5%, a precision of 99.2%, and a specificity of 99.1%. On the other hand, Logistic Regression performed better with a sensitivity of 100% and an F1_score of 80%. On the other hand, Naive Bayes and KNN cannot perform well in cluttered environments. In terms of F1_score, specificity, accuracy, and KNN's precision, it is the most accurate classifier, while the random forest is the best at dealing with such environments. The third table shows the various types of decision trees that can be used in different areas. For instance, the random forest is more suited for areas with low ceilings. Table 4 illustrates the various parameters of decision trees and regression procedures when dealing with cluttered settings. For instance, the former has a 92.3% accuracy and a sensitivity of 91.6%, while the latter has a 91.6% accuracy and a sensitivity of 92.3%. Classifiers for decision trees perform better than those used for KNN and SVM in cluttered environments when it comes to accuracy, specificity, sensitivity, F1_score, and precision. A random forest classifier can be more accurate when it comes to precision, specificity, and sensitivity.

Logistic regression is not an ideal choice when it comes to identifying and managing cluttered environments. The decision tree can perform better than this. Both Naive Bayes and SVM are known to be very inefficient when it comes to managing and identifying cluttered environments. Table 3 shows that the former suits for smaller spaces, while the latter is ideal for bigger ones. According to the findings, a single ML algorithm may not be used in every situation. Instead, a device should use a combination of methods that are based on the environment and samples. For instance, if a facility has a sample count of only ten, the random forest method is favored. On the other hand, if it has a lot of clutter, the decision tree is a better choice when it comes to managing and identifying areas. It has been observed that the effectiveness of these methods rises as the number of samples increases.

6. GRAPHICAL REPRESENTATIONS

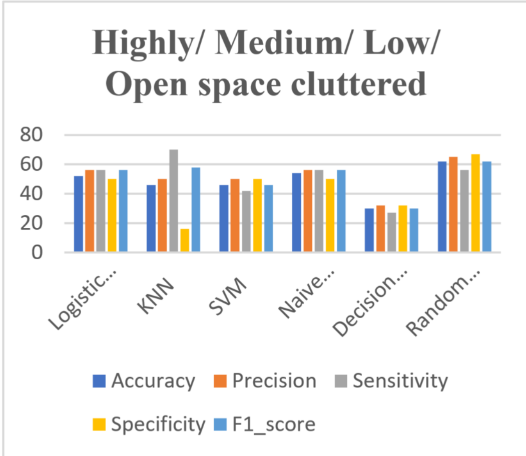

Fig.6: *Comparative analysis of different parameters and ML algorithms concerning different environments for 10 samples.*

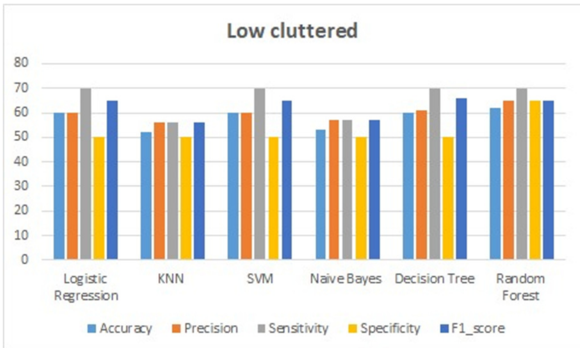

Fig.7: *Comparative analysis of different parameters and ML algorithm concerning low environment.*

Table 2: *For a performance analysis of various ML algorithms, 10 samples are examined show they performed in different environments.*

Environment	Type of algorithm	Accuracy	Precision	Sensitivity	Specificity	F1_score	Confusion Matrix
Low/ Medium/ highly/ Open space cluttered	Logistic Regression	67	67	100	0	80	$\begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix}$
	KNN Classifier	67	67	100	0	80	$\begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix}$
	SVM Classifier	67	67	100	0	80	$\begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix}$
	Naive Bayes	67	67	100	0	80	$\begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix}$
	Decision Tree	68	99	52	99	68	$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$
	Random Forest	99.5	99.2	99.3	99.1	99.9	$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

