Communications and Network, 2025, 17(3), 55-79
https://www.scirp.org/journal/cn
ISSN Online: 1947-3826
ISSN Print: 1949-2421

Group Owner Selection Based on Artificial Neural Networks in Mobile Ad hoc Wi-Fi Direct Networks

Asmaou Abdou Goggo Petel[1], Rémy Maxime Mbala[1,2], Paul Dayang[1,2], Jean Michel Nlong[1,2], Julien Aristide Ngay[1]

[1]Laboratory of Mathematics, Computer Science and Applications, Faculty of Science, The University of Ngaoundéré, Ngaoundéré, Cameroon
[2]Department of Mathematics and Computer Science, Faculty of Science, The University of Ngaoundéré, Ngaoundéré, Cameroon
Email: asmaougoggo4@gmail.com, mbalaremy@gmail.com, piusday@gmail.com, julienaristidengay@gmail.com, jmnlong@yahoo.fr

How to cite this paper: Goggo Petel, A.A., Mbala, R.M., Dayang, P., Nlong, J.M. and Ngay, J.A. (2025) Group Owner Selection Based on Artificial Neural Networks in Mobile Ad hoc Wi-Fi Direct Networks. *Communications and Network*, **17**, 55-79.
https://doi.org/10.4236/cn.2025.173003

Received: May 1, 2025
Accepted: August 26, 2025
Published: August 29, 2025

Abstract

This paper investigates the use of Artificial Neural Networks (ANN) to enhance Group Owner (GO) selection in Mobile Ad hoc Networks based on Wi-Fi Direct technology. These networks are decentralised, with no fixed access point, and require optimal GO selection to ensure group persistence with maximum stability. Traditional GO selection methods, based on a single criterion such as Intent Value, can lead to inappropriate GO selection and fail to consider the heterogeneous and the dynamic nature of the network. To overcome this issue, we design a classification model and a regression model, both based on Artificial Neural Network techniques. The classification model identifies the most suitable node to act as the GO based on several parameters. It enables fast, efficient decision-making by directly selecting the GO from the available nodes, based on a binary output (0 for GO or 1 for noGO). The regression model provides a continuous estimate of Intent Value of each node based on several parameters collected on the node, offering a finer measure of a device's willingness to become a GO. These models were trained with a weighted dataset and evaluated using the performance metrics recommended for classification and regression in Artificial Neural Networks. The results show that these ANN models offer a promising solution for improving the management of ad hoc networks by providing more adaptive and intelligent GO selection decisions. Our approach ensures that the most capable node is elected as group owner.

Keywords

Artificial Neural Networks, Group Owner Selection, Wi-Fi Direct, Intent Value, Classification Model, Regression Model

1. Introduction

Mobile Ad hoc Networks (MANETs) have emerged as a cornerstone of modern wireless communication, enabling mobile devices to establish autonomous, infrastructure-free connections. These decentralized networks are particularly valuable in scenarios where traditional network infrastructure is unavailable or impractical, such as disaster recovery, military operations, or remote areas [1]. The self-organizing and self-configuring natures of MANETs and their dynamics make them well-suited for various applications, but they also introduce challenges related to resource allocation, topology management, and network stability.

Within the broader scope of MANETs, Wi-Fi Direct has gained prominence as a specialized protocol designed to facilitate Peer-to-Peer (P2P) communication between devices. Wi-Fi Direct eliminates the need for a traditional central Access Point, relying instead on the selection of a Group Owner (GO) to coordinate communications within a P2P group [2] [3]. The role of the GO is negotiated during the group formation phase and remains unchanged until the group is destroyed or reconfigured. The GO selection process is dynamic and essential for network performance and quality of service. Once elected, the GO acts like an access point, the only difference being that it is not fixed, therefore it is called Soft Access Point (SoftAP). In a group, the GO is responsible for resource management (channel access scheduling, power measurements), data routing (capability of forwarding messages between nodes), and maintaining network stability (coordination of intra and inter-group communications). Work and traffic are centralized at the GO since he is the only node allowed to cross connect nodes and he has knowledge of all node labels within a group. Consequently, the selection of the GO must be optimal to elect the best equipment with sufficient resources to manage the group. Knowing that groups are generally heterogeneous, where different types of equipment with different characteristics may be included. A number of studies have already been carried out to improve GO selection and the WiFi Direct group formation process [1] [4]-[7], but most of these studies are based on a single parameter and do not take into account all the factors that can influence device's ability to manage the group.

