International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8473
***Article Received:** 24 July 2023 **Revised:** 15 September 2023 **Accepted:** 02 October 2023*

Experimental Testing and Validation of Adaptive Equalizer Using Machine Learning Algorithm

Annapurna H. S[1], A. Rijuvana Begum[2]
[1]Research scholar, PRIST Deemed To be University, Thanjavur, TN-612002, India
e-mail: anupankaj1@gmail.com
[2]Research Supervisor, PRIST Deemed To be University, Thanjavur, TN-612002, India
e-mail: arbbegum@gmail.com

Abstract— Due to the increasing demand for high-speed data transmission, wireless communication has become more advanced. Unfortunately, the various kinds of impairments that can occur when carrying data symbols through a wireless channel can affect the network performance. Some of the solutions that are proposed to address these issues include channel equalization, and that can be solved through machine learning techniques. In this paper, a hybrid approach is proposed that combines the features of tracking mode and training mode of adaptive equalizer. This method utilizes the concept of machine learning (ML) to classify different environments (highly, medium, low, open space cluttered) based on the measurements of their RF signal. The results of the study revealed that the proposed method can perform well in real-time deployments. The performance of ML algorithms namely Logistic Regression, KNN Classifier, SVM Classifier, Naive Bayes, Decision Tree classifier and Random Forest classifier is analyzed for different number of samples such as 10, 50 and 100. The performance of these algorithms is evaluated by comparing their accuracy, sensitivity, specificity, F1 score and Confusion Matrix. The objective of this study is to demonstrate that a single ML algorithm cannot perform well in all kinds of environments. In order to choose the best algorithm for a given environment, the decision device has to analyze the various factors that affect the performance of the system. For instance, the random forest classifier performed well in terms of accuracy (100 percent), specificity (100 percent), sensitivity (100 percent), and F1_score (100 percent). On the other hand, the logistic regression algorithm did not perform well in low cluttered environment.

Keywords- Classification; Machine Learning; SVM; Cognitive radio; Spectrum sensing.

I. INTRODUCTION

The rapid emergence and evolution of new wireless communication technologies such as the Internet of Things and virtual reality are expected to have a significant impact on the future of wireless communications. To meet the increasing demand for high-speed data, a higher bandwidth is required. Due to the complexity of these new applications, the development of new wireless communication systems has become more challenging. For instance, the low-latency requirements of large-scale networks have become an issue [1], [2]. Due to the increasing number of wireless communication applications, the fading of the channel is becoming worse. This issue usually occurs due to the lack of bandwidth [3]. The effect of phase and amplitude distortion and the limited band channel are some of the factors that cause the inter symbol interference (ISI).

The evolution of the channel distortion is a time-related phenomenon that affects the performance of mobile communication. Therefore, the adaptive equalizers should be able to recognize the various characteristics of the channel [4-6]. To achieve channel equalization, an adaptive filter first needs to adjust the taps weight. This process can be carried out through a training sequence mode, which is designed to adjust the coefficients according to the specific criteria determined by the algorithm. Therefore, in order to maintain a reliable transmission of data, an adaptive equalizer must be implemented in the receiver. This type of device is usually used in mobile transmissions since the channel model is unknown. The best performance of this type of adaptive equalizer is usually obtained with the use of non-linear structures [7].

Due to the remarkable success of machine learning within various fields, including computer science, the increasing number of people interested in using it to address wireless communication issues has been attributed to the machine learning (ML)'s potential [8]. It has excelled in various applications, such as image detection [9] and complex games [10]. ML can learn the complicated relationships among variables, particularly those that are difficult to model accurately using mathematical models [9]. This makes it an ideal tool for developing wireless communication systems that do not require knowledge of existing principles. Due to the various advantages of ML, it has been widely used in the development of wireless communication systems [10]. For instance, it has been widely used in the design of physical layer networks for the Internet of Things (IoT) and 5G cellular networks [11], [12].

The equalization problem can be considered as a classification problem that requires an optimal solution based on the Bayes theory. However, the channel model is not known, it implies that the states are not available at the receiver [13]. Some classification algorithms that can be used to estimate these states include the K-means [14, 15], competitive algorithms [16], and the orthogonal least squares (OLS) algorithm [17]. Although the K-means algorithm is relatively simple to implement, it suffers from poor performance when it comes to time-varying channels. On the other hand the OLS algorithm is more efficient but its computational complexity is more.

A literature survey revealed that deep learning techniques can offer better channel equalization compared to traditional methods [18] - [22], in wireless communication. It is believed that ML algorithms are more advantageous than DL when it comes to adaptive channel equalization. Therefore, in our contribution, we focused much on ML algorithms due to their advantages over DL, as follows

- An ML model is simple and collects the data and learns from the data. With time, it becomes more sophisticated and trained as it continues to consume and learn from the collected information. On the other hand, the structure of a neural network (deep learning) is complex. It involves moving the data through several interconnected layers, each of which has its own classification criteria.
- Machine learning models are constantly learning through new experiences and data. This allows them to identify patterns in the collected information. Data is the only input layer in a model, but there are multiple layers in a neural network.

Due to the complexity of wireless communication algorithms, ML can be beneficial in developing effective channel equalizers. In addition to being able to design efficient and accurate channel models, the capabilities of ML-based equalizers can also be utilized in diverse environments. The goal of this study is to investigate the performance of ML-based channel equalization in a complex physical layer structure. In contrast to other studies that extensively explore the use of ML in wireless communication, this paper focuses on the design methodology for implementing the ML approach in wireless communication. It provides a comprehensive overview of the various aspects of the design process, including the implementation of an ML-based algorithm and an ML architecture. This paper identifies the user signals and rejects the multipath interference using ML approach.

