

Article

Seamless Connections: Harnessing Machine Learning for MAC Optimization in Home Area Networks

Bilal Muhammad Khan ^{1,*} and Muhammad Bilal Kadri ² 

¹ Electronics and Power Engineering, National University of Sciences and Technology, Islamabad 44000, Pakistan

² Department of Computer Science, College of Computer Science and Information Systems, Prince Sultan University, Riyadh 11586, Saudi Arabia; mkadri@psu.edu.sa

* Correspondence: bmkhan@pnec.nust.edu.pk

Abstract: The latest technologies and communication protocols are arousing a keen interest in automation, in which the field of home area networks is the most prominent area to work upon toward solving the issues and challenges faced by wireless home area networks regarding adaptability, reliability, cost, throughput, efficiency, and scalability. However, managing the immense number of communication devices on the premises of a smart home is a challenging task. Moreover, the Internet of Things (IoT) is an emerging global trend with billions of smart devices to be connected in the near future resulting in a huge amount of diversified data. The continuous expansion of the IoT network causes complications and vulnerabilities due to its dynamic nature and heterogeneous traffic. In the applications of IoT, the wireless sensor network (WSN) plays a major role, and to take benefits from WSN, medium access control (MAC) is the primary protocol to optimize, which helps in allocating resources to a huge number of devices in the smart home environment. Furthermore, artificial intelligence is highly demanded to enhance the efficiency of existing systems and IoT applications. Therefore, the purpose of this research paper is to achieve an optimized medium access control protocol through machine learning. The machine learning classifier, e.g., random forest (RF) and linear regression model, is adopted for predicting the features of home area networks. The proposed technique is helpful and could overcome the demerits of existing protocols in relation to scalability, throughput, access delay, and reliability and help in achieving an autonomous home area network (HAN).

Keywords: IoT; random forest; linear regression; CSMA; MAC protocols; artificial intelligence; machine learning; home automation; home area network



Citation: Khan, B.M.; Kadri, M.B. Seamless Connections: Harnessing Machine Learning for MAC Optimization in Home Area Networks. *Electronics* **2023**, *12*, 4082. <https://doi.org/10.3390/electronics12194082>

Academic Editors: Apostolis Xenakis, Costas Chaikalas, Ilias K. Savvas and Kosmanos Dimitrios

Received: 16 August 2023

Revised: 10 September 2023

Accepted: 12 September 2023

Published: 29 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Note that each past decade has witnessed phenomenal changes and innovations in communication technologies. Ideas proposed in movies regarding communication technology along with artificial intelligence are now becoming a reality. This rapid growth in autonomous wireless technologies has brought an interest in home automation with the intention of enhancing efficiency, reliability, and throughput with minimized access delay. As discussed in [1], with the rapid increase in demand for portable and rapid communication devices, the wireless sensor network (WSN) plays a vital role in addressing these modern needs. It would not be wrong to state that the wireless sensor network (WSN) is the backbone of every smart city, smart home, and building automation. With a large number of benefits from wireless sensor networks, there are also a huge number of challenges because of insufficient resources and wireless communication. Therefore, in this regard, the medium access control (MAC) protocol is considered the most important factor in designing any WSN application because this protocol assists in enhancing the lifespan and performance of the wireless sensor network.

A major portion of research in home automation networks focuses on the comparison and assorted usage of the MAC protocols in order to achieve better throughput and minimal latency. The architecture of an effective home automation system with artificial intelligence is a key research objective in the context of home area networks (HANs). Moreover, the performance of the IEEE 802.15.4 multi-hop wireless network is analyzed and improved [2], as it is considered a foundation for many applications, such as home automation, industrial sector control, and small-scale grid systems. However, according to existing analysis, homogeneous traffic along with ideal carrier sensing are usually assumed for such smart applications that are far apart from reality when it is a matter of predicting the performance of such a multi-hop network. Furthermore, the Internet of Things (IoT) in its practical manifestation is emerging ubiquitously, but the IoT alone cannot make HANs autonomous. The requirement of intelligence brings the concept of machine learning (ML), a subset of artificial intelligence (AI), into the existing systems to make the systems more efficient and reliable.

1.1. Technical Issues

As WSNs are contributing significantly to making life more comfortable by significantly monitoring and controlling home appliances [3], the MAC protocol is the base that needs to be designed and optimized first before availing advantages from any WSN application. Coordination and management of shared channel resources including transmission data rate, scalability, applicability, and coverage area are all in the control of the medium access control (MAC) protocol layer discussed in [4]. Therefore, the MAC protocol has a great influence on WSNs for the HAN environment. Therefore, it becomes necessary to optimize MAC protocol to take benefit from WSN.

According to the work discussed in [5,6], the CSMA (carrier sense multiple access) protocol is applied in designing an HAN environment, for which periodic traffic flow is mainly considered as not meeting the current demands of automation. Therefore, the appropriate performance of embedded devices combating access to a shared medium through the carrier sense multiple access/collision detection (CSMA/CD-MAC protocol) with aperiodic traffic as well is important.

Moreover, the number of Internet of Things (IoT) devices is increasing day by day with a heterogeneous nature that causes complexity in maintaining connectivity and quality of service in the dynamic nature of situations discussed in [7]. Thus, machine learning is considered to bring artificial intelligence to the IoT devices to automate the system. This research work has shown a new perspective for network flow scheduling in home area network applications.

1.2. Benefits of Wireless Networks with Incorporating Machine Learning Techniques

Advancement in technologies has brought comfort in human life with automatic and wireless techniques being preferred over manual tasks. Home automation is the most prominent application in this regard and is moving forward at a strong pace to achieve the maximum benefits of the wireless network [8]. Every researcher in this field is trying hard to reach the time to market of home area network applications. With artificial intelligence in home area networks [9], the system provides services to each and every user and constantly gathers data to extend the learning process of the model. The smart home should be able to predict the behavior of users after acquiring the data patterns that help develop the system to be aware of the situation and change the parameters accordingly. Therefore, as discussed in [10], the evolution of wireless technology along with embedded systems and micro-electrical mechanical systems demands the wireless sensor network (WSNs) and the Internet of Things (IoT) to incorporate artificial intelligence in such applications of home automation, industrial automation, target tracking, and security, etc.

1.3. Project Scope

The proposed scheme for the HAN environment is expected to outperform in perspective of system throughput, scalability, and efficiency. The scheme is appropriate so as to fulfill the requirements of different scenarios specifically targeting home area networks.

The HAN applications including several smart metering, smart appliances, smart power outlets, light control, remote control, smart energy, health care, and security applications are expected to deliver favorable results. Such applications of the HAN environment are intolerant to false alarms and need a high level of perfection for immediate actions and demand reliability and timely data delivery for which the proposed scheme is beneficial in every regard.

2. Medium Access Control (MAC) Protocol

The medium access control (MAC) protocol is a part of the datalink layer in the open system interconnections (OSI) model that is placed at the top of the physical layer, and the responsibility of the MAC protocol is to control and manage the access of the devices in the shared wireless medium as discussed in [1]. The data link layer (DLL) is usually enough for the scenario where there is a dedicated path between sender and receiver for data transfer and communication with each other [11]. The need for MAC protocol arises when there is no dedicated path between the sender and receiver to communicate, so multiple stations have access to the channel to share and transmit the data that cause collisions, and cross-talk may occur [12]. Thus, to avoid such collisions and for the successful transmission of data, multiple-access protocol is required. Figure 1 categorizes the multiple/medium access protocol into subcategories, and each category has its specific fields and applications.

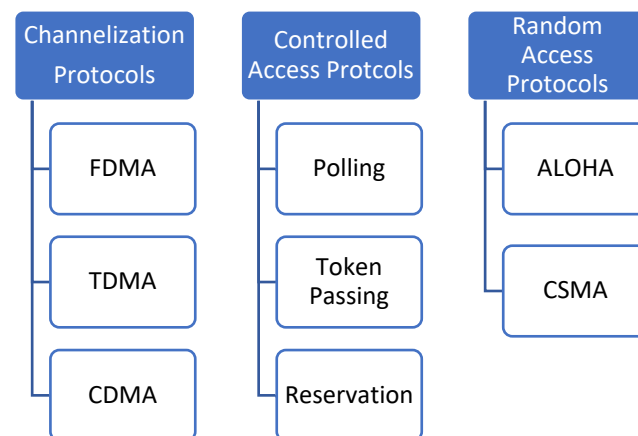


Figure 1. MAC protocol classification.

3. Carrier Sense Multiple Access (CSMA)

The most prominent protocol of medium access control is carrier sense multiple access (CSMA), which is widely used in home area networking applications, and this research paper has also adopted this method for optimizing the MAC protocol. The carrier sense multiple access protocol can sense the channel before starting communication over the channel. As per its mechanism [13], the station has to wait until it senses the channel is busy. So, it is surely the protocol that minimizes the chances of retransmission of data frames by avoiding collisions. CSMA protocol has the following common operation modes according to [13].