To overcome these limitations, there are a number of advanced techniques, such as Artificial Neural Networks (ANN), which are being developed as generalizations of mathematical models of human cognition or neural biology [8]. They offer promising solutions to complex problems thanks to their ability to learn and adapt [9]. The models used in ANNs make it possible to combine several input parameters to categorise objects (classification) or to predict a result (regression). Deep Learning and Artificial Neural Networks have already been used to solve cluster head election problems in Wireless Sensor Networks [10]-[14].

In this paper, we are exploring the use of ANNs to optimise the selection of GOs in Wi-Fi Direct groups. By addressing the limitations of traditional methods, the proposed approach consists to make two types of deep learning models (classification and regression) whose incorporates dynamic factors like signal strength, battery level, the amount of equipment supported, the degree of mobility, processor clock frequency, and Intent Value as input. Based on these parameters the classification model identifies the most suitable node to act as the GO. The regression model computes the Intent Value of each node based on the same parameters and compare it to the Intent Value chosen during the negociation to predict the node's ability to become GO.

The proposed classification and regression models are designed to work in synergy. The classification model enables fast, efficient decision-making by directly selecting the GO from the available nodes, based on a binary output (0 or 1). The regression model, on the other hand, provides a continuous estimate of Intent Value, offering a finer measure of a device's willingness to become a GO, so the node with the highest Intent Value is selected as the GO. This holistic approach, based on Machine Learning, provides a better response to the challenges of modern ad hoc networks, as it integrates multiple criteria and continuous evaluations for optimal designation of the GO in a group. By taking advantage of the learning capabilities of neural networks, our solution aims to ensure that the right GO is chosen, which will improve the overall performance of netwoks, ensuring smooth and reliable communication in various usage scenarios. To validate these models, we also propose in this paper an approach for electing the GO from among a set of neighbouring devices that are all within communication range.

The rest of the paper is organized as follows: Section 2 presents the overview of Wi-Fi Direct and related work on the GO selection; in Section 3, we provide the proposed models scheme; Section 4 provides theorical evaluation of the proposed models; Section 5 presents our GO selection approach based on the proposed models; conclusion and future works are presented in Section 6.

2. Overview of Wi-Fi Direct and Related Work

Wi-Fi Direct, also named Wi-Fi P2P, is a popular wireless communication technology that enables compatible devices such as laptops, smartphones, printers, smart TVs, cameras and other appliances to connect directly to each other without the need for an intermediate Access Point (AP) [15]. Unlike traditional infrastructure mode, where devices connect to a central access point, it allows Wi-Fi Direct-enabled divices to dynamically negotiate and select one of the mobile devices as Group Owner. The Group Owner, once elected for a group, must play the same role as an access point as in Wi-Fi infrastructure mode [16]. So the choice of GO must be optimal enough to build a sustainable group and keep the group running smoothly. In the subsequent sections, we provide a technical overview of Wi-Fi Direct and recent developments for the group formation and GO selection.

2.1. Technical Overview

According to [17], the Wi-Fi Direct protocol is based on the standard WLAN infrastructure mode. This network known as a P2P Group is a topology consisting of a mandatory P2P Group Owner (GO) and zero or more P2P Clients. A P2P Group is functionally equivalent to a Basic Service Set (BSS) in legacy Wi-Fi network and the GO takes over all the functions of a real Access Point. In this architecture, a P2P device, if implemented with multiple MAC functionality, can also support simultaneous operation by connecting to a P2P Group and to a conventional Wi-Fi Access Point [18]. **Figure 1** illustrates the Wi-Fi Direct architecture [18].