1.1 Multipath Interference

All the symbols in the receiver are independent of one another when all is well. However, when a symbol interacts with another, its waveform can corrupt the value of the nearby symbol, which makes it difficult to interpret the message. Interference between symbols is referred to as inter-symbol interference (ISI) [23]. When there is no interference from a source to the receiver, the system's impulse response is affected by the message's recovery. The delay and amplitude of this phenomenon are dependent on the transmission losses. When this occurs, the single spike in the channel is duplicated for each path in it, and the number of non-zero terms in its impulse response increases. A linear filter or finite-impulse response can be used to model the channel. The delay spread is the amount of time that the reflections take to arrive.

An equalizer is a type of filter that aims to counteract the effects of the channel. It should also unscatter the impulse response, which is the goal of the design. This concept can be stated as the objective of the design to create a combined channel equalizer with a single spike. This type of optimization problem can be solved by implementing various techniques. For instance, if the transmitted signal is $x(t)$, then the value of the received signal is $y(t)$ which is the addition of delayed signals (T_1, T_2, T_3, $T_4 \ldots T_N$) with the reflections' strength (r_1, r_2, r_3, $r_4, \ldots r_N$) can be written as

$$y(t) = r_1 x(t - T_1) + r_2 x(t - T_2) + \cdots + r_n x(t - T_N) + w(t) \qquad (1)$$

A transmission channel is modeled as a finite impulse-response filter that takes into account the delay spread in the physical medium over which the response is non-zero. This type of transmission channel is usually modeled digitally. The Eq (1) approximation is based on the assumption that the sampling period is fixed as follows

$$y(kT_s) = r_1 x(kT_s) + r_2 x((k-1)T_s) + \cdots + r_n x((k-n)T_s) + w(kT_s) \qquad (2)$$

To properly represent the system, the total time that the impulse response takes must be larger than the maximum delay. Since the delay doesn't depend on the period T_s' symbol period, smaller T_s should be filtered more.

II. ADAPTIVE EQUALIZATION

It is used to adjust the channel's time-varying characteristics automatically. It can be implemented on a regular basis or it can be periodically adjusted. The receiver can be notified about the adjustments by transmitting a short training sequence or preamble. The continuous adjustment process involves replacing the training sequence with a set of data symbols that are estimated from the output of the equalizer. This method is referred to as "decision directed." There are two different modes of operation for the adaptive equalizer: training and tracking.

(A) Training Mode

The receiver's equalizer is first set by the transmitter in a training sequence that's fixed. This sequence can be a pseudo random signal or a prescribed bit pattern. Following the sequence, the user's data is sent.

(B) Tracking mode

The algorithm uses the users' data to track the channel change. It then continuously changes the characteristics of the filter over time. This feature is widely used in TDMA systems.

In this case, the output of the equalizer is expressed as Eq(3) after taking into account the L adjustable coefficients [24].

$$y(\mathrm{n}) = \sum_{n=0}^{L-1} h(n)x(k+T-n) \tag{3}$$

The nominal delay between the transmission of the signal and the processing of it through the filter is *T*. The equalizer is then trained by transmitting a sequence of data known as *T*(*n*) (see Figure 1). When the output of the equalizer is compared with

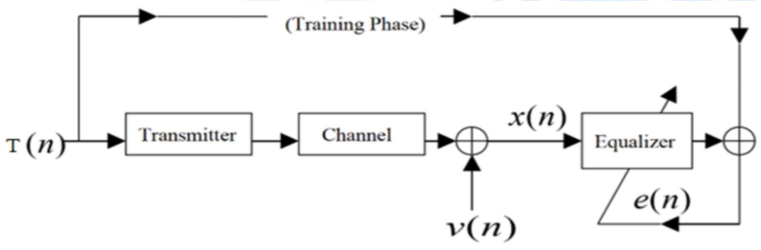


Figure 1. Adaptive Channel equalization

that of *T*(*n*), an error *e*(*n*) is generated to optimize the filter's coefficients.

We can minimize the quantity by selecting the coefficients h(n) if we adopt the least squares criterion again.

$$\Psi_L = \sum_{n=0}^{N}[T(n)-y(n)]^2 = \sum_{n=0}^{N}\left[T(n) - \sum_{n=0}^{N} h(n)x(k+T-n)\right]^2 \tag{4}$$

The optimization results in a set of linear equations as follows

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

The $r_{xx}(l)$ and $r_{tx}(l)$ function is used to determine the correlation between the given sequence x(n) and the desired sequence *T*(*n*). Although the Eq(5) solution can be obtained recursively, we can observe that the coefficients of the equation provide the initial adjustments for the equalizer. The information sequence *x*(*n*) is transmitted by the transmitter after a brief training period. The adjustment of the coefficients is carried out in an adaptive manner to keep track of the channel's time variations. As shown in Figure 1, this method usually involves treating the decisions made by the decision device correctly and using the errors in place of the references *T*(*n*). This works well when the errors happen frequently. For instance, if error occurs rarely, the coefficients should only get slightly mis-adjusted.

III. EXPERIMENTAL SETUP-SOFTWARE DEFINED RATIO

Instead of categorizing an environment into different types of structures, existing literature [25]–[28] uses a classification approach that focuses on the surrounding environment. For instance, the literature presents an approach that categorizes the environment into three categories: outdoor, semi-outdoor, and indoor. One of the most common approaches to categorizing an environment is by collecting data from various cell towers [25]. This method was not very popular since it requires a large amount of data. Other methods, such as sound signals and cell identity maps, can also be used to classify environments [26], [27]. A semi-supervised learning method was then developed [28] to classify environments such as indoor/outdoor.