3.1. Persistent Mode

The channel is continuously sensed first, and if the channel is found idle, it transmits the data immediately otherwise it waits for the channel to be in idle condition to start transmission of data packets. The persistent mode works as shown in Figure 2.

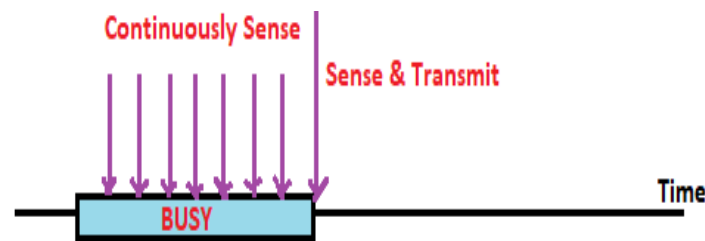


Figure 2. Persistent mode.

3.2. Non-Persistent Mode

It is different from the previous mode of operation in the sense that it does not sense the channel continuously. Whenever it becomes active, it senses the channel, and if it senses that the channel is idle, it transmits the data frames immediately, and then it waits for another random time of transmission. The non-persistent mode works as shown in Figure 3.

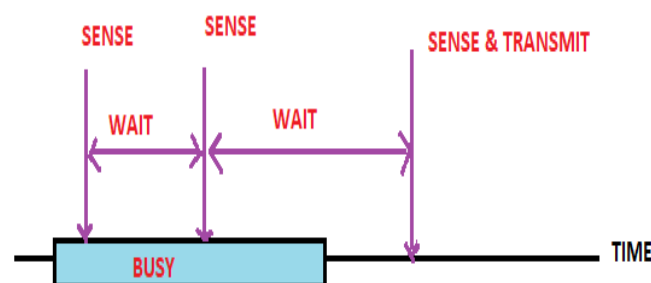


Figure 3. Non-persistent mode.

The proposed scheme in this research paper has also followed these two modes of the CSMA protocol. CSMA has two different working mechanisms discussed below.

3.2.1. CSMA/CD

CSMA/CD refers to carrier sense multiple access/collision detection. The channel is sensed first before starting transmission, and when the channel becomes idle, the transmission of data packets is started as discussed in [14]. On successful transmission, the station starts to transmit other packets, but if any collision is detected, then it terminates the process by sending a stop signal to the shared channel and waits for another random time to start the retransmission of data frames over the channel.

3.2.2. CSMA/CA

CSMA/CA is abbreviated as carrier sense multiple access/collision avoidance. In this variant of CSMA [15], collision is avoided in different ways.

- Inter-frame space (IFS): After sensing that the channel is inactive, stations are still not allowed to start transmission immediately and have to wait for some time further. This waiting period is referred to as inter-frame spacing.
- Contention window: It contains divided time slots, and whenever any station has to send data over a channel, it takes a random time slot to wait, and if the channel is sensed busy until then, instead of starting the whole procedure, it only restarts the timer for transmission of data frames after confirmation that the channel is idle.
- Acknowledgement: If acknowledgement is not received by the sender within the specified time, then the sender transmits data packets over the channel.

4. Importance of Machine Learning in Home Area Network

Home automation systems have been the center of attention for the last decade. To automate the HAN environment, activity recognition is a major application that is achieved

using the deep learning approach, and for assisting residents, the deep neural network (DNN) and recurrent neural network (RNN) algorithms are explored in [16] to achieve better results of deep learning. Moreover, without the concept of inter-related and inter-connected things, it is not possible to imagine smart applications. The Internet of Things (IoT) is the backbone of the smart world. The amalgamation of IoT with machine learning (ML) is a requirement of every smart application nowadays because without machine learning ML [17], IoT alone cannot make the system smart until there is no artificial intelligence that helps the systems to be independent and self-standing.

The machine learning technique is also used in predicting mobility patterns that are used to identify the future location of the user in a particular network [18]. The capability of machine learning (ML) is that it is effective in enhancing connectivity and reducing latency. Moreover, authors have discussed the artificial neural network (ANN) applications in [19]. ANN is good at enhancing behavior modeling, objectives classification, and speed of mobility prediction. The field of machine learning is expanding with new technologies [20], and the authors have designed a combination of k-means and compared it with SVM for recognizing the fingerprint of a smartphone, but it can be enhanced to any other IoT application of smart home as well. Connectivity among multiple IoT devices while maintaining QoS is necessary for smart HANs. Thus the author in [21], designed a bonsai algorithm that predicts IoT devices efficiently but locally without connecting to the cloud. In this process, machine learning is playing key role in predicting the system by allowing the devices to be connected every time, no matter whether the system is connected to the cloud or not.

With the advancement in IoT systems, better connectivity, and improved QoS, reliability is always demanded. IoT devices help in collecting time-series data where machine learning (ML) techniques are applied to improve the quality of service (QoS) of the network. Thus the combination of ML and IoT makes the system smart with enhanced connectivity, reliability, and QoS of the system.

This research paper has achieved the expected performance for a home area network through a random forest machine learning algorithm which will be discussed in detail in the next section to understand its working and functionality.

5. Random Forest Classifier

One of the most prominent and optimally suited machine learning algorithms for classification and regression problems is the random forest (RF) classifier, which comes under the category of supervised learning algorithms as discussed in [22]. It follows the rules of ensemble learning by combining multiple classifiers, hence solving the problem of over-fitting, which is necessary to optimize the model performance [23].

Ensemble Learning

The random forest classifier follows the rules of ensemble learning. Ensemble learning works by integrating numerous models to obtain predictions from each of them, and final judgment takes place based on voting discussed in [23,24]. It is divided into two categories:

1. BAGGING

Random forest works on the principles of bagging, discussed in [24]; e.g., training data is used to obtain multiple subsets, and the best result is chosen based on voting, as shown in Figure 4.

“Bootstrap Aggregation” is another name for the bagging technique of ensemble learning algorithm [24]. The multiple models generated from the training data set (bootstrap samples) with the process of substitution are referred to as “row sampling” and “bootstrap”. Each model is now independent to predict values whose results are “aggregated” at the end, and based on voting, the final output is generated. Therefore, it is also known as bootstrap aggregation.

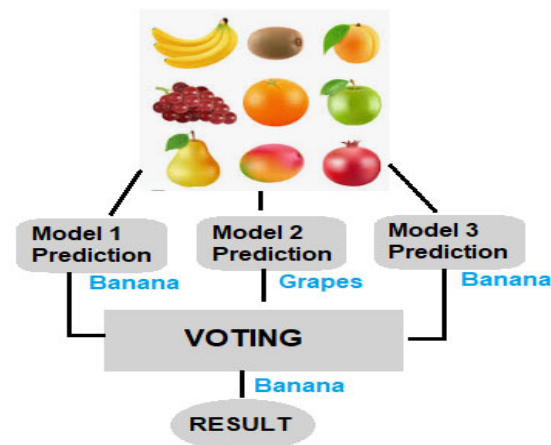


Figure 4. Method of bagging.

Figure 5 shows the visual analysis of bootstrap aggregation showing raw data and multiple decision trees (n number of samples and models are generated). Each model is trained independently from model 1 to model ' n ', as each model will generate independent results, and based on the majority of voting for classification and averaging for regression, the final output is selected, in this case, the shape of a "circle" from all other geometric shapes.

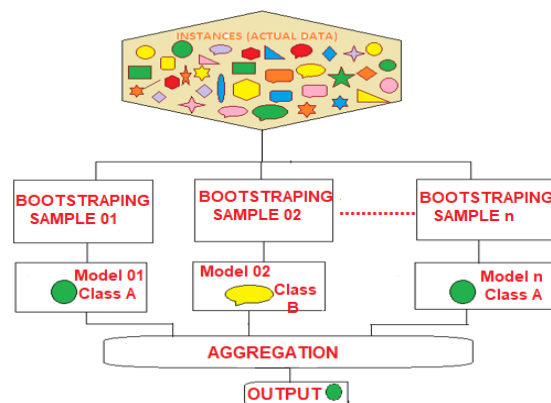


Figure 5. Working of ensemble classifier.

2. BOOSTING

Boosting is also an ensemble learning method discussed in [24] that works sequentially. It works by combining weak learners with stronger ones to obtain the most accurate final decision. Boosting takes the random sample training set and fits it with the classifier or model and all classifiers are trained sequentially based on the predictive outcome of the previous classifier. In this method, the voting mechanism is considered as a weighted vote of all previous weak learners; this method has to be trained to reach the final iteration.