Table 3: *For a performance analysis of various ML algorithms, 50 samples will be examined, in different environments.*

Environment	Type of algorithm	Accuracy	Precision	Sensitivity	Specificity	F1_score	Confusion Matrix
highly cluttered	KNN	61	62.5	71.4	50	66.6	$\begin{bmatrix}2 & 4\\2 & 5\end{bmatrix}$
	Logistic Regression	62	64	73	50	66.3	$\begin{bmatrix}3 & 3\\2 & 5\end{bmatrix}$
	SVM	45.1	51	43.8	51	45.1	$\begin{bmatrix}3 & 3\\4 & 3\end{bmatrix}$
	Naive Bayes	45.1	51	43.8	51	45.1	$\begin{bmatrix}3 & 3\\4 & 3\end{bmatrix}$
	Decision Tree	62.5	67	68.1	67	63	$\begin{bmatrix}2 & 4\\3 & 4\end{bmatrix}$
	Random Forest	45.1	50	58.1	33.3	54.3	$\begin{bmatrix}2 & 4\\3 & 4\end{bmatrix}$
Medium cluttered	Logistic Regression	53.8	57.1	57.1	50	57.1	$\begin{bmatrix}3 & 3\\3 & 4\end{bmatrix}$
	KNN	38.4	42.8	42.8	33.3	42.8	$\begin{bmatrix}2 & 4\\4 & 3\end{bmatrix}$
	SVM	53.8	60	42.8	66.6	50	$\begin{bmatrix}4 & 2\\4 & 3\end{bmatrix}$
	Naive Bayes	45.1	51	43.8	51	45.1	$\begin{bmatrix}3 & 3\\4 & 3\end{bmatrix}$
	Decision Tree	60.5	63.5	73	53	66.6	$\begin{bmatrix}3 & 3\\2 & 5\end{bmatrix}$
	Random Forest	47	52	45.8	50	48.1	$\begin{bmatrix}3 & 3\\4 & 3\end{bmatrix}$
Low cluttered	Logistic Regression	60.5	61.5	71	51	65.6	$\begin{bmatrix}3 & 3\\2 & 5\end{bmatrix}$
	KNN	56	59	59.1	54	57.1	$\begin{bmatrix}3 & 3\\3 & 4\end{bmatrix}$
	SVM	64.5	65.5	73.4	54	64.6	$\begin{bmatrix}3 & 3\\2 & 5\end{bmatrix}$
	Naive Bayes	54	56	57	48	56	$\begin{bmatrix}3 & 3\\3 & 4\end{bmatrix}$
	Decision Tree	62	63	72	48	68	$\begin{bmatrix}3 & 4\\2 & 4\end{bmatrix}$
	Random Forest	64	67	72.5	65	64	$\begin{bmatrix}4 & 2\\3 & 4\end{bmatrix}$
Open space	Logistic Regression	53.8	57.1	57.1	50	57.1	$\begin{bmatrix}3 & 3\\3 & 4\end{bmatrix}$
	KNN	46.1	50	71.4	16.6	58.8	$\begin{bmatrix}1 & 5\\2 & 5\end{bmatrix}$
	SVM	45	48	44	49	45	$\begin{bmatrix}3 & 3\\4 & 3\end{bmatrix}$
	Naive Bayes	53	56	58	49	58	$\begin{bmatrix}3 & 3\\3 & 4\end{bmatrix}$
	Decision Tree	31	34	29	34	31	$\begin{bmatrix}2 & 4\\5 & 2\end{bmatrix}$
	Random Forest	62	67	58	65	62	$\begin{bmatrix}4 & 2\\3 & 4\end{bmatrix}$