The goal of this work is to use machine learning techniques to classify different types of environments in real-time. In this software defined radio [29] – [30] experimental setup, USRP N210 is used as a hardware and GNU Radio is used as software. As shown in Figure 2, we can see the various components of the work, including two USRP hardware boards and a couple of GNU Radio-installed computers. The classification process was performed by splitting each square area into uniform grid patterns with a spacing of 12.5cm, which is the wavelength of WiFi bands (operating frequency f = 2.4GHz) that are part of the IEEE 802.11g standard. It should be noted that the measurements are performed in real- time in a stationary environment [31]. The signals are analyzed using various parameters such as frequency, amplitude, phase, and signal to noise ratio from different distances in cluttered environments such as highly, medium, low and open space cluttered environments for 10, 50 and 100 number of samples. A sample dataset presented in Table 1 containing the 7 user signals and 3 noise outputs from different environments. The signal amplitude and frequency are proportional to the SNR value. Repeated experiments are performed on the same set of attributes for varying numbers of samples (50 and 100) and the outputs are analyzed using different ML frameworks.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8473
***Article Received:** 24 July 2023 **Revised:** 15 September 2023 **Accepted:** 02 October 2023*

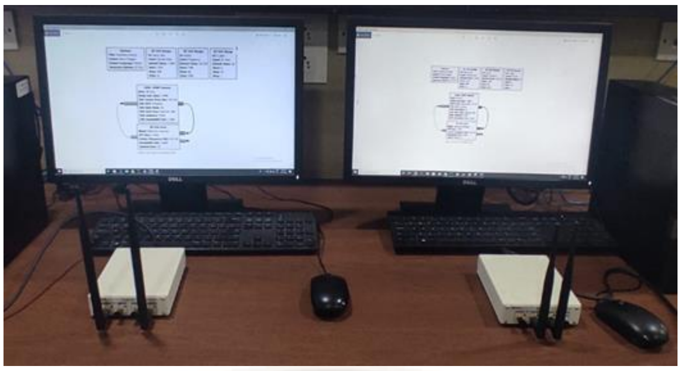


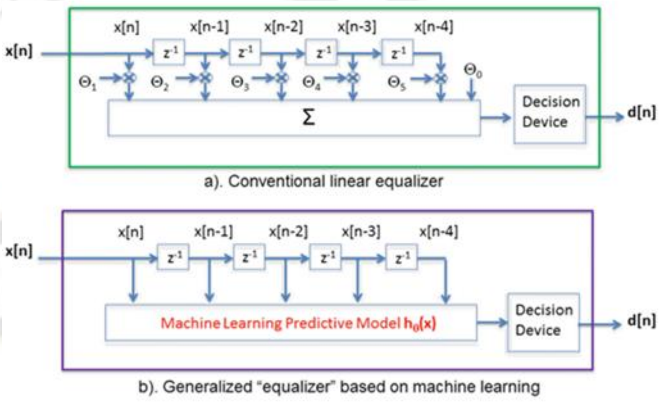
Figure 2: Experimental setup

Table 1: Sample dataset for 7 number of user signals and 3 noise signals

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**
10001	20	45	97	154	2400000000	90	35	1
10002	17	41	94	151	2400166667	87	320	1
10003	14	38	89	146	2400333333	83	67	1
10004	12	35	86	139	2400500000	81	45	1
10005	10	33	85	131	2400666667	76	53	0
10006	9	31	82	128	2400833333	73	24	0
10007	7	28	73	119	2401000000	70	18	1
10008	5	23	68	112	2401166667	64	27	0
10009	3	22	57	105	2401333333	58	60	1
10010	2	21	46	98	2401500000	52	13	1

IV. PROPOSED SYSTEM MODEL

The block diagram of a conventional linear equalizer is shown in Figure 3(a). It can be used as a linear predictor or a linear regression, where the features are derived from N input samples [32]. The predictive model is also defined by the N+1 parameters that were learned during training. The output of the linear equalizer is sent to a decision device, whose algorithm takes into account the minimum distance between the nearest symbol and the center of the circle. We then generalize the algorithm to a structure shown in Figure 3(b). The goal of this method is to replace the FIR filter processing and predict the output symbols directly. Although the algorithm's input features are the same as those derived from N samples, it can be programmed to either follow a regression or a classification method that predicts the exact symbols. In our contribution, we focus on classification rather than regression due to its advantages.


x[n]
x[n-1]
x[n-2]
x[n-3]
x[n-4]
z^{-1}
Θ_1
Θ_2
Θ_3
Θ_4
Θ_5
Θ_0
Σ
Decision Device
d[n]
a). Conventional linear equalizer
Machine Learning Predictive Model $h_\theta(x)$
b). Generalized "equalizer" based on machine learning
Figure 3. (a) Conventional equalizer (b) ML based equalizer

The workflow for the proposed system is shown in Figure 4. The values of the signals sent by the transmitter are then counted and the characteristics of these are noted. Some of the required python libraries include pandas, numpy, sklearn, and matplotlib are imported. The resulting dataset is then uploaded to a python console with signal characteristics for highly cluttered environment. The attributes are then separated into outputs and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8473
***Article Received:** 24 July 2023 **Revised:** 15 September 2023 **Accepted:** 02 October 2023*

inputs. The presence of the user signal or noise is considered as the outputs. The collected dataset is split into two groups: a training set and a test set. The multiple ML algorithms that are used for training are KNN, Naive Bayes, Random Forest, SVM, Decision Tree, and Logistic Regression. The performance of these models is evaluated according to their accuracy, sensitivity, specificity, precision, Confusion Matrix and F1_score. For each environment, the training set is analyzed and evaluated according to its requirements. The decision device then chooses an algorithm that has the best performance in that environment. The equalizer then adjusts the weights based on the received signal. Finally, the equalizer sends to the further processing in the receiver section as decoding, demodulation and so on.