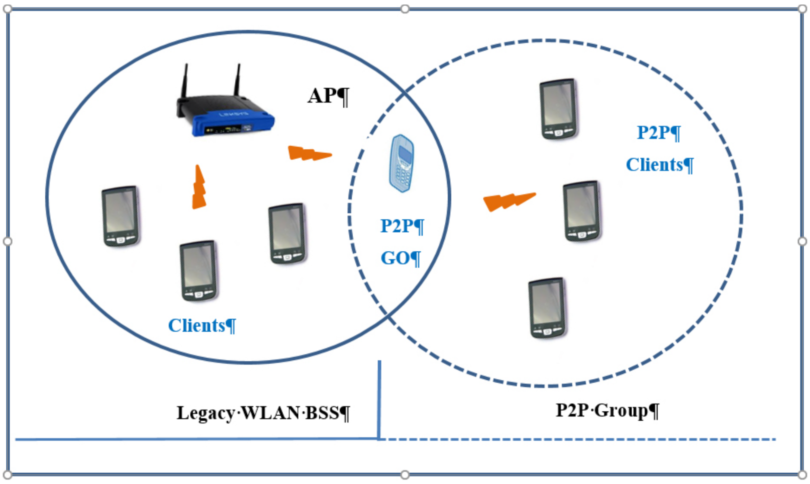

Figure 1. Wi-Fi direct architecture [18].

Several features have been implemented in Wi-Fi Direct technology such as: Device Discovery, Service Discovery, Group Formation, Power Saving Schemes and Security [3] [15]-[18].

The P2P Device Discovery procedure enables two devices to find each other in their wireless range and establish a connection. It is based on the standard discovery method used in traditional Wi-Fi, involving the scanning and finding phases denoted in the IEEE 802.11-2012 standard [19]. During the scanning phase, a P2P device performs traditional Wi-Fi scan (passive scan) through all supported channels in order to collect information on existing P2P groups and Wi-Fi networks. In the finding phase, the P2P Device alternates between two states (Search and Listen) in order to come to a common channel for the communication.

Service Discovery is an optional procedure in Wi-Fi Direct. The procedure starts after the Device Discovery and prior to the Group Formation procedure. By using Generic Advertisement Service (GAS) protocol [20]. It allows a P2P Device to limit the search to specific P2P devices or types and to connect to the latter only if they offer the intended service [15].

After successfully completing the Device Discovery procedure, which is mandatory, and the optional Service Discovery procedure, P2P Devices can establish the P2P Group. During the Group Formation, the device that will act as GO is determined. Three types of P2P Group Formation schemes are possible in Wi-Fi Direct: Standard Group Formation, Autonomous Group Formation and Persistent Group Formation.

- In Standard Group Formation, two P2P Devices negotiate the role of the P2P GO. The GO Negotiation is a three-way handshake (GO Negotiation Request/Response/Confirmation). During the handshake, the two devices send to each other a randomly chosen numeric value called Intent Value. The Intent Value ranges from 0 to 15. It measures the desire of the P2P Device to be the P2P GO. The P2P Device sending the higher Intent Value shall become GO. In case both P2P devices send equal GO Intent values, a tie breaker bit is used for decision and the device with tie breaker bit set to 1 shall become GO. Figure 2 illustrates the native procedure for the GO selection, based on the Intent value comparison between two P2P devices during Standard Group Formation.