The method of bagging is better at resolving the problem of over-fitting, and boosting is good at reducing biases among good and weak learners. Figure 6 shows the pictorial diagram of the boosting learning method.

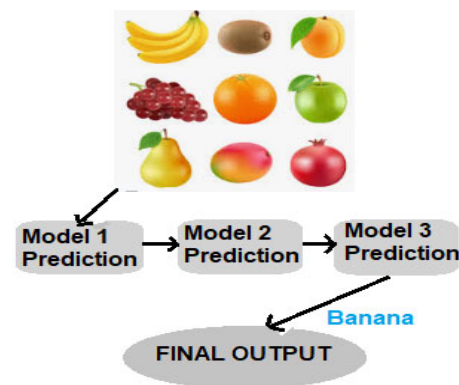


Figure 6. Method of boosting.

3. Procedure for Random Forest Algorithm

There are a few steps in this process:

- Taking 'n' number of samples from actual data;
- Creating decision tree models for every single sample;
- Obtaining predicted values from each model;
- Performing voting on predicted values;
- Considering majority voting (averaging) as the final decision (output).

4. Postulates for Random Forest Classifier

The random forest classifier makes decisions by combining and obtaining predictions from multiple decision trees, and judgment is based on averaging (voting) of classification and regression. Although it is accurate in decision making due to the presence of various decision trees [25], there remains the chance obtaining wrong predictions as well, so few assumptions are made before applying this classifier:

- The data set should have actual values as a feature variable to help predict accurate decisions; otherwise, there is a possibility of obtaining approximate results rather than the actual and accurate ones.
- The predictions of decision trees should not be highly correlated to each other.

6. Proposed Scheme

Our research aim is to provide a more straightforward optimization of MAC CSMA/CD protocol for wireless networks. Local area networks (LANs) that do not rely on traditional cable or wired systems are known as WLANs (Wireless LANs). This project is created as part of a separate study and uses the Python Jupyter notebook to simulate an 802.11 wireless network. The goal of this article is to show readers how we solved challenges and built the 802.11 MAC protocol.

6.1. Traditional CSSMA/CD

The proposed work focuses on CSMA/CD and its optimization with a random forest classifier. However, the primary premise of CSMA/CD is that a station must be able to receive while transmission to identify a collision between two stations. Since the CSMA/CD protocol does not contain an acknowledgment signal, it only has collision signals to interpret whether the channel is busy or not. So, the algorithm is designed in such a way that it can detect collisions in a timely fashion while transmitting the data packets with minimum delay.

The flow of the project starts with simulating a CSMA/CD protocol, taking reference from the GitHub Python implementation [26] of the CSMA/CD protocol. Because the CSMA/CD Python built-in library is not available, its simulator can be downloaded from GitHub. However, the modification is performed over the CSMA/CD simulator, and the

algorithm steps are discussed in Section 8 for obtaining the results of interest and collecting the data samples.

6.2. Data Set Generation

There are a number of sequential steps, mentioned in Figure 7, that need to be followed to generate a data set using the Python Jupyter IDE platform.

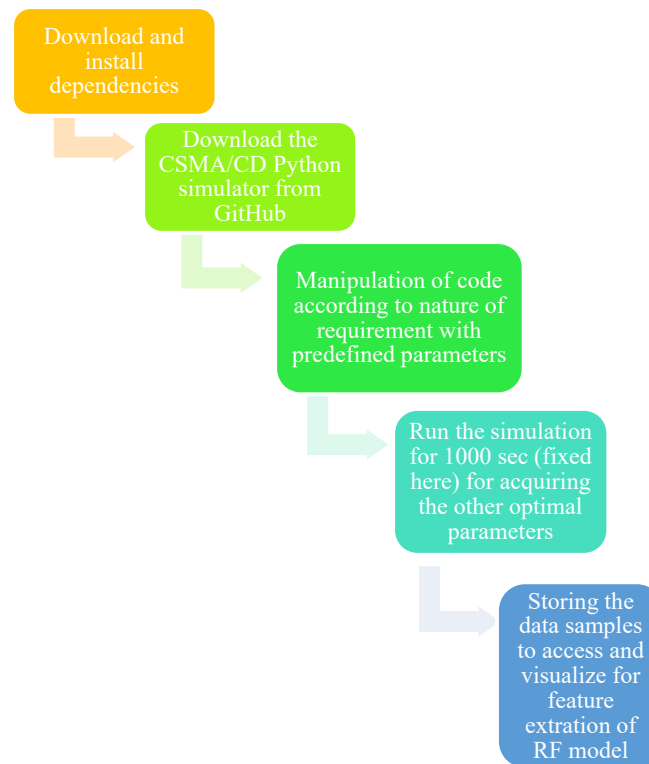


Figure 7. Data generation steps.

1. The following parameters are considered predefined:
 - 'N' is the number of nodes present in the wireless local area network;
 - 'A' is described as the average packet arrival rate;
 - 'R' is denoted as the speed of the WLAN in bits per second;
 - 'L' is determined as packet length in bits;
 - 'D' is termed as the distance between adjacent nodes on the bus/channel;
 - 'S' is termed as propagation speed in m/s.
2. The parameters achieved from the algorithm include the following:
 - 'Current time', or packet current lifetime, is the time taken by a data packet from generation to transmission, based on virtual simulation time;
 - 'Persistent state' (if is-persistent is '1' or 'True');
 - 'Non-persistent state' (if is-persistent is '0' or 'False');
 - 'Node queue' (the present waiting nodes) and is calculated as
 - ◆ $\text{node.queue}[i] = (\text{curr_time} + t_{\text{prop}} + t_{\text{trans}})$
 - It becomes 'Previous node queue' if $(\text{curr_time} + t_{\text{prop}}) < \text{node.queue}[i] < (\text{curr_time} + t_{\text{prop}} + t_{\text{trans}})$:
 - ◆ $\text{previousnodequeue} = \text{node.queue}$
 - Propagation time (ns) is the time taken to propagate;
 - Node location;
 - State of collision.

These parameters are helpful in feature extraction, shown in the following Tables 1–4, for the implementation of the random forest classifier.

Table 1. Summary of independent features.

Independent Features	Values
Number of Nodes (N)	In range (20,101,20)
Average Packet Arrival Rate (A)	In [7,10,19]
Speed of WLAN (R)	$1^{(10,6)}$ (10, 6)
Packet Length (L)	1500 in bits
Distance b/w adjacent nodes (D)	10 (bus/channel)
Propagation Speed (S)	$S = (2/\text{float}(3)) \times C$ In m/s

Table 2. Summary of dependent features.

Dependent Features	Values
is-persistent	either true (1) or false (0)
Node.queue	in seconds
Current time	virtual simulation time
Propagation Time	in nano-seconds
Will-collide	either true or false
Collision occurred once	either true or false
Node location	multiple of D
Node Collisions (before and after waiting)	Less than 10

Table 3. Will-collide feature extraction set.

# of Samples	N	A	R	L	D	S	Is_Persistent	Node.Queue [0]	Curr_Time (s)	t_Prop (ns)	Will_Collide
0	20	7	1,000,000	1500	10	200,000,000	1	0.057714	0.008647	5×10^7	False
1	20	7	1,000,000	1500	10	200,000,000	1	0.037964	0.008647	4.5×10^7	False
2	20	7	1,000,000	1500	10	200,000,000	1	0.193615	0.008647	4×10^7	True
3	20	7	1,000,000	1500	10	200,000,000	1	0.123876	0.008647	3.5×10^7	False
4	20	7	1,000,000	1500	10	200,000,000	1	0.297117	0.008647	3×10^7	True
...
1,048,572	60	7	1,000,000	1500	10	200,000,000	1	5.543805	4.968325	8.5×10^7	False
1,048,573	60	7	1,000,000	1500	10	200,000,000	1	4.968326	4.968325	8×10^7	False
1,048,574	60	7	1,000,000	1500	10	200,000,000	1	5.183740	4.968325	7.5×10^7	False

Table 4. Collision cases feature extraction set.

# of Samples	N	A	R	L	D	S	Is_Persistent	Collision_Occurred_Once	Node_Location	Node_Collisions	Node.Wait_Collision	Node.Max Collisions
0	20	7	1,000,000	1500	10	200,000,000	True	False	190	0	0	10
1	20	7	1,000,000	1500	10	200,000,000	True	False	190	0	0	10
2	20	7	1,000,000	1500	10	200,000,000	True	False	190	0	0	10
3	20	7	1,000,000	1500	10	200,000,000	True	False	190	0	0	10
...
255,342	100	20	1,000,000	1500	10	200,000,000	False	False	990	0	0	10
255,343	100	20	1,000,000	1500	10	200,000,000	False	False	990	0	0	10

3. The generated data sets are briefly discussed below:

Data-Set 1 (will-collide list) is a predictive analysis of whether the nodes will collide or not depending on features, e.g., persistent state, current time, propagation time, and node queue. It contains 1,048,575 samples with 11 features.