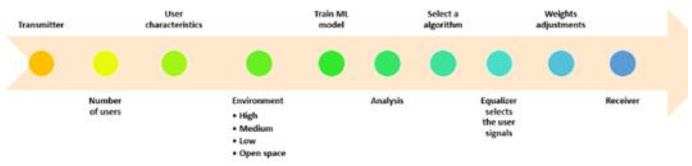


Figure 4. Workflow of the proposed system.

V. MACHINE LEARNING ALGORITHMS

5.1 Logistic Regression

The process of linear regression is used to describe the relationship between a single variable or set of independent variables and one or more dependent binary variables. It can also explain the characteristics of a given data set. Unfortunately, with linear regression, the predicted values are not always within a range [33]. The Logistic Regression method can be used to solve these problems. It converts the best fit linear regression line into a S-curve curve by using the sigmoids function. The first step is to assume that the given function, which is $p(x)$, is the linear function. But, since it is an absolute, unconstrained linear equation, its value should not deviate from the value of 0 or 1. To solve this issue, we will assume that the log $p(x)$ function is a linear function that follows the path of x. We will then use the logit transformation to get the value between the range of 0 and 1.

$$\log \frac{p(x)}{1-p(x)} = \beta_0 + \beta . x \qquad (6)$$

After solving for $p(x)$

$$p(x) = \frac{e^{\beta_0 + \beta}}{e^{\beta_0 + \beta + 1}} \qquad (7)$$

To make a linear classifier, the threshold is set at which it will automatically classify a set of data. For instance, if the prediction of the logistic regression for the given set of data is 0.5, then it can minimize the classification rate by predicting that the data is 1.

5.2 KNN Classifier

The KNN algorithm is a supervised learning framework that's used to classify data according to the way it's neighbours are ranked. It stores all the available cases and considers new cases according to their similarity. Another parameter in the framework is the number of nearest neighbours that will be included in the voting process [34], [35]. For a given value of K, the algorithm will find the nearest neighbour of an unseen data point. It will then assign the class with the highest number of points from all the classes of K neighbours to the data point. For distance metrics, the algorithm will use the metric known as the Euclidean that is given by

$$d(a, a') = \sqrt{(a_1 - a_1')^2 + \cdots + (a_n - a_n')^2} \qquad (8)$$

The given class is the largest probability based on the input x.

$$P(B = j | x = a) = \frac{1}{k} \sum_{i \in A} I(B^i = j) \qquad (9)$$

The method for regression will be similar to that of the previous ones, except that instead of taking the neighbors' classes, we will take the target's value and find it for the unseen datapoint using an average, mean, or any other suitable function.

5.3 SVM classifier

The SVM is a widely used and effective method for solving the non-linear equations. It is mainly done by using the kernel method, which makes it easy to solve. The main objective of this algorithm is to find the ideal hyperplane that can efficiently separate the two components. The input instances of a given model are represented by binary class labels and data instance in the SVM. These are usually addressed by a couple of variables. For instance, the R^n information example is addressed by a pair of values, namely, (x_k, y_k) [36], [37]. The training information T for the given model and its hyperplane H_p can be defined in the following way.

$$T = \{(x_k, y_k) \in \Re^n, \quad k = 1, \ldots. n\} \qquad (10)$$

$$H_p : w_g . x_i + \tau \qquad (11)$$

Where the value of y_k is determined by the boundary between various classes of data that are defined by the vector weighing vector 'w_g and scalar threshold 'τ'. This is done through the use of a statistical method known as SVM. There are two kinds of SVM: linearly [38] and non-linear classifications [39].

5.4 Naïve Bayes Algorithm

The Naive Bayes classification method is based on the Bayes Theorem, which states that the features that predict a target value are independent from each other [40]. It then

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8473
***Article Received:** 24 July 2023 **Revised:** 15 September 2023 **Accepted:** 02 October 2023*

chooses the class with the highest probability of success. Although it is commonly used for various applications, such as natural language processing, it is very effective when dealing with NLP problems. The Naive Bayses classifier takes into account that the various features that are used to predict a target do not interact with each other and are independent of one another. Although the independence assumption used in the

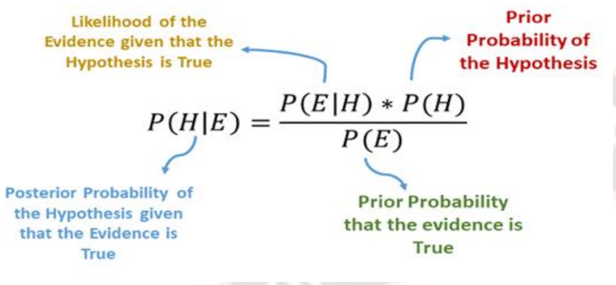

Likelihood of the Evidence given that the Hypothesis is True
Prior Probability of the Hypothesis
$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$
Posterior Probability of the Hypothesis given that the Evidence is True
Prior Probability that the evidence is True

Figure 5. Bayes Theorem.

Naive Bayes method is never correct, it is still very effective in practice. The Bayes Theorem states that the probability of an occurrence is based on the prior knowledge of factors that can affect that event as shown in Figure 5.

5.5 Decision Tree classifier

A decision tree is a set of rules or decisions that are usually dependent on a single variable at a given time. These trees refine the level of detail in each iteration to produce the final label, also known as the leaf node. The information in a decision tree is a measure of purity, and the impurities can be measured in various ways, such as the Gini Index and Entropy [41].