7. CONCLUSION

The implementation of traditional methods used in the communication channel for the concept of adaptive filter and channel equalization is challenging. The various machine learning-based algorithms can be effectively implemented in the decision device for all kinds of cluttered environments. The receiver setup can be simplified by using machine learning algorithms for the channel equalization concept. The paper suggest the analysis of various machine learning algorithms, tabulated for different kinds of cluttered environments. The experimental setup suggest the assessment in an indoor environment. The experimental results suggest that the number of samples increases the accuracy of the algorithms and other parameters. The Model has to be subjected to machine learning algorithms, considering all the environmental factors in real- time, and the conclusions are included based on which ML algorithm is more appropriate for real real-time environment. The system considers the sample values in real- time and selects the appropriate type of algorithm to activate; hence,

Table 4: *For a performance analysis of various ML algorithms, 100 samples examined in different environments.*

Environment	Type of algorithm	Accuracy	Precision	Sensitivity	Specificity	F1_score	Confusion Matrix
highly cluttered	KNN	73	65.7	93.6	65.5	76.5	$\begin{bmatrix} 8 & 5 \\ 1 & 11 \end{bmatrix}$
	Logistic Regression	64	62	77	54.8	68.6	$\begin{bmatrix} 7 & 6 \\ 3 & 9 \end{bmatrix}$
	SVM	74	68	83.3	62.5	74	$\begin{bmatrix} 8 & 5 \\ 2 & 10 \end{bmatrix}$
	Naive Bayes	78	73.4	83.3	67.2	76.9	$\begin{bmatrix} 9 & 4 \\ 2 & 10 \end{bmatrix}$
	Decision Tree	97	99	84	99.5	95	$\begin{bmatrix} 13 & 0 \\ 2 & 10 \end{bmatrix}$
	Random Forest	86	99	75	99.5	84	$\begin{bmatrix} 13 & 0 \\ 3 & 9 \end{bmatrix}$
Medium cluttered	Logistic Regression	77	67.7	91.6	63.5	78.5	$\begin{bmatrix} 8 & 5 \\ 1 & 11 \end{bmatrix}$
	KNN	74	68.7	91.6	56.8	75.8	$\begin{bmatrix} 7 & 6 \\ 1 & 11 \end{bmatrix}$
	SVM	75	64.6	83.3	63.5	74	$\begin{bmatrix} 8 & 5 \\ 2 & 10 \end{bmatrix}$
	Naive Bayes	98	99	100	97	100	$\begin{bmatrix} 13 & 0 \\ 0 & 12 \end{bmatrix}$
	Decision Tree	99.5	99.4	99	99.7	99.5	$\begin{bmatrix} 13 & 0 \\ 0 & 12 \end{bmatrix}$
	Random Forest	94	86	98	86.6	95.3	$\begin{bmatrix} 11 & 2 \\ 0 & 12 \end{bmatrix}$
Low cluttered	Logistic Regression	63.5	55.2	73.5	46.1	65	$\begin{bmatrix} 6 & 7 \\ 3 & 9 \end{bmatrix}$
	KNN	85.5	79.5	94	79.9	87	$\begin{bmatrix} 10 & 3 \\ 1 & 11 \end{bmatrix}$
	SVM	68.5	65	84	54	72	$\begin{bmatrix} 7 & 6 \\ 2 & 10 \end{bmatrix}$
	Naive Bayes	68.5	66.4	75.5	62	68.9	$\begin{bmatrix} 8 & 5 \\ 3 & 9 \end{bmatrix}$
	Decision Tree	94.5	99.8	86	99.5	93	$\begin{bmatrix} 13 & 0 \\ 2 & 10 \end{bmatrix}$
	Random Forest	99.7	99.5	99	99.7	99.8	$\begin{bmatrix} 13 & 0 \\ 0 & 12 \end{bmatrix}$
Open space	Logistic Regression	78	67.7	93.6	63.5	77.5	$\begin{bmatrix} 8 & 5 \\ 1 & 11 \end{bmatrix}$
	KNN	92	85	100	84.6	94	$\begin{bmatrix} 11 & 2 \\ 0 & 12 \end{bmatrix}$
	SVM	73	70	75.5	69	72.9	$\begin{bmatrix} 9 & 4 \\ 3 & 9 \end{bmatrix}$
	Naive Bayes	71.5	70	76	69.4	72	$\begin{bmatrix} 9 & 4 \\ 3 & 9 \end{bmatrix}$
	Decision Tree	99.5	99.5	99	99.8	99.9	$\begin{bmatrix} 13 & 0 \\ 0 & 12 \end{bmatrix}$
	Random Forest	95.6	89	99.5	87	98	$\begin{bmatrix} 11 & 2 \\ 0 & 12 \end{bmatrix}$