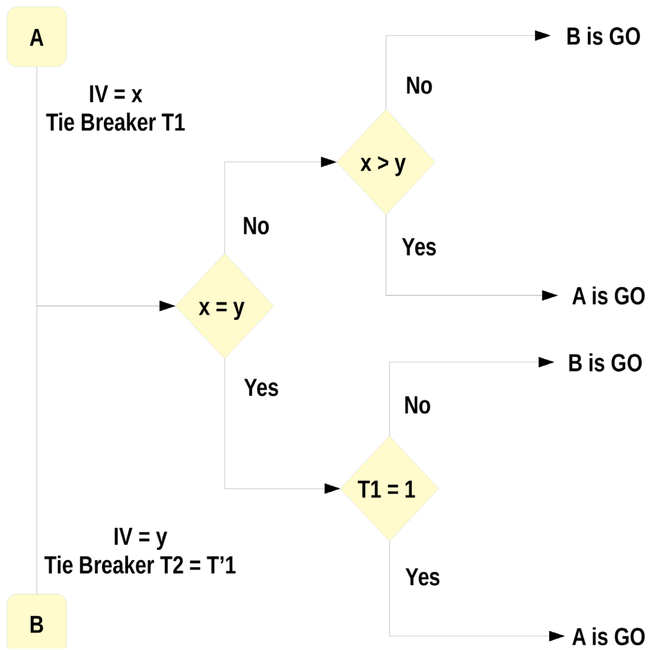

Figure 2. Native procedure for GO selection.

- In Autonomous Group Formation, the role of GO is not negotiated. Instead, a P2P Device announces itself as GO and starts sending Beacons. This process is very similar to the legacy Wi-Fi in which an AP directly sends Beacons into the network to become discoverable.
- In Persistent Group Formation, by using P2P invitation procedure, a P2P Device sends an invitation to another P2P Device, which was previously connected to it in a P2P Group, in order to reinstantiate the P2P Group. Thus, the group should first be declared as persistent so that if a P2P device recognizes that it has established a persistent group with another corresponding device in the past, either P2P device can quickly re-establish the group.

In Wi-Fi Direct, the P2P GO, which acts as a Soft-AP, must be a battery powered device and have limited lifetime. Hence, Wi-Fi Direct introduces two novel schemes for power saving in the P2P Devices. These schemes are: Opportunistic Power Save (OppPS) [17] and Notice of Absence (NoA) [17].

Wi-Fi Direct requires all P2P Devices to implement Wi-Fi Protected Setup (WPS) [21] in order to secure the connection establishment process and communication in the P2P Group. In WPS scheme, the P2P GO implements the internal Registrar whereas the P2P Client implements Enrollee. The internal Registrar generates and issues the network credentials to Enrollee. The Enrollee (P2P Client) reconnects to the internal Registrar (P2P GO) using the new credentials.

2.2. Related Work on Group Formation and GO Selection

Several studies have been carried out on Group Formation and GO selection, which is a critical aspect directly affecting the performance of Wi-Fi Direct networks, including latency, power consumption, quality of service, throughput and so on.

Authors in [22] proposed multi-hop communication in Wi-Fi Direct using P2P Concurrent Device and Ad-hoc On Demand Distance Vector (AODV) reactive routing protocol [23]. Routing is established during Device Discovery phase to allow multi-hop communication.

In [24], authors proposed EMC (Efficient Multi group formation and Communication) protocol for Wi-Fi Direct. In EMC scheme, the P2P GO is elected based on remaining battery status of the device.

In [5], authors proposed a dynamic election of P2P GO in Wi-Fi Direct. The elected leaders can be replaced dynamically based on clustering strategy such as battery status, speed of the user and direction of motion etc. The authors also presented a template for writing clustering algorithms for efficient Group Formation.

Authors in [18] proposed a combined metric approach to select the P2P GO based on several parameters. The parameters are normalized and weighted to compute Intent Values of each device. An election algorithm is then used to select the P2P GO. The same idea is explored in [7], but the authors go further by proposing a framework for multi-hop ad hoc networking using Wi-Fi Direct in Android smart devices. The framework includes a connection establishment protocol and a group management protocol.

The work carried out in [16] proposes a modified group formation scheme among multiple devices. The proposed scheme formulates the GO selection problem as an optimization problem which is solved using integer programming (IP). The GOs are selected based on link capacities with the objective to maximize the overall network throughput.