5.5.1 Entropy

The amount of information that's needed to properly describe a data set is known as Entropy. If data is homogeneous, then its total number of elements is zero, while if it's all similar, then its total number of elements is one. This means that if all elements are in the same region, then its total number of elements is zero, while if they are equally divided, then its number of elements is one. Mathematically the entropy can be defined as

$$Entropy = -\sum_{i=1}^{n} p_x * \log(p_x) \tag{12}$$

5.5.2 Gini index/Gini impurity

The node's impurities are measured using the Gini index, which is between 0 and 1. The sample is homogeneous, while the elements are similar. On the other hand, the Gini index for value 1 indicates that the elements are unequal. It is defined as,

$$Gini\ Index = 1 - \sum_{i=1}^{n} {p_x}^2 \tag{13}$$

The objective of impurity is to measure the homogeneity of data. If the data is homogeneous, then it will belong to the same class and tree as well.

5.6 Random Forest

The Random Forest is a framework that combines multiple classifiers in order to improve the efficiency of a model. It uses the ensemble learning process to solve complex problems. According to its name, the program takes the average of the various decision trees in the given dataset to improve its predictive accuracy [42]. Instead of relying on a single decision tree, the Random Forest takes the predicted outcomes from each tree and predicts the results based on the majority of the votes.

VI. RESULTS AND DISCUSSION

Different machine learning algorithms, such as KNN, SVM, Naive Bayes, Decision Tree classifier and Random Forest classifier and Logistic Regression, are evaluated for their adaptive equalizer performance in different environments (highly, medium, low and open space cluttered). The results of the study are presented in three Tables: 1, 2, and 3 for number of samples 10, 50 and 100, respectively. The performance of these algorithms is evaluated in terms of accuracy, sensitivity, Precision, specificity, F1_score and Confusion Matrix.

A classification matrix of size 2x2 is known as the confusion matrix, and it shows the predicted values on one side of the axis and the actual ones on the other as shown in Figure 6. Although the columns represent the predictions of an algorithm, the rows can identify which ones were wrong. A false selection is when a prediction is made that is not true, while a true selection is when the prediction is correct. The quality of a model can be measured by using the confusion matrix. This paper shows that the training set is 75% and the test size is 25% of the total input data set that means 25% of 100 is 25 which is a sum of test size 8+5+1+11 (see Table 3, Row 1) and so on. In terms of accuracy, it is better to have a symmetric dataset with close false positives and false negatives.

If the cost of false negatives and false positives are different, then F1 is the best option. F1 is also ideal for situations where the class distribution is uneven. The precision is determined by how sure they are that they are of their true positives while the recall is determined by how confident they are that they are not missing any. If the idea of recalling false positives is better than recalling false negatives, then choose sensitivity. This is because recalling false alarms is preferable to saving false negatives if they are not accepted. If we want to ensure that all true negatives are covered, choose specificity. This means that we don't want false alarms.

It is observed from Table 2 that all algorithms have shown similar performance in all four environments.

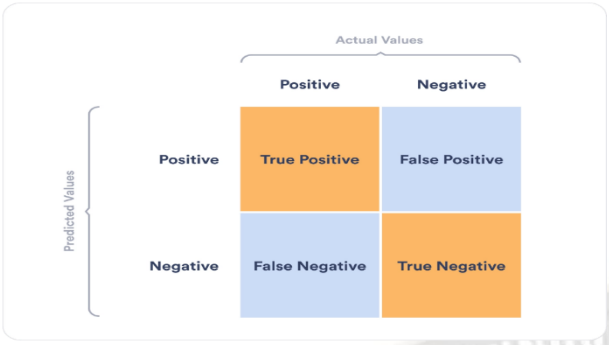


Figure 6. Confusion Matrix

Random Forest classifier has shown a 100% performance compared to all other algorithms and it is highlighted in the table for better visibility.

It is observed from Table 3 that all algorithms have shown different performance in different environment. Decision Tree classifier has shown best performance in terms of Accuracy (61.5 %), Precision (66.6 %), Specificity (66.6 %), F1_score (64 %), whereas Logistic Regression has shown better Sensitivity of 71.4% for highly cluttered environment. Decision tree classifier has shown best performance in terms of Accuracy (61.5 %), Precision (62.5 %), Specificity (71.4 %), Sensitivity (50 %) and F1_score (66.6 %), whereas KNN has shown poor performance for medium cluttered environment. Random forest classifier has shown best performance in terms of Accuracy (63 %), Precision (66.6 %), Specificity (71.4 %), Sensitivity (66.6 %) and F1_score (66 %), whereas KNN and Naive Bayes have shown poor performance for low cluttered environment. Random forest classifier has shown best performance in terms of Accuracy (61.5 %), Precision (66.6 %), Specificity (66.6 %), F1_score (61.5 %), whereas KNN classifier has shown better Sensitivity of 71.4% for open space cluttered environment. Therefore, it is concluded from the Table 3 that Decision tree classifier is preferable for highly and medium cluttered environments, whereas Random forest classifier is preferable for low and open space cluttered environments for 50 number of samples.

Table 2: Performance analysis of different ML algorithms for 10 number of samples over different environments.

Environment	Type of algorithm	Accuracy	Precision	Sensitivity	Specificity	F1_score	Confusion Matrix
highly/ Medium/ Low/ Open space cluttered	Logistic Regression	66	66	100	0	80	$\begin{bmatrix}0&1\\0&2\end{bmatrix}$
	KNN Classifier	66	66	100	0	80	$\begin{bmatrix}0&1\\0&2\end{bmatrix}$
	SVM Classifier	66	66	100	0	80	$\begin{bmatrix}0&1\\0&2\end{bmatrix}$
	Naive Bayes	66	66	100	0	80	$\begin{bmatrix}0&1\\0&2\end{bmatrix}$
	Decision Tree	66	100	50	100	66	$\begin{bmatrix}1&0\\1&1\end{bmatrix}$
	Random Forest	100	100	100	100	100	$\begin{bmatrix}1&0\\0&2\end{bmatrix}$

Table 3: Performance analysis of different ML algorithms for 50 number of samples over different environments.