without any human intervention, the system processes the signal effectively. Various Machine learning classifiers are embedded, which improves the system performance and optimization process to yield better efficient transmission. The novelty of the developed Models is any type of environment the Model can handle effectively by selecting a appropriate algorithm. Depending on the type of environment the system the total number of samples the algorithm is selected. Random forest is for analysis with a few samples or a minimum sample. The decision tree is best suited for all types of environments. the decision tree can be effectively implemented for the real-time environment with high and medium clutter environments. Further, various machine learning and deep learning algorithms can be implemented and trailed based on various experimental results. The authors aim to extend the concepts of deep learning network Models.

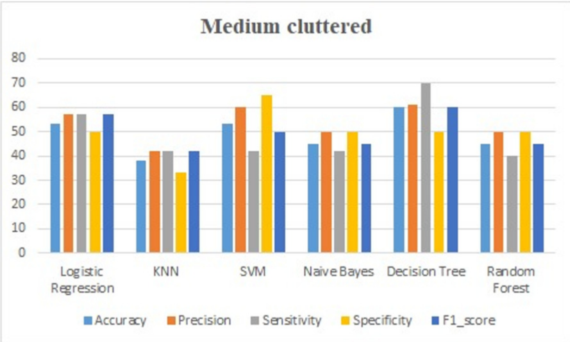

Fig.8: *Comparative analysis of different parameters and ML algorithms concerning the medium environment.*

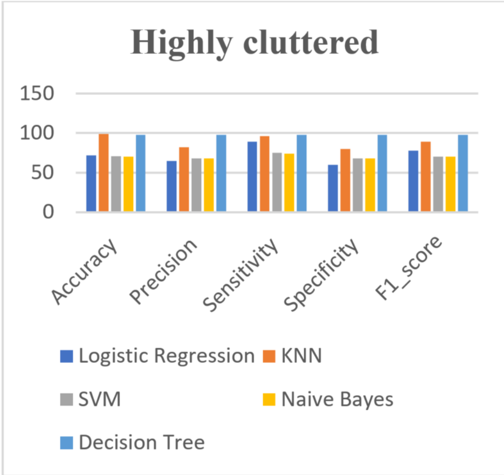

Fig.11: *Comparative analysis of different parameters and ML algorithm concerning a highly cluttered environment.*

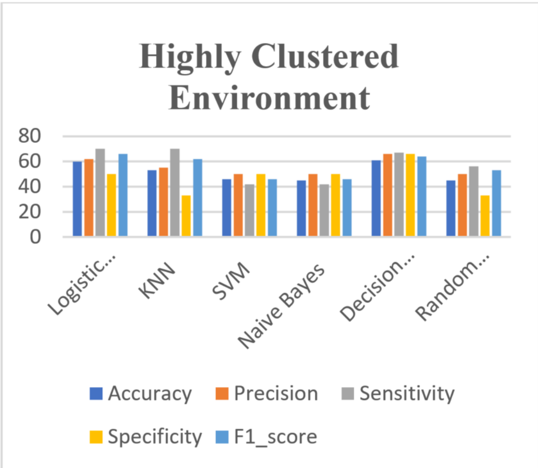

Fig.9: *Comparative analysis of different parameters and ML algorithms concerning a highly cluttered environment.*