Data-Set 2 (collision-case data set) includes detailed data samples of collision cases, i.e., whether the collision once occurred, on what node location, number of node collisions, or collisions after waiting (node wait collisions). It contains 255,344 samples with 12 parameters.

7. Feature Extraction

There are a number of features extracted (dependent and independent) are summarized in Tables 1 and 2.

Tables 3 and 4 represent the feature extraction set (data sets) discussed in the previous section.

8. Classification

Classification is defined as a process of categorizing a given data set into categories called classes. Classification itself is supervised machine learning, knowing the targets with the input data set. Our targets of research are to optimize the CSMA/CD by detecting collisions, and try to reduce the number of collisions so to transmit more packets successfully. The proposed random forest classifier has classified the data inputs into two classes. It means that binary classification with two outcomes is preferred here. Class '0' represents 'no collision occurred', and Class '1' represents 'the collision occurred'. These classes are named as Class 0, which is "successful data transmission state", and Class 1, which is "unsuccessful data transmission state". In later sections, it might be observed that the number of samples of class '0' will be more than the number of samples of class '1'. The reason behind the encouragement of class '0' is to enhance the network performance with successful transmission of data frames. There will always be a trade-off between these factors; more successfully transmitted packets means lower collision occurrence, indicating less appearance of Class 1, while less successfully transmitted packets means that a greater number of collisions occurred, indicating a greater number of samples of Class 1 appearance.

9. Feature Selection and Data Transformation

After the successful feature extraction process, there is the turn to analyze the extracted features and select the relevant features according to their significance in the linear regression model. To select the relevant features, a correlation matrix is being computed, and highly correlated features are being selected for random forest modeling.

The process of feature transformation is a group of techniques that create new features (predictor features) from the existing feature set. Feature selection is a subset of feature transformation; it is performed for knowledge discovery, interpretability, and data analysis and to diminish the curse of dimensionality. There are two ways of performing feature selection in supervised machine learning [27]. Filter-type techniques select features based on statistical analysis instead of using an ML-based algorithm. They are based only on features' correlation with the target using ANOVA, LDA, Chi-Square, Pearson's coefficient, etc. Filter techniques suppress the least interesting features. They are mainly applied as a pre-process method. In contrast, the Wrapper technique evaluates multiple subsets of features using a model from which the best-performing feature combination is chosen. Unlike filter approaches, this allows the detection of the possible interactions between features.

Out of the above explanatory features of the generated data-sets, six numerical inputs and five categorical outputs are selected from Data-Set 1 (will-collide data set), and four categorical outputs are selected from Data-Set 2 (collision-case data set) to simplify the analytical process. The numerical inputs are the same/constant in both data sets, with only a difference in categorical outputs. The set of selected features for RF model is mentioned in Table 5.

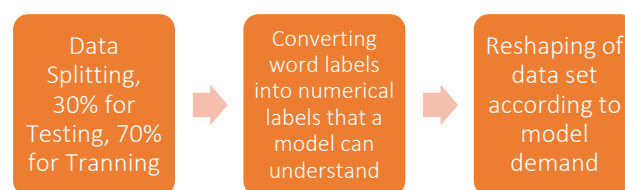
The selection of these features was a priority, except for the numerical data ID, years, and months, which were considered irrelevant for the analysis. The selected columns remain, and the undesired columns are deleted.

At the start all features were analyzed, but relevant selected features have a great impact on simplifying the algorithm, reducing the time and complexity, and giving accurate outputs with minimized error.

Table 5. Selected features.

Numerical Inputs	
Number of Nodes (N)	
Average Packet Arrival Rate (A)	
Speed of WLAN (R)	
Packet Length (L)	
Distance b/w adjacent nodes (D)	
Propagation Speed (S)	
Categorical Output (Data-Set 1)	
Is-persistent	
Current time	
Propagation time	
Node.queue	
Will-collide	
Categorical Output (Data-Set 2)	
Is-persistent	
Collision Occurred	
Node Location	
Node Collision	

Figure 8 mentions the steps applied to finalize the data set for training models.

**Figure 8.** The procedure of preparing feature set to train models.

10. Linear Regression Modeling

The random forest classifier is used for classification and regression of the defined features and for achieving better predictive analysis. But the non-linear nature of the random forest classifier can give it a leg up over the linear algorithm with which a linear regression model is also taken to represent the relationship between dependent and independent variables. In this research paper, both linear regression model and random forest classifier are used to achieve optimized outputs. There are a few assumptions before considering the linear regression model.

1. Linearity: The relationship between the dependent and independent variables should be linear.
2. Independence: Observations should be independent of each other.
3. Homoscedasticity: Variance should be the same as residual (error term) at any value of the response variable 'X'.
4. Normality: The predictor's 'Y' value is normally distributed for the values of the dependent (response) variable of 'X'.

The following Equations (1)–(10) are applied to estimate the linear regression model.

Estimating Linear Regression Models Using Least Squares

The system of n equations can be represented in matrix notation using Equation (1):

$$y = X\beta + \varepsilon \quad (1)$$

$$\text{where } y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

Hence, the equations will be formed using these matrices, as shown in Equations (2)

$$\text{and (3): } \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

$$y_1 = \beta_0 + x_1 \beta_1 + \varepsilon_1 \quad (2)$$

$$y_n = \beta_0 + x_n \beta_1 + \varepsilon_n \quad (3)$$

The matrix X is referred to as the design matrix. It contains information about the levels of the predictor features at which the observations are obtained. The vector β contains all the regression coefficients. To obtain the regression model, β should be known. β is estimated using least square estimates. The following Equation (4) is applied:

$$\beta = (X'X)^{-1}X'y \quad (4)$$

where $'$ represents the transpose of the matrix, and $'^{-1}$ represents the matrix inverse. Knowing the estimates, β , the linear regression model can now be estimated according to Equation (5):

$$\hat{y} = X\beta \quad (5)$$

The estimated regression model is also referred to as the fitted model. The observations, y_i , may be different from the fitted values \hat{y}_i obtained from this model. The difference between these two values is the residual, e . The vector of residuals, e , is obtained using Equation (6):

$$e = y - \hat{y} \quad (6)$$

The fitted model can also be written as follows, using Equations (7) and (8):

$$\hat{y} = Hy \quad (7)$$

where

$$H = (X'X)^{-1}X' \quad (8)$$

The matrix, H , is referred to as the hat matrix. It transforms the vector of the observed response values, y , to the vector of fitted values, \hat{y} .

The least square estimates, $\beta_0, \beta_1, \dots, \beta_p$, are unbiased estimators of the value of $\beta_0, \beta_1, \dots, \beta_p$, provided that the random error terms, ε_i , are normally and independently distributed. The variances of the $\hat{\beta}_s$ are obtained using the $(XX)^{-1}$ matrix. The variance-covariance matrix of the estimated regression coefficients is obtained using Equation (9):

$$C = \hat{\sigma}^2(XX)^{-1} \quad (9)$$

C is a symmetric matrix whose diagonal elements C_{ij} represent the variance of the estimated j_{th} regression coefficient, $\hat{\beta}_j$. The off-diagonal elements, C_{ij} , represent the covariance between the i_{th} and j_{th} estimated regression coefficients, $\hat{\beta}_i$ and $\hat{\beta}_j$. The value of $\hat{\sigma}^2$ is obtained

using the error mean square, MS_E . The positive square root of C_{jj} represents the estimated standard deviation of the j th regression coefficient, $\hat{\beta}_j$, and it is known as the estimated standard error of $\hat{\beta}_j$ (abbreviated as $se(\hat{\beta}_j)$) and calculated as given in Equation (10).

$$se(\hat{\beta}_j) = \sqrt{c_{jj}} \quad (10)$$

11. Process of Implementation of Proposed Algorithm

1. Initialization of parameters:

To run the algorithm, 'N' is first set as 20, which is the number of nodes/computers connected to the WLAN, 'A' is set to be 7, which is the average packet arrival rate (packets per second), 'R' = $1^{(10,6)}$ (10, 6), which is the speed of the LAN/channel/bus (in bps), and 'L' is set as 1500, which is the packet length (in bits).

Run the algorithm by using the above functions in a loop according to the number of nodes and number of packets and evaluate the occurrence of collision and S propagation speed (meters/sec). And then, all these above parameters will be given as the input to the proposed algorithm.