Environment	Type of algorithm	Accuracy	Precision	Sensitivity	Specificity	F1_score	Confusion Matrix
highly cluttered	Logistic Regression	61	62.5	71.4	50	66.6	$\begin{bmatrix}3&3\\2&5\end{bmatrix}$
	KNN	53.8	55.5	70	33.3	62.5	$\begin{bmatrix}2&4\\2&5\end{bmatrix}$
	SVM	46.1	50	42.8	50	46.1	$\begin{bmatrix}3&3\\4&3\end{bmatrix}$
	Naive Bayes	46.1	50	42.8	50	46.1	$\begin{bmatrix}3&3\\4&3\end{bmatrix}$
	Decision Tree	61.5	66.6	67.1	66.6	64	$\begin{bmatrix}4&2\\2&5\end{bmatrix}$
	Random Forest	46.1	50	57.1	33.3	53.3	$\begin{bmatrix}2&4\\3&4\end{bmatrix}$
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	$\begin{bmatrix}4&2\\4&3\end{bmatrix}$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8473
***Article Received:** 24 July 2023 **Revised:** 15 September 2023 **Accepted:** 02 October 2023*

	Naive Bayes	46.1	50	42.8	50	46.1	$\begin{bmatrix}3 & 3\\4 & 3\end{bmatrix}$
	Decision Tree	61.5	62.5	71.4	50	66.6	$\begin{bmatrix}3 & 3\\2 & 5\end{bmatrix}$
	Random Forest	46.1	50	42.8	50	46.1	$\begin{bmatrix}3 & 3\\4 & 3\end{bmatrix}$
Low cluttered	Logistic Regression	61.5	62.5	71.4	50	66.6	$\begin{bmatrix}3 & 3\\2 & 5\end{bmatrix}$
	KNN	53.8	57.1	57.1	50	57.1	$\begin{bmatrix}3 & 3\\3 & 4\end{bmatrix}$
	SVM	61.5	62.5	71.4	50	66.6	$\begin{bmatrix}3 & 3\\2 & 5\end{bmatrix}$
	Naive Bayes	53.8	57.1	57.1	50	57.1	$\begin{bmatrix}3 & 3\\3 & 4\end{bmatrix}$
	Decision Tree	61.5	62.5	71.4	50	66.6	$\begin{bmatrix}3 & 4\\2 & 4\end{bmatrix}$
	Random Forest	63	66.6	71.4	66.6	66	$\begin{bmatrix}4 & 2\\3 & 4\end{bmatrix}$
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	46.1	50	42.8	50	46.1	$\begin{bmatrix}3 & 3\\4 & 3\end{bmatrix}$
	Naive Bayes	53.8	57.1	57.1	50	57.1	$\begin{bmatrix}3 & 3\\3 & 4\end{bmatrix}$
	Decision Tree	30.7	33.3	28.5	33.3	30.7	$\begin{bmatrix}2 & 4\\5 & 2\end{bmatrix}$
	Random Forest	61.5	66.6	57.1	66.6	61.5	$\begin{bmatrix}4 & 2\\3 & 4\end{bmatrix}$

It is observed from Table 4 that all algorithms have shown different performance in different environment. Decision Tree classifier has shown best performance in terms of Accuracy (92 %), Precision (100 %), Specificity (100 %), F1_score (90.9 %), whereas Logistic Regression has shown better Sensitivity of 91.6 % for highly cluttered environment. Decision tree classifier has shown best performance in terms of Accuracy (100 %), Precision (100 %), Specificity (100 %), Sensitivity (100 %) and F1_score (100 %), whereas KNN and SVM classifiers have shown poor performance for medium cluttered environment. Random forest classifier has shown best performance in terms of Accuracy (100 %), Precision (100 %), Specificity (100 %), Sensitivity (100 %) and F1_score (100 %), whereas Logistic regression algorithm has shown poor performance for low cluttered environment. Decision tree classifier has shown best performance in terms of Accuracy (100 %), Precision (100 %), Specificity (100 %), Sensitivity (100 %) and F1_score (100 %), whereas SVM and Naive Bayes classifiers have shown poor performance for open space cluttered environment. Therefore, it is concluded from the Table 3 that Decision tree classifier is preferable for highly, medium and open source cluttered environments, whereas Random forest classifier is preferable for low cluttered environments for 100 number of samples.

From the obtained results, it is concluded that a single ML algorithm is not suitable to all kind of environments, and a decision device has to choose an efficient algorithm based on the environment and samples. Random forest algorithm is selected for all kind of environments if the number samples are very less like 10. Decision tree classifier is preferable for highly and medium cluttered environments, whereas Random forest classifier is preferable for low and open space cluttered environments for more number of samples. It is also observed that as number of samples increases, the performance values are also increased.

Table 4: Performance analysis of different ML algorithms for 100 number of samples over different environments.