In [25], authors proposed an alteration to the standard Group Formation. In their proposed scheme, during GO Negotiation, each device sends an flag bit set to 1, to indicate its willingness to serve as the so called Emergency Group Owner (EGO). The EGO is present in the group after the GO is selected. The EGO takes over the group and assumes GO role if the GO leaves voluntarily or involuntarily.

In [26] the authors propose the GO selection method by maximizing bit rate.

However, the performance of the proposed system is severely acted when GO selection is required for Internet connectivity via a Wi-Fi access point or a cellular base station (BS). Wi-Fi access point or a cellular base station (BS). One of the reasons for the poor performance of this method is that the quality of the link between the GO and the AP/BS imposes a bottleneck on the average network throughput. Furthermore, the work is limited to the formation of a single group, and the proposed scheme cannot be used to create several P2P groups to connect users with high population density [16].

Another method is proposed by [27]. This method is particularly useful during the device discovery phase. WD2 applies a parameter to control the number of messages sent. Before sending the probe packets, each device generates copies which are sent consecutively. Assuming there are N devices that can discover each other, receiving multiple packets enables the devices to discover each other [27]. Despite its advantages, WD2 introduces an additional overhead due to the increased number of messages exchanged to accurately measure RSSI and calculate the intent value.

None of the works presented above explores the possibility of using techniques based on Artificial Intelligence (AI), in particular Artificial Neural Networks (ANN), to elect the GO.

3. Proposed ANN Models

Our principal aspect in this research is the need to improve GO selection to ensure stable connectivity and energy efficiency in Wi-Fi Direct networks. Traditional methods, often based on simple criteria such as Received Signal Strength Indicator (RSSI), do not take into account the complexity and dynamics of modern ad hoc networks. They can lead to sub-optimal GO choices, resulting in reduced quality of service, increased energy consumption, and network instability. Faced with these limitations, our aim is to design models based on Artificial Neural Networks (ANNs) for optimized GO election and calculation of the Intent Value, a quantitative measure of a device's willingness to become a GO. In this section, we present the parameters used to collect the data and the proposed architecture for the classification and regression models.

3.1. Dataset Preparation

Data quality is essential for effective training of Artificial Neural Network models. Machine Learning techniques are designed to identify and exploit hidden patterns in data for predicting the outcome of future events for classification and regression problems [28]. Data collection is an important step since Machine Learning require representative data, possibly without bias, to build an effective Machine Learning model for our problem. In this section we describe the key steps in data preparation, which include data collection, pre-processing and division into training and test sets.

1) **Data collection**: The data used to train our models come from simulations,

real measurements in ad hoc network environments and synthetic data obtained from various repositories [29]-[32]. Features collected from our dataset include:

(a) **ID**: This feature simply refers to the device ID.

(b) **Received Signal Strength Indicator (RSSI)**, which measures the quality of the signal received by a device. The data rate of the wireless link is badly affected by low RSSI. If the P2P group is used for content distribution for example, the strong connection between the group members and the GO is more crucial. The RSSI of GO to group members have significant impact on the P2P group performance.

(c) **Battery level**, indicating the amount of energy remaining in the device. P2P devices, including the GO, are battery powered devices. If the battery life is not considered in electing a GO, there is a probability that a P2P device having low value of remaining battery is elected as GO. The GO being the most active device in the P2P group would exhaust soon and the P2P group will be broken.

(d) **Processing capability (CPU clock and memory)**, which reflects the device's computing and storage capacity. The device which becomes GO shall be equipped with enough processing power and large memory to better serve the connected clients. The processing power and memory requirements for GO might become more significant when the P2P group consists of a large number of nodes and the group is intended for multimedia application.

(e) **Number of neighbors**: Number of devices in direct range. A node with more neighbors can ensure better connectivity within the network. Because the P2P group is intended to connect large number of devices, then it is important to elect as GO the P2P device which has more devices in its range to connect as clients. There should also be a limit on maximum number of nodes in a P2P Group. In our case, this is the maximum number of devices that can be connected simultaneously (the amount of device it can support without compromising network quality).