2. In case of collision occurrence:

Firstly, for observing the collision occurrence, it is required to define maximum collision to nodes, and the value of maximum collision is '10' defined here. So, in case collisions are greater than the maximum collision allowed during transmission, then the packet is dropped. Then, the dropped packet is de-queued (trying to send it again). With defining location as distance, numbers of packets are updated to be resent, and the collision counter is reset to '0', waiting for collision occurrence. Exponential back-off time is added to this collision waiting time. Figure 9 shows the steps to be taken in case of collision:

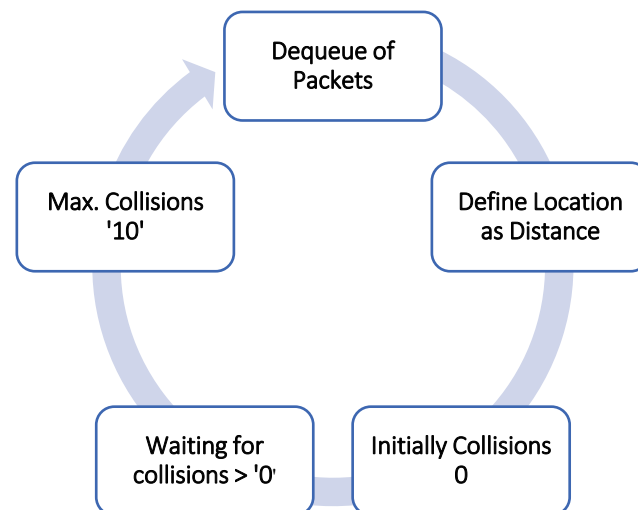


Figure 9. Steps in case of collision occurrence.

3. In case of successful transmission:

In the best-case scenario (i.e., the transmission of packets between nodes is successful), the number of collisions is set to zero, and the number of packets waiting during that transmission is also set to zero as all the packets are successfully transferred.

4. Get random value between generating a queue of packets:

It is necessary to generate the queue of arrived packets and set the arrival time as the exponential random variable according to the average packet arrival rate (packets per second), which is denoted as A. The work also requires exponential random variables

during our transmission between nodes; for that, we need to obtain uniform random values between 0 (exclusive) and 1 (inclusive).

5. In case of a non-persistent bus:

In the case of a non-persistent bus, the exponential back-off time will be according to the speed of the wireless WLAN/channel/bus (in kbps) with the 512 kbits times. It is essential to first check if a collision has occurred in the case of which the packet should be dropped and the exponential back-off time is added to the waiting period.

6. Random forest algorithm in CSMA/CD:

With the pre-defined parameters, e.g., (N, A, R, L, D, S), the protocol has achieved all samples, trained through the random forest algorithm. It includes a collision-occurred list, successful packet transmitted list, transmitted packet list, current-time list, propagation-time list, transmission-time list, node location, node queue, will-collide list, and collision-case list, from which the algorithm can sense the link idleness to send the data frames for successful transmission.

The working mechanism of the random forest (RF) model-based CSMA/CD is detailed in Figure 10.

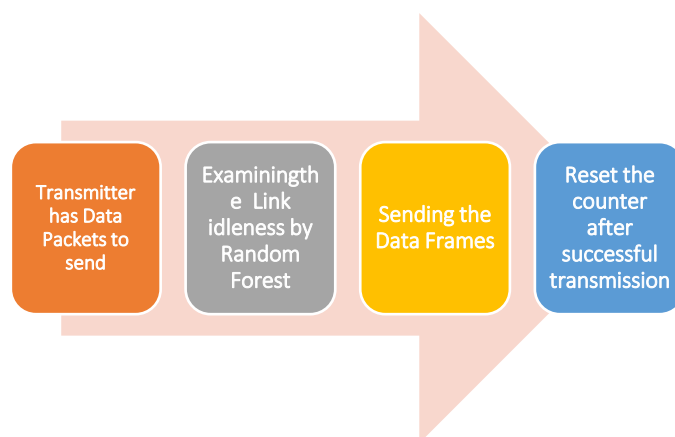


Figure 10. CSMA/CD with ML classifier.

Overlapping data and collisions can be avoided, as mentioned below in Figure 11.

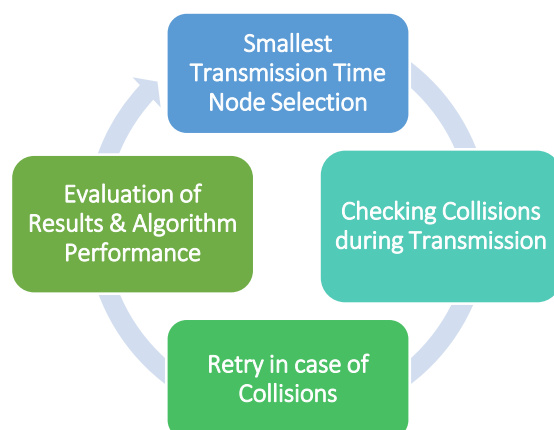


Figure 11. Procedure of collision avoidance.

These steps are discussed below:

1. Pick the node with the smallest transmission time out of all the nodes:

The purpose of selecting the smallest transmission time node is that the CSMA/CD prefers the node with the shortest frame size in a large network. This is because with a

larger number of nodes, the chances of a collision increase, and if any collision is detected on the other side of the network, the jam signal should have enough time to detect the collision, and the transmitter can obtain knowledge of the collided packet in a timely fashion to start the process of retransmission. So, node parameters are defined here that are essential in the packet's sending and receiving process to evaluate the occurrence of collision and to check and verify that the transmitter is ready to transfer data packets. Figure 12 shows the sequential steps of this process;

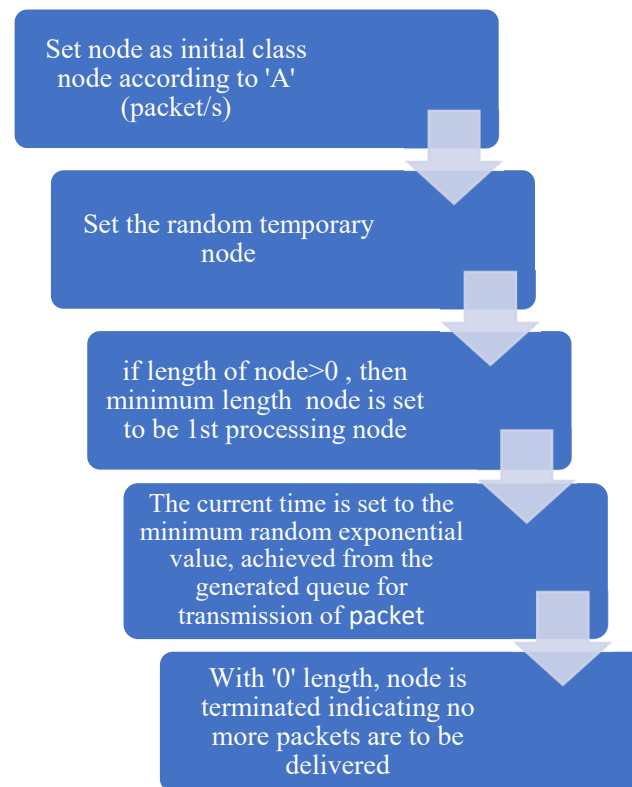


Figure 12. Steps to transmit packets of smallest time nodes.

2. Check if a collision will happen:

During these loops of packet transfer, it is important to evaluate whether a collision happened during transmission or not and to check if all other nodes except the minimum distance node will collide or not; for that, a random forest trained model is applied to evaluate the collision according to the given scenario and parameters of the transmission step, and it will be checked if the WLAN is busy on every node or not. The transmission velocity (VTP) is computed in Equation (11) by dividing the change in node locations by the node propagation speed (meters/sec).

$$V_{TP} = \frac{\text{prev Location} - \text{Current Node Location}}{\text{Propagation Speed (meters/sec)}} \quad (11)$$

The packet transmission is computed as given in Equation (12); T_{pT} = Transmission Packet Transfer Time.

$$T_{pT} = \frac{\text{Transmitted Packet Length in bits}}{\text{Speed of } \frac{\text{WLAN}}{\text{channel}} / \text{bus in bps}} \quad (12)$$

Then, it is analyzed whether it will collide or not via a random forest model with parameters of node packet queue initial time, current time, and transmission propagation velocity. If the WLAN is busy, the persistent state will be true, and the node packet time queue will be saved as the previous queue time. And the queue will be updated on that

node with the sum of the current time, transmission propagation velocity, and transmission packet transfer time, and if the packet transmission between nodes collides, then the collision case will be true, and the transmission of the packet will be finished, and the node collision will be set as true; 'b', the speed of the WLAN/channel/bus (in bps).

3. If a collision occurs, then retry:

If a collision occurs, then retry to send the packet; otherwise, update all nodes and the latest packet arrival times, and proceed to the next packet if no collision happened; the packet will be successfully transmitted.

4. Run algorithm:

Run the algorithm to obtain the final output.