Environment	**Type of algorithm**	**Accuracy**	**Precision**	**Sensitivity**	**Specificity**	**F1_score**	**Confusion Matrix**
Highly cluttered	Logistic Regression	76	68.7	91.6	61.5	78.5	$\begin{bmatrix}8 & 5\\1 & 11\end{bmatrix}$
	KNN	64	60	75	53.8	66.6	$\begin{bmatrix}7 & 6\\3 & 9\end{bmatrix}$
	SVM	72	66.6	83.3	61.5	74	$\begin{bmatrix}8 & 5\\2 & 10\end{bmatrix}$
	Naive Bayes	76	71.4	83.3	69.2	76.9	$\begin{bmatrix}9 & 4\\2 & 10\end{bmatrix}$
	Decision Tree	92	100	83.3	100	90.9	$\begin{bmatrix}13 & 0\\2 & 10\end{bmatrix}$

	Random Forest	88	100	75	100	85.7	$\begin{bmatrix} 13 & 0 \\ 3 & 9 \end{bmatrix}$
Medium cluttered	Logistic Regression	76	68.7	91.6	61.5	78.5	$\begin{bmatrix} 8 & 5 \\ 1 & 11 \end{bmatrix}$
	KNN	72	64.7	91.6	53.8	75.8	$\begin{bmatrix} 7 & 6 \\ 1 & 11 \end{bmatrix}$
	SVM	72	66.6	83.3	61.5	74	$\begin{bmatrix} 8 & 5 \\ 2 & 10 \end{bmatrix}$
	Naive Bayes	100	100	100	100	100	$\begin{bmatrix} 13 & 0 \\ 0 & 12 \end{bmatrix}$
	Decision Tree	100	100	100	100	100	$\begin{bmatrix} 13 & 0 \\ 0 & 12 \end{bmatrix}$
	Random Forest	92	85.7	100	84.6	92.3	$\begin{bmatrix} 11 & 2 \\ 0 & 12 \end{bmatrix}$
Low cluttered	Logistic Regression	60	56.2	75	46.1	64.2	$\begin{bmatrix} 6 & 7 \\ 3 & 9 \end{bmatrix}$
	KNN Classifier	84	78.5	91.6	76.9	84.6	$\begin{bmatrix} 10 & 3 \\ 1 & 11 \end{bmatrix}$
	SVM	68	62.5	83.3	53.8	71.4	$\begin{bmatrix} 7 & 6 \\ 2 & 10 \end{bmatrix}$
	Naive Bayes	68	64.2	75	61.5	69.2	$\begin{bmatrix} 8 & 5 \\ 3 & 9 \end{bmatrix}$
	Decision Tree	92	100	83.3	100	90.9	$\begin{bmatrix} 13 & 0 \\ 2 & 10 \end{bmatrix}$
	Random Forest	100	100	100	100	100	$\begin{bmatrix} 13 & 0 \\ 0 & 12 \end{bmatrix}$
Open space	Logistic Regression	76	68.7	91.6	61.5	78.5	$\begin{bmatrix} 8 & 5 \\ 1 & 11 \end{bmatrix}$
	KNN	92	85.7	100	84.6	92.3	$\begin{bmatrix} 11 & 2 \\ 0 & 12 \end{bmatrix}$
	SVM	72	69.2	75	69.2	71.9	$\begin{bmatrix} 9 & 4 \\ 3 & 9 \end{bmatrix}$
	Naive Bayes	72	69.2	75	69.2	71.9	$\begin{bmatrix} 9 & 4 \\ 3 & 9 \end{bmatrix}$
	Decision Tree	100	100	100	100	100	$\begin{bmatrix} 13 & 0 \\ 0 & 12 \end{bmatrix}$
	Random Forest	92	85.7	100	84.6	92.3	$\begin{bmatrix} 11 & 2 \\ 0 & 12 \end{bmatrix}$

CONCLUSION

The results indicate that the implementation of traditional methods in communication systems can be challenging compared to the use of ML techniques. When channel equalization is regarded as a classification issue, the ML techniques can be used to solve it, resulting in simpler receiver setups. The paper aims to introduce the methodology and showcase the advantages of using machine learning in classification. It also shows how it can be utilized in indoor environment assessment. The results were obtained by means of line-of-sight measurements. Based on the results, it has been concluded that there is a limit to the number of ML algorithms that can be used in a given environment. For instance, if the number of samples is less than 10, the random forest algorithm is preferred. The decision tree classifier is better suited for medium and highly cluttered environments, while the random forest algorithm is preferred for smaller and open space environments. It is also observed that the performance of the algorithm increases as the number of samples increases.

The paper's findings will be further explored in the future by extending the ML techniques to other scenarios, such as time-varying channels and non-line of sight measurements. Machine learning will be utilized to extract the features of these sophisticated scenarios.

Compliance with Ethical Standards

Disclosure of potential conflicts of interest: We have no conflicts of interest to disclose.

Research involving Human Participants and/or Animals: We declare that our research does not include any Human and/or Animals Participants.

Informed consent: It is not applicable to our research.

REFERENCES

[1] Dai, Linglong, et al. "Deep learning for wireless communications: An emerging interdisciplinary paradigm." *IEEE Wireless Communications* 27.4 (2020): 133-139.

[2] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5G,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[3] Mohammed, Bashar A., and Siddeeq Y. Ameen. "Impact of Different Training Mode on Adaptive Equalization Techniques for MIMO-OFDM System." *Article in Communications on Applied Electronics, Volume 7 – No.2,* · May 2017. DOI: 10.5120/cae2017652603

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8473
Article Received: *24 July 2023* ***Revised:*** *15 September 2023* ***Accepted:*** *02 October 2023*

[4] Malik, Garima, and Amandeep Singh Sappal. "Adaptive equalization algorithms: an overview." *International journal of advanced computer science and applications* 2.3 (2011).

[5] Ali, Farman, et al. "Adaptive equalization for dispersion mitigation in multi-channel optical communication networks." *Electronics* 8.11 (2019): 1364.

[6] Acharya, Upendra Kumar, and Sandeep Kumar. "Genetic algorithm based adaptive histogram equalization (GAAHE) technique for medical image enhancement." *Optik* 230 (2021): 166273.