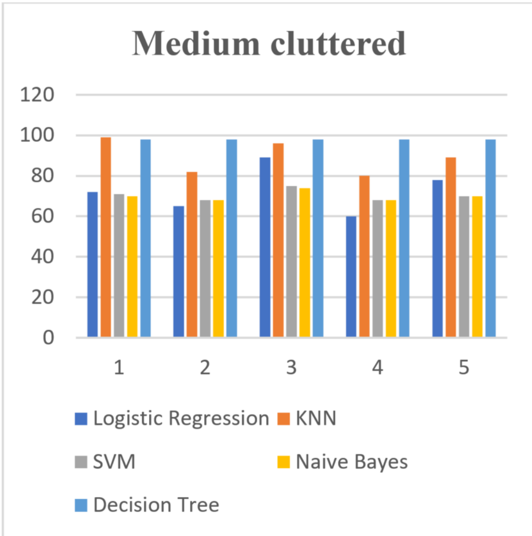

Fig.12: *Comparative analysis of different parameters and ML algorithms concerning a medium cluttered environment.*

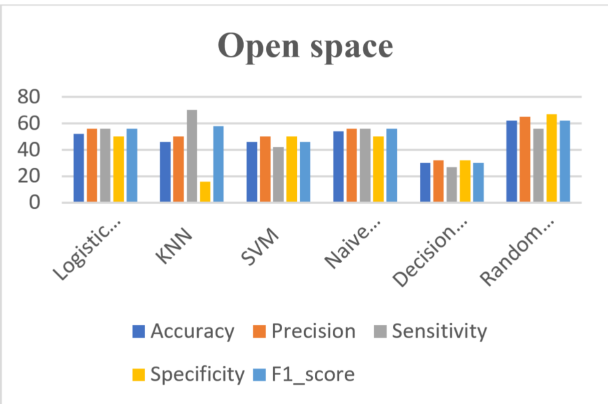

Fig.10: *Comparative analysis of different parameters and ML algorithm concerning an open cluttered environment.*

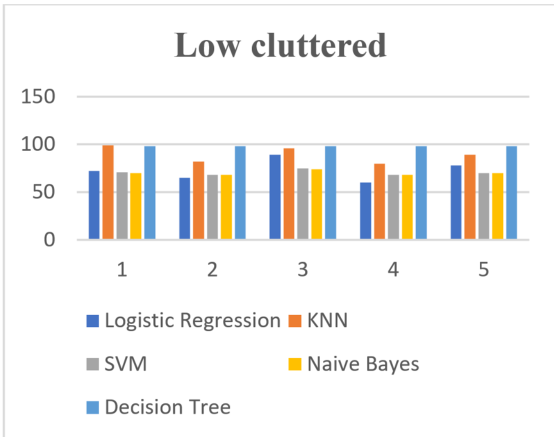

Fig.13: *Comparative analysis of different parameters and ML algorithms concerning low low-cluttered environment.*

AUTHOR CONTRIBUTIONS

Annapurna H. S Contributed substantially to the conception and simulation, or analysis and interpretation. S. Devi provided critical article revision and supervised the work.

References

[1] L. Dai, R. Jiao, F. Adachi, H. V. Poor and L. Hanzo, "Deep Learning for Wireless Communications: An Emerging Interdisciplinary Paradigm," in *IEEE Wireless Communications*, vol. 27, no. 4, pp. 133-139, August 2020.

[2] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta and P. Popovski, "Five disruptive technology directions for 5G," in *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74-80, February 2014.

[3] B. A. Mohammed and S. Y. Ameen, "Impact of Different Training Modes on Adaptive Equalization Techniques for MIMO-OFDM System," *Communications on Applied Electronics (CAE)*, vol. 7, no. 2, pp. 29-33, May 2017.

[4] G. Malik and A. S. Sappal, "Adaptive equalization algorithms: an overview," *International journal of advanced computer science and applications*, vol. 2 , no. 3, pp. 62-67, March 2011 .

[5] F. Ali, S. Ahmad, F. Muhammad, Z. H. Abbas, U. Habib and S. Kim, "Adaptive equalization for dispersion mitigation in multi-channel optical communication networks," *Electronics*, vol. 8, no. 11, p. 1364, 2019.