12. Modeling Results

12.1. Random Forest Classifier Results

The random forest classifier is used for classification and regression of the defined features and for achieving better predictive analysis. The data sets are trained independently. One thing that should be noted about the random forest classifier is that carefully obtaining random 'm' variables out of 'M' input variables is essential. The correlation and strength of the classifier highly depend on this input random variable.

The final model has a significant overall p -value of less than 0.05. The model's results suggest that all independent features are partially important and have a strong relationship with the busy state of WLAN and collisions in the network. Tables 6 and 7 exhibit the statistical scores of the random forest (RF) and linear regression algorithms for a performance comparison between the two approaches.

Table 6. Random forest performance results.

Criterion	RF
Score	0.99
MAE	0.1
RMSE	0.3
MSE	0.1
Mu	0.34
Sigma	0.1
Pearson correlation coefficient	0.99999999
	9,998,516
p value	0.1

Table 7. Random forest classification performance report results.

Label	Precision	Recall	F1 Score	Support
Collision-Case	0.956	0.964	0.964	25,970
Will-Collide	0.970	0.967	0.968	243,908
Macro Average mean	0.963	0.965	0.966	262,144
Weighted Average mean	0.968	0.966	0.978	262,144

The model includes enough statistically significant features, indicating that it is very predictive of the response (outcome).

The coefficient of determination (R-squared value) is a measure of how well the independent features explain the variance in the response feature. As previously stated, the characteristics (independent features) in this final model account for 82 percent of the variation in the class and forecast values.

As a result, the mean absolute error (MAE), root mean square error (RMSE), and Pearson correlation coefficient R can be evaluated using these two array values, actual and

anticipated. The following mathematical formula defines these comparison measures or criteria in Equation (13):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (13)$$

where y_i is the actual value, \hat{y}_i is the predicted value, and n is the number of observations.

Root mean square error (RMSE) is calculated using Equation (14).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|^2} \quad (14)$$

Pearson correlation coefficient (R) is calculated in Equation (15).

$$R = \sqrt{\frac{\sum_{i=1}^n (y_i - \bar{y})^2 \sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}{\sum_{i=1}^n (y_i - \bar{y})^2 \sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}} \quad (15)$$

where y_i is the actual value, \hat{y}_i is the predicted value, \bar{y} is the mean of the actual value, $\bar{\hat{y}}$ is the mean of the predicted value, and n is the number of observations.

The random forest performance results are shown in Table 6, while Table 7 shows the classification performance of the random forest classifier. Following are the outputs of this segment after applying a random forest classifier over data sets.

12.2. Confusion Matrix

The confusion matrix is described as a synopsis of the classifier's predictions (whether it is correct or not). It represents four major scores of Class 0 and Class 1 as true positive (TP), true negative (TN), false positive (FP), and false negative (FN), and these are whole numbers. Figures 13 and 14 shows the confusion matrix of selected data sets. The results show that it is a binary classification; the occurrence of class 0 is more than the class 1 due to the network requirement of having a smaller number of collisions and due to the performance of classifiers being excellent, as false predictions are fewer as compared to true predictions.

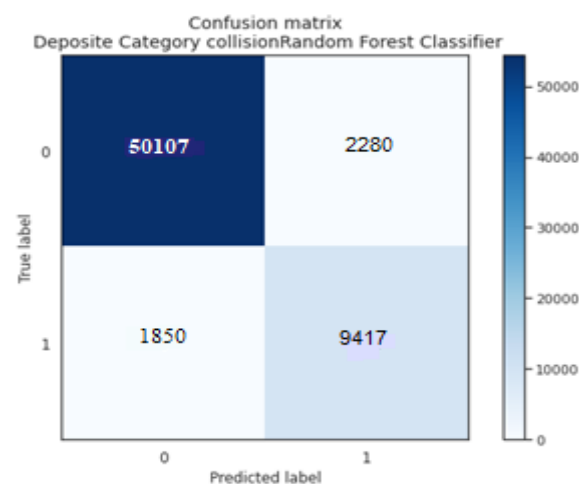


Figure 13. Collision-case data set confusion matrix.

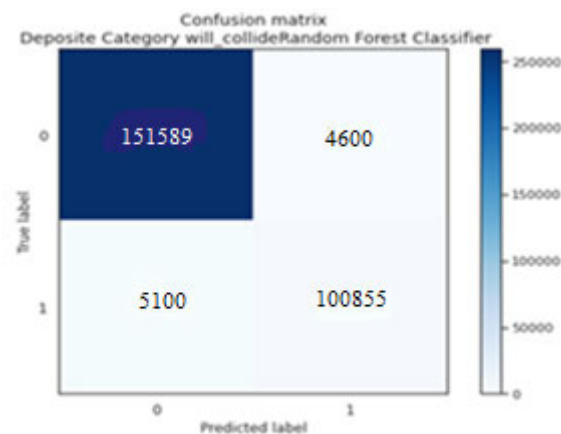


Figure 14. Will-collide data set confusion matrix.

12.3. Precision–Recall Curve

The precision–recall (PR) curve and receiver operating characteristics (ROC) curve are used to visualize the trade-off in performance for different threshold values in binary (two-class) classification.

It is simply an x-y curve representing precision on the y-axis and recall on the x-axis. Recall is basically the sensitivity of the classifier. The performance of the classifier can be analyzed from the area under the curve (AUC). The range of AUC lies between 0 and 1. But the classifiers having good performance show $AUC > 0.5$; because 0.5 AUC is the performance of a random/no-skill classifier, and if AUC is < 0.5 , then the classifier performance is considered worse than the no-skill (unable to distinguish among the classes) classifier. However, the AUC of the perfect classifier is '1', which highly considers the correct positive values in contrast to incorrect negative values. Figures 15 and 16 show the precision–recall curve of the collision-case and will-collide classifiers. The blue dotted line represents the no-skill classifier, while the red line represents the PR curve of the logistic. Results show that there is an AUC equal to 0.974 for the collision case classifier, and an AUC of 0.989 for the will-collide classifier representing the very good and skillful classifiers. The area under the curve (AUC) of the will-collide classifier and collision-case classifier is almost the same due to using the same RF model, but the AUC of the will-collide classifier is somehow better due to the larger and balanced data samples than the collision case classifier. High AUC represents high recall and high precision. However, high precision represents a low false positive rate, and high recall represents a low false negative rate.

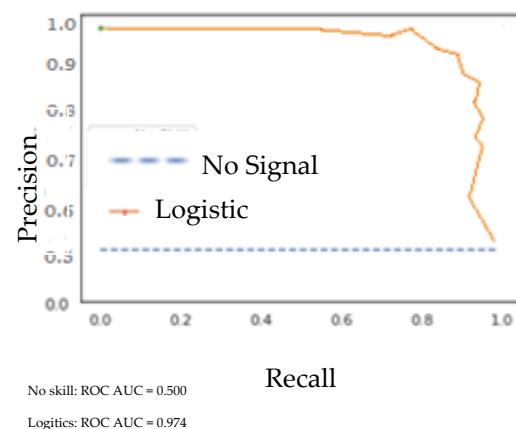


Figure 15. Precision of collision-case classifier.

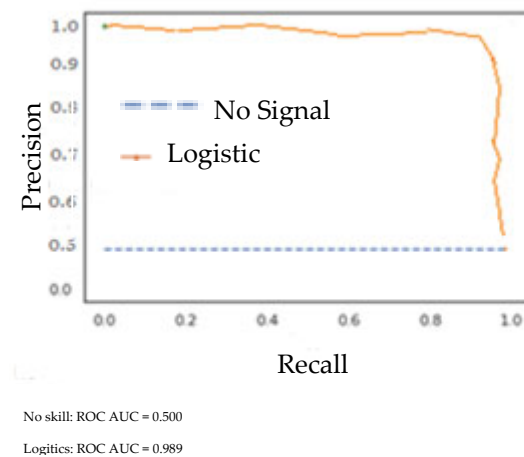


Figure 16. Precision of will-collide classifier.

Precision and recall can be calculated using Equations (16) and (17).

$$\text{Precision} = TP / (TP + FP) \quad (16)$$

$$\text{Recall} = TP / (TP + FN) \quad (17)$$

Other important parameters used in evaluating the performance of the random forest classifier are highlighted below:

1. **F1 Score:** Precision and recall are two important factors used in evaluating model performance. F1 Score is another useful metric that considers both factors and is calculated using Equation (18).

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

2. **Macro Average:** It is the arithmetic mean (un-weighted mean/usual average) of all the individual per-class F1 Scores (add all classes up and divide it by the number of classes).
3. **Weighted Average:** It is calculated by taking the mean of all per-class F1 Scores but with consideration of the class's support.
4. **Support:** It is termed as the number of occurrences of the particular class in the data set.
5. **Micro Average:** It is computed after the overall summation of TP, FP, and FN values of all classes and then substituted in Equation (19) of the F1 Score to obtain a Micro Average.

$$\text{F1 Score} = 2 \times \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (19)$$

6. **Accuracy:** It somehow has the same result as the Micro Average because the Micro Average computes the proportion of correctly classified observations out of total observations, which is the same as the purpose of computing the accuracy of the classifier.