[7] Proakis, J.G.: 'Digital communications in communications' (MGGrawHill, New York, 2003, 2nd edn.)

[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, Jan. 2016

[10] Z. Qin, H. Ye, G. Y. Li, and B. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019.

[11] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communications in the Internet of Things: A comprehensive survey," *Ad Hoc Netw.*, vol. 93, 2019, Art. no. 101913.

[12] R. Li et al., "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 175–183, Oct. 2017

[13] Assaf, Rima, et al. "Efficient classification algorithm and a new training mode for the adaptive radial basis function neural network equaliser." *IET communications* 6.2 (2012): 125-137.

[14] Mulgrew, B., Grant, P., Chen, S.: 'A clustering technique for digital communications channel equalization using radial basis function networks', *IEEE Trans. Neural Netw*., 1993, 4, (4), pp. 570–579

[15] Chen, S., Labib, K., Hanzo, L.: 'Clustering-based symmetric radial basis function beamforming', *IEEE Signal Process. Lett.*, 2007, 14, (9), pp. 589–592

[16] El Assad, S., Miclau, N., Hamad, D.: 'Apprentissage compe´titif pour re´seaux neuronaux RBF utilise´s en e´galisation adaptative des canaux de transmission non line´aires'. *Int. Conf. on Image and Signal Processing, ICISP,* Agadir, Morocco, 2003, pp. 330–339y

[17] Chen, S., Cowan, C.F.N., Grant, P.M.: 'Orthogonal least squares learning algorithm for radial basis function networks', *IEEE Trans. Neural Netw*., 1991, 2, (2), pp. 302–309

[18] Zhao, Yiheng, Peng Zou, and Nan Chi. "3.2 Gbps underwater visible light communication system utilizing dual-branch multi-layer perceptron based post-equalizer." *Optics Communications* 460 (2020): 125197.

[19] Panda, Sashmita, and Ganapati Panda. "On the development and performance evaluation of improved radial basis function neural networks." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.6 (2021): 3873-3884.

[20] Xu, Zhaopeng, et al. "Computational complexity comparison of feedforward/radial basis function/recurrent neural network-based equalizer for a 50-Gb/s PAM4 direct-detection optical link." *Optics Express* 27.25 (2019): 36953-36964.

[21] Zhang, Lin, and Lie-Liang Yang. "Machine learning for joint channel equalization and signal detection." *Machine Learning for Future Wireless Communications* (2020): 213-241.

[22] Allander, Martin. "Channel Equalization Using Machine Learning for Underwater Acoustic Communications." *Master of Science Thesis in Electrical Engineering Department of Electrical Engineering*, Linköping University, 2020.

[23] Mohanty, Durgesh Nandini, and Debashree Mohapatra. Performance Evaluation of Adaptive Equalizer in a Communication System. Diss. 2009.

[24] Gurrapu, Omprakash. "Adaptive filter algorithms for channel equalization." *Thesis submitted to University of Borås*, Sweden, (2009).

[25] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen, "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, 2012, pp. 113–126.

[26] 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, Jan. 2015.

[27] 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, Dec. 2015.

[28] 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, 2014, pp. 280–294*

[29] Reddy, Bathula Siva Kumar, Kiran Mannem, and K. Jamal. "Software defined radio based non-orthogonal multiple access (NOMA) systems." *Wireless Personal Communications* 119.2 (2021): 1251-1273.

[30] Reddy, Bathula Siva Kumar. "Experimental validation of non-orthogonal multiple access (NOMA) technique using software defined radio." *Wireless Personal Communications* 116.4 (2021): 3599-3612.

[31] AlHajri, Mohamed I., Nazar T. Ali, and Raed M. Shubair. "Classification of indoor environments for IoT applications: A machine learning approach." *IEEE Antennas and Wireless Propagation Letters* 17.12 (2018): 2164-2168.

[32] Lyubomirsky, I. "Machine learning equalization techniques for high speed PAM4 fiber optic communication systems." *CS229 Final Project Report, Stanford University* (2015).

[33] Wood SN. Generalized additive models: an introduction with R. *CRC press*; 2017 May 18.

[34] Farahani, Gholamreza. "Black hole attack detection using K-nearest neighbor algorithm and reputation calculation in mobile ad hoc networks." *Security and Communication Networks* 2021 (2021): 1-15.

[35] Xu, Shiwu, et al. "Adaptive residual weighted K-nearest neighbor fingerprint positioning algorithm based on visible light communication." *Sensors* 20.16 (2020): 4432.

[36] Sheykhmousa, Mohammadreza, et al. "Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020): 6308-6325.

[37] Tavara, Shirin. "Parallel computing of support vector machines: a survey." *ACM Computing Surveys (CSUR)* 51.6 (2019): 1-38.

[38] Ghosh, Sourish, Anasuya Dasgupta, and Aleena Swetapadma. "A study on support vector machine based linear and non-linear pattern classification." *2019 International Conference on Intelligent Sustainable Systems (ICISS)*. IEEE, 2019.

[39] Rizwan, Atif, et al. "WR-SVM model based on the margin radius approach for solving the minimum enclosing ball problem in support vector machine classification." *Applied Sciences* 11.10 (2021): 4657.

[40] Shi, Yuxin, et al. "Efficient jamming identification in wireless communication: Using small sample data driven naive bayes

classifier." *IEEE Wireless Communications Letters* 10.7 (2021): 1375-1379.

[41] Ahmad, Hassaan Bin. "Ensemble classifier based spectrum sensing in cognitive radio networks." *Wireless Communications and Mobile Computing* 2019 (2019).

[42] Noshad, Zainib, et al. "Fault detection in wireless sensor networks through the random forest classifier." *Sensors* 19.7 (2019): 1568.