(f) **Device mobility**: Indicates the speed of movement of each device. In realistic mobile ad-hoc environments, each mobile host is free to change its mobility parameters randomly at any time [33]. Less mobile devices are preferred to become GO to minimize frequent topology changes and to make the group more stable. There are several models of mobility, such as Gauss-Markov Mobility (GMM), Random Waypoint (RWP), Random Walk (RW), Random Direction (RD), and Recurrent Self-Similar Gauss-Markov (RSSGM) [33]. In our case, the data presented for this feature were obtained using the RSSGM model, which shows the greatest accuracy compared to other models [31] [33].

These data, obtained from 356 pieces of equipment are then aggregated to form a dataset representative of the different network conditions that the models will have to manage.

2) **Data preprocessing**: Pre-processing is an important step to ensure that the data are in an optimal format for training ANN models. The following steps are applied:

(a) **Data cleaning**: Missing or abnormal data (such as outliers) are processed. Missing values can be imputed using the mean or median of the other values in the same column. In our dataset, we do not have any missing data or outliers.

(b) **Normalization**: Features such as RSSI, battery level and CPU clock can have very different scales. To ensure that each feature contributes equally to model training, data are normalized to a standard range (usually between 0 and 1). Here we have used the Min-Max Scaler method [34]. The formula to scale a feature is given by **Equation 1**, where x, $x_{\min}$ and $x_{\max}$ are respectively the value to scale, the minimum value and the maximum value of the dataset.

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

(c) **Categorizing feature values**: In order to determine the suitability of a device and to understand the values of these features, we have graded these values according to different mentions. This procedure is described in **Table 1** and **Table 2**, which present the categorization the RSSI, Battery and CPU level feature values respectively. For RSSI, we use the categorization proposed in [26].

Table 1. Categorization of RSSI levels by received power (P_r).

RSSI Level	Min P_r	Max P_r	Description
0	$-\infty$	−81	Very bad signal
1	−81	−78	Bad signal
2	−78	−73	Average signal
3	−73	−65	Good signal
4	−65	0	Excellent signal

Table 2. Battery and CPU level categorization.

Level	Battery in (%)	CPU Clock in (MHz)	Description
0	0 - 20	1000 - 1300	Very Bad
1	30 - 50	1400 - 1600	Bad
2	60 - 80	1700 - 2000	Good
3	80 - 100	2100 - 3000	Very Good

As one of our models is a binary classification model then for the Intent Value column we will categorize the values into two classes 0 and 1 according to **Table 3**.

Table 3. Intent value classes.

Intent Value	Classe	Description
0 - 7	0	noGO
8 - 15	1	GO

This distribution is justified by the fact that in the classic GO selection, the role is assigned to the device that has emitted the highest value of intent. We have therefore considered values between 0 and 7 to be low and those between 8 and 15 to be high. And 8 is the midpoint of the range [0, 15]. In the Wi-Fi Direct protocol, two Intent values are compared at the time of negotiation, whereas our predictive models only know the local characteristics (RSSI, battery level, memory, etc.) of a device and try either to predict whether this node has a good chance of becoming a GO, or to predict what its Intent Value might be. Consequently, as our models have no knowledge of the other nodes at the time of prediction, we simply teach it to classify nodes as GO or noGO according to their Intent Value level in absolute value. This is where the empirical threshold of 8 comes into play. This threshold is not a protocol truth, but a practical approach that we have used to formulate the problem in binary classification, which makes it possible to identify nodes with a high propensity to become GOs where several nodes will be compared. This balanced cut-off also makes it possible to create a symmetrical binary classification, leading to a more balanced model.