It can be seen how well the random forest performed in the detection of collision by using the given parameters and conditions in Table 6. Table 7 computes the classification performance parameters of the random forest classifier. Random forest has resulted in an accuracy of 99%.

12.4. Receiver Operating Characteristics Curve (ROC)

A graph is used to illustrate the performance of the model at different classification thresholds, having two parameters of the true and false positive rate. After analyzing the

true positive rate and false positive rate on different thresholds, a curve is plotted that is stretched between the bottom left and top right, with the bowing towards the top left being called the ROC curve. ROC curves are useful where observations are balanced between binary classes; otherwise, for imbalanced observations, precision–recall curve is preferred. Figures 17 and 18 show the ROC curve of the collision-case and will-collide classifiers. The blue dotted diagonal line represents a classifier that is unable to distinguish between the true positive rate and false positive rate, and the green curve represents the classifiers that have very good skills, which indicates that there are more correct predictions of positive classes than predictions of incorrect negative classes in the given scenarios.

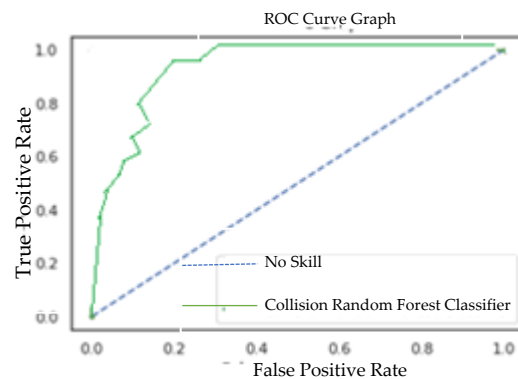


Figure 17. ROC curve graph of collision-case classifier.

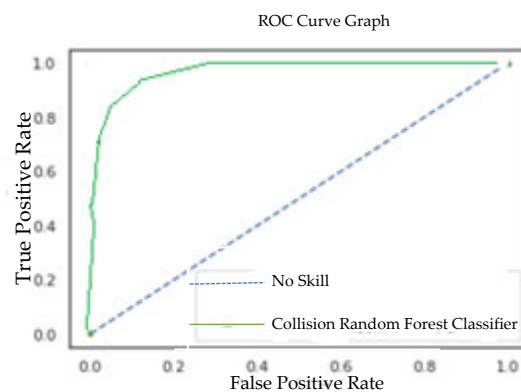


Figure 18. ROC curve graph of will-collide classifier.

12.5. Linear Regression Model Output

It can be observed from Table 8 how well the linear regression performed in predicting the busy state of the WLAN using the given parameters and conditions, and the linear regression model has scored 0.97. The following Figure 18 exhibits the linear regression model performance.

Table 8. Linear regression performance results.

Criterion	Linear Regression
Score	0.97
MAE	$1.35467510866254 \times 10^{-13}$
RMS	$5.214331110962009 \times 10^{-12}$
MSE	$0.7189248934746295 \times 10^{-23}$
Pearson correlation coefficient	1.0
p value	0.0
Valid Mu	5.06
Valid Sigma	2.90
Predicted Mu	5.06
Predicted Sigma	2.90

Figure 19 shows a valid vs. predicted probability plot of a linear regression model. The red line depicts the theoretical regression model. The analysis is necessary to validate the regression model whether it satisfies the assumptions of normality, linearity, and independence or not. As the linear regression predicts a continuous valued output, the output value can be any possible integer number. So, the red line is the regression line. The blue curve represents the outputs achieved using the linear regression model. The regression line is considered as the best fit of a set of data where few data outcomes will touch the regression line, and others cannot touch it. The linear regression probability plot identifies the regression problems of not following the given assumptions, and the data point must be at varying distances or outliers or have a non-linear nature.

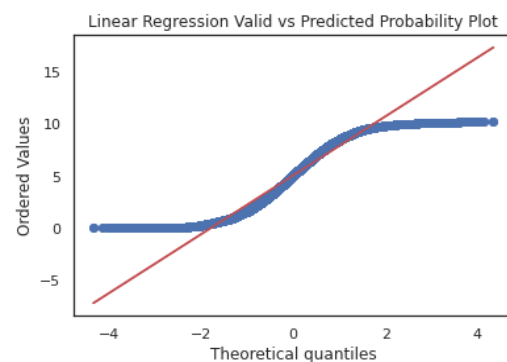


Figure 19. Linear regression probability plot.

12.6. Comparison of Traditional vs. Predicted CSMA/CD Outputs

For a comparison of the random forest-based CSMA/CD algorithm with traditional CSMA/CD algorithm, the efficiency and throughput Equations are (20) and (21).

$$\text{Efficiency} = \frac{\text{Successfully Transmitted Packets}}{\text{Transmitted Packets}} \quad (20)$$

$$\text{Throughput} = \frac{L \times \text{Successful Transmitted Packets}}{\text{float}(\text{curr} - \text{time} + (L/R))^{10,-6}} \quad (21)$$

Following are the outputs before and after implementing the random forest classifier as a machine learning algorithm; the results are consecutive outputs of a local area network (LAN) in Mbps.

Figure 20 shows the average packet delay of persistent nodes (persistent true) in which the blue line represents traditional CSMA packet delay, and the orange line represents the predicted average packet delay after implementing the RF model.

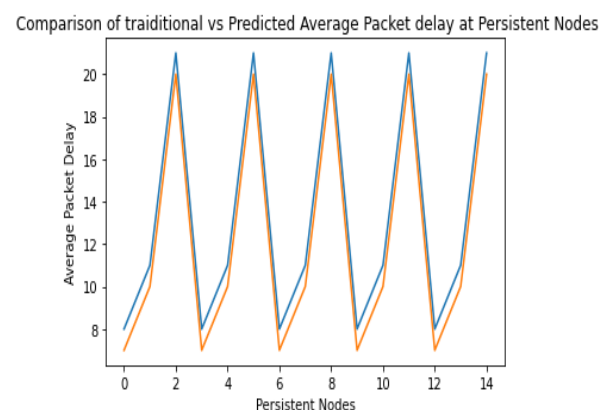


Figure 20. Average packet delay of persistent true.

Figure 21 shows average packet delay of non-persistent nodes (persistent false) in which the blue line represents traditional CSMA packet delay, and the orange line represents the predicted average packet delay after implementing the RF model.

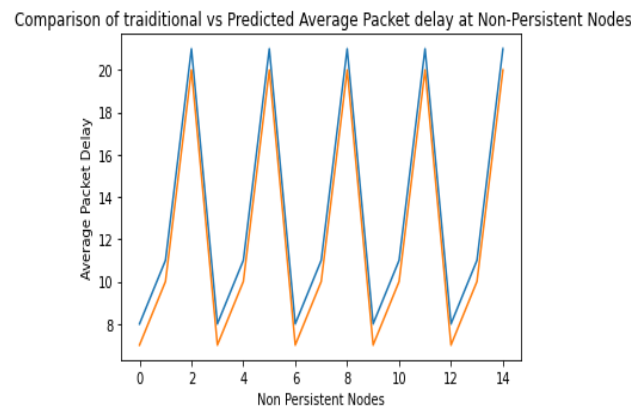


Figure 21. Average packet delay of persistent false.

Figure 22 shows the efficiency of persistent nodes (persistent true) in which the blue curve represents the traditional CSMA efficiency, and the orange curve represents the predicted efficiency after implementing the RF model.

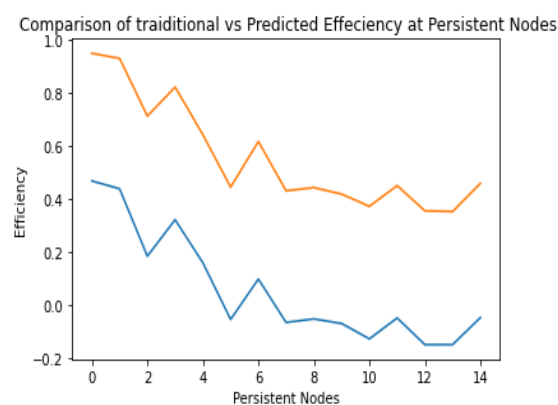


Figure 22. Efficiency of persistent true.

Figure 23 shows the efficiency of non-persistent nodes (persistent false) in which the blue curve represents the traditional CSMA efficiency, and the orange curve represents the predicted efficiency after implementing the ML model.



Figure 23. Efficiency of persistent false.