[6] U. K. Acharya and S. Kumar, "Genetic algorithm-based adaptive histogram equalization (GAAHE) technique for medical image enhancement," *Optik*, vol. 230, p.166273, March 2021.

[7] J. G. Proakis, *Digital communications in communications*, McGraw-Hill, 2nd ed., New York, 2003.

[8] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

[9] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, January 2016.

[10] M. Li, P. Zhou, Y. Zheng and Z. Li, "IODetector: A Generic Service for Indoor Outdoor Detection," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, ser. SenSys '12*, New York, NY, USA: ACM, pp. 113–126, 2012.

[11] Z. Liu, H. Park, Z. Chen and H. Cho, "An Energy-Efficient and Robust Indoor-Outdoor Detection Method Based on Cell Identity Map," *Procedia Computer Science*, vol. 56, pp. 189–195, January 2015.

[12] R. Sung, S.-h. Jung and D. Han, "Sound- based indoor and outdoor environment detection for seamless positioning handover," *ICT Express*, vol. 1, no. 3, pp. 106–109, December 2015.

[13] V. Radu, P. Katsikouli, R. Sarkar and M. K. Marina, "A Semi-supervised Learning Approach for Robust Indoor-outdoor Detection with Smartphones," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, ser. SenSys '14*, New York, NY, USA: ACM, pp. 280–294, 2014.

[14] B. S. K. Reddy, K. Mannem and K. Jamal, "Software-defined radio-based non-orthogonal multiple access (NOMA) systems," *Wireless Personal Communications*, vol. 119, pp. 1251-1273, February 2021.

[15] B. S. K. Reddy, "Experimental validation of non-orthogonal multiple access (NOMA) technique using software defined radio," *Wireless Personal Communications*, vol. 116, pp. 3599-3612, November 2020.

[16] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi and S. Homayouni, "Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 6308-6325, 2020.

[17] S. Tavara, "Parallel computing of support vector machines: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6 pp. 1-38, January 2019.

[18] S. Ghosh, A. Dasgupta and A. Swetapadma, "A Study on Support Vector Machine based Linear and Non-Linear Pattern Classification," *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, Palladam, India, pp. 24-28, 2019.

[19] A. Rizwan, N. Iqbal, R. Ahmad and D.-H. Kim, "WR-SVM model based on the margin radius approach for solving the minimum enclosing ball problem in support vector machine classification," *Applied Sciences*, vol. 11, no. 10, p.4657, May 2021.

[20] Y. Shi, X. Lu, Y. Niu and Y. Li, "Efficient Jamming Identification in Wireless Communication: Using Small Sample Data Driven Naive Bayes Classifier," in *IEEE Wireless Communications Letters*, vol. 10, no. 7, pp. 1375-1379, July 2021.

[21] H. B. Ahmad and Z. Su, "Ensemble classifier-based spectrum sensing in cognitive radio networks," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[22] Z. Noshad, *et al.*, "Fault detection in wireless sensor networks through the random forest classifier," *Sensors*, vol. 19, no. 7, p. 1568, 2019.

Annapurna H. S. received the Engineering degree in Electronics and Communication Engineering from Bellary Institute Of Technology & Management(BITM) in 2005.She received the Master degree in Digital Communication & networking(DCN) from M.S. Ramaiah Institute Of Technology (MSRIT),Bengaluru in 2010.Currently she's a research scholar at PRIST University, Thanjavur. Currently she's working as an Assistant Professor at Presidency University ,Bengaluru having 18 years of teaching experience with her research at PRIST Deemed to be University,Thanjavur. Her research interests include wireless communication and sensors with ML applications .She has published many research papers. She can be contacted at email:anupankaj1@gmail.com.

S. Devi completed B. E. in Electronics and Communication Engineering in Bharathidasan University, Trichy and M.Tech in Communication Systems in Dr. M.G.R University, and Ph.D in Anna University. She has more than ninteen years of experience. She has published more than twenty papers in reputed peer reviewed Journals, Conferences, and book. She can be contacted at email:devi.bharath@gmail.com.