Figure 24 shows the throughput of persistent nodes (persistent true) in which the blue curve represents the traditional CSMA throughput, and the orange curve represents predicted throughput after implementing the RF model.

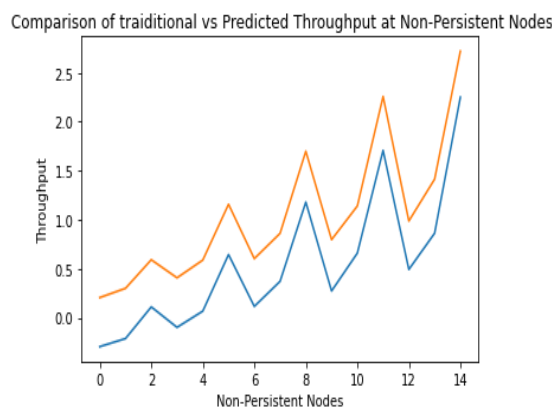


Figure 24. Throughput of persistent false.

All the outputs shown represent the fact that the overall performance of the network is enhanced by implementing the RF model on the traditional CSMA/CD MAC protocol and that it is producing thoroughly better results.

13. Conclusions

The demand for artificial intelligence in IoT-based systems arouses keen interest in studying and exploring machine learning techniques and their specific applications. Undoubtedly, the IoT has unique capabilities of collecting and gathering data, and sensing and identifying the situation. However, the incorporation of ML adds intelligence to IoT devices, which helps them to become smart and become capable of making decisions on their own. The amalgamation of ML and IoT devices has enhanced the quality of service (QoS) and connectivity of devices and has improved the human level of comfort. IoT applications are vast, and to be application specific, this research work targets smart home area networks. Therefore, this work comprises a detailed survey regarding the home area network environment and the urge to deploy artificial intelligence for HAN optimization. However, according to this study, it is analyzed for any IoT application, and it is necessary to handle the insufficient resources and hard challenges of WSN for which optimizing the MAC protocol is an important factor. Moreover, without optimizing the MAC protocol, it is not possible to reap the benefits of WSN nodes of IoT that are spatially co-located and sharing the same spectrum.

This research work has surveyed the relevant CSMA/CD MAC protocols and their variants and random forest machine learning techniques, and the study concluded that different approaches are practiced to achieve an autonomous HAN environment but lack efficiency and optimal throughput required for the network. Due to the outdated techniques using CSMA in the world of smartness, where every application demands artificial intelligence and automation, and to meet the QoS of smart home area network, the optimization of the MAC protocol through a machine learning algorithm is a valuable solution. This research work has provided an optimized MAC protocol using the random forest (RF) classifier as the machine learning algorithm and highlights the role of artificial intelligence in automating home area networks.

Author Contributions: Conceptualization, B.M.K.; methodology, B.M.K.; software, B.M.K.; validation, B.M.K. and M.B.K.; formal analysis, B.M.K.; investigation, B.M.K. and M.B.K.; resources, M.B.K.; data curation, B.M.K.; writing—original draft preparation, B.M.K.; writing—review and editing, M.B.K.; visualization, M.B.K.; supervision, B.M.K.; project administration, B.M.K.; funding acquisition, M.B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data set is available in [26].

Acknowledgments: The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Huang, P.; Xiao, L.; Soltani, S.; Mutka, M.W.; Xi, N. The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 101–120. [CrossRef]
- Zhu, J.; Lv, C.; Tao, Z. Performance Analyses and Improvements for IEEE 802.15.4 CSMA/CA Scheme in Wireless Multihop Sensor Networks Based on HTC Algorithm. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 452423. [CrossRef]
- Kaiwen, C.; Kumar, A.; Xavier, N.; Panda, S.K. An intelligent home appliance control-based on WSN for smart buildings. In Proceedings of the 2016 IEEE International Conference on Sustainable Energy Technologies (ICSET), Hanoi, Vietnam, 14–16 November 2016; pp. 282–287. [CrossRef]
- Sadeq, A.S.; Hassan, R.; Sallehudin, H.; Aman, A.H.M.; Ibrahim, A.H. Conceptual Framework for Future WSN-MAC Protocol to Achieve Energy Consumption Enhancement. *Sensors* **2022**, *22*, 2129. [CrossRef]
- Ahmad, T.A.; Esraa, S.; Ahmed, M.M.; Ibrahim, A.H.; Shaima, A.E. Deep Learning Based Hybrid Intrusion Detection Systems to protect Satellite Networks. *J. Netw. Syst. Manag.* **2023**, *31*, 82. [CrossRef]
- Fatima, T.Z.; Kamalrulnizam, A.B.; Adnan, A.A.; Khaled, M.A.; Tanzila, S.; Khalid, H.; Naveed, I. LLTP-QoS: Low Latency Traffic Prioritization and QoS-Aware Routing in Wireless Body Sensor Networks. *IEEE Access* **2019**, *7*, 152777. [CrossRef]
- Ning, H.; Wang, Z. Future internet of things architecture: Like mankind neural system or social organization framework? *IEEE Commun. Lett.* **2011**, *15*, 461–463. [CrossRef]
- Asadullah, M.; Raza, A. An overview of home automation systems. In Proceedings of the 2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), Rawalpindi, Pakistan, 1–2 November 2016; pp. 27–31. [CrossRef]
- Gladence, L.M.; Anu, V.M.; Rathna, R.; Brumancia, E. Recommender system for home automation using IoT and artificial intelligence. *J. Ambient. Intell. Hum. Comput.* **2020**. [CrossRef]
- Ahmed, E.; Walid, S. Design and analysis of data link impersonation attack for wired LAN application layer services. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 13465. [CrossRef]
- Mahmood, S.; Mohsin, S.M.; Akber, S.M.A. Network Security Issues of Data Link Layer: An Overview. In Proceedings of the 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 29–30 January 2020; pp. 1–6. [CrossRef]
- Bouani, A.; Maissa, Y.B.; Saadane, R.; Hammouch, A.; Tamtaoui, A. A Comprehensive Survey of Medium Access Control Protocols for Wireless Body Area Networks. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5561580. [CrossRef]
- Wadii, B.; Ayyub, A.; Anis, K.; Bilel, B.; Adel, A. Early detection of red palm weevil infestations using deep learning classification of acoustic signals. *Comput. Electron. Agric.* **2023**, *212*, 108154. [CrossRef]
- Safa, B.A.; Maha, D.; Henda, B.G. FedMicro-IDA: A federated learning and microservices-based framework for IoT data analytics. *Internet Things* **2023**, *23*, 100845. [CrossRef]
- Collotta, M.; Scatà, G.; Pau, G. A Priority-Based CSMA/CA Mechanism to Support Deadline-Aware Scheduling in Home Automation Applications Using IEEE 802.15.4. *Int. J. Distrib. Sens. Netw.* **2013**, *2013*, 139804. [CrossRef]
- Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* **2019**, *119*, 3–11. [CrossRef]
- Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1686–1721. [CrossRef]
- Fernandes, R.; D'Souza, G.L.R. A New Approach to Predict user Mobility Using Semantic Analysis and Machine Learning. *J. Med. Syst.* **2017**, *41*, 188. [CrossRef]
- Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, e00938. [CrossRef]
- Han, X.; He, Z. A Wireless Fingerprint Location Method Based on Target Tracking. In Proceedings of the 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Hangzhou, China, 3–6 December 2018; pp. 1–4. [CrossRef]
- Kumar, A.; Goyal, S.; Varma, M. Resource-efficient Machine Learning in 2 KB RAM for the Internet of Things. *Proc. Mach. Learn. Res.* **2017**, *70*, 1935–1944. Available online: <https://proceedings.mlr.press/v70/kumar17a.html> (accessed on 3 June 2023).
- Farnaaz, N.; Jabbar, M.A. Random Forest Modeling for Network Intrusion Detection System. *Procedia Comput. Sci.* **2016**, *89*, 213–217. [CrossRef]
- Cvitić, I.; Peraković, D.; Periša, M.; Gupta, B. Ensemble machine learning approach for classification of IoT devices in smart home. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 3179–3202. [CrossRef]
- Bühlmann, P. Bagging, Boosting and Ensemble Methods. In *Handbook of Computational Statistics*; Springer Handbooks of Computational Statistics; Gentle, J., Härdle, W., Mori, Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2012. [CrossRef]

25. Lin, W.; Wu, Z.; Lin, L.; Wen, A.; Li, J. An Ensemble Random Forest Algorithm for Insurance Big Data Analysis. *IEEE Access* **2017**, *5*, 16568–16575. [[CrossRef](#)]
26. GitHub Python Implementation of CSMA/CD Network Simulator. Available online: <https://github.com/thechausenone/csma-cd-simulation/tree/master/main> (accessed on 3 June 2023).
27. Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 160. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.