3) **Data Processing Division**: To properly evaluate model performance, we have used the standard Machine Learning distribution. The dataset is divided into three distinct sets:

- **Training Set**: Representing 70% of the dataset, this set is used to train the classification and regression models. This majority share enables the model to learn the relationships between inputs and outputs. Sufficient data are required for the model to generalise.
- **Validation Set**: Representing 15% of the data set, it is used to adjust model hyperparameters and prevent overfitting. The model does not see this data during training, which gives a good indicator of the performance of the data during adjustment.
- **Test Set**: The remaining 15% is used to evaluate the final performance of the models on data they have never seen before.

This division allows us to ensure that the models generalize well beyond the data on which they have been trained.

3.2. Classification and Regression Models

Our classification (**Figure 3**) and regression (**Figure 4**) models are structured in the form of a multi-layer neural network (Multi-Layer Perceptron, MLP). MLP is composed of multiple layers, including an input layer, hidden layers, and an output layer, where each layer contains a set of perception elements known as neurons. The neurons of the input layer of our model is composed of the device features mentioned in section 2, except for the ID.

Both networks include two hidden layers (each with eight neurons) using the Rectified Linear Unit (ReLU) as activation function (**Equation 3**). The ReLU function is chosen for its acceleration of learning and mitigation of the problem of the gradient disappearing. In fact, the simplicity of the ReLU reduces the time required to perform calculations during forward and reverse propagation in the network, which speeds up the learning process. ReLU tends to activate only some of the neurons at a time, making the network lighter and less costly in terms of computation [35]. In addition, unlike other functions, the ReLU allows a freer flow of the gradient, avoiding problems where the gradients become too small for effective updating during training [35]. Our models have 2 hidden layers, each consisting of 8 neurons. The choice of these quantities was guided by the tests carried out during the training. In ANN, the number of hidden layers and the number of neurons generally depend on the number of input features and the size of the dataset [11]. In our case, we have observed that increasing these values tends to overfitting.

The output layers of both models are formed by a single neuron. For the classification model, since we are doing a binary classification a Sigmoid activation function (**Equation 4**) is used for the output, producing a probability indicating whether a device is suitable for election as a GO or not. Beside the classification model, the regression model calculates the Intent Value of each node. Precise calculation of this value enables more nuanced and dynamic decision-making, by adjusting the probability of a device being chosen as a GO according to changing network conditions. So the output layer of our regression model is activated with Linear activation (**Equation 5**), to produce a continuous value representing the Intent Value, ranging from 0 to 15 according to the Wi-Fi direct technology specification.

If we note $Y^{(n)} = \left\langle y_1^{(n)}, y_2^{(n)}, \cdots, y_m^{(n)} \right\rangle$ the set of neurons in the n^{th} layer, the functioning of the neurons in the model is broken down into two phases:

- **Aggregation phase**: It consists of calculating the weighted sum ($X_i^{(j)}$) of the neuron's inputs, to within one bias ($b^{(j)}$), as shown in **Equation 2**:

$$X_i^{(j+1)} = \sum_{k=1}^{m} w_{ik}^{(j)} y_k^{(j)} + b^{(j+1)} \tag{2}$$

- **Activation phase**: The neurons are then activated, to give an output value, by applying an activation function to the aggregated values. The output values of the hidden layers are obtained by **Equation 3**:

$$Z_i^{(j)} = \max\left(0, X_i^{(j)}\right) = \begin{cases} X_i^{(j)}, \text{if } X_i^{(j)} > 0 \\ 0, \text{otherwise} \end{cases} \tag{3}$$

The output values of the two models are obtained at the output layer level by **Equation 4** for the classification model and **Equation 5** for the regression model.

$$Y = \frac{1}{1 + \mathrm{e}^{-Z_i^{(j)}}} \tag{4}$$

$$Y = \begin{cases} 0, & \text{if } Z_i^{(j)} \le 0 \\ Z_i^{(j)}, & \text{if } Z_i^{(j)} \in \left]0,15\right[\\ 15, & \text{if } Z_i^{(j)} \ge 15 \end{cases} \tag{5}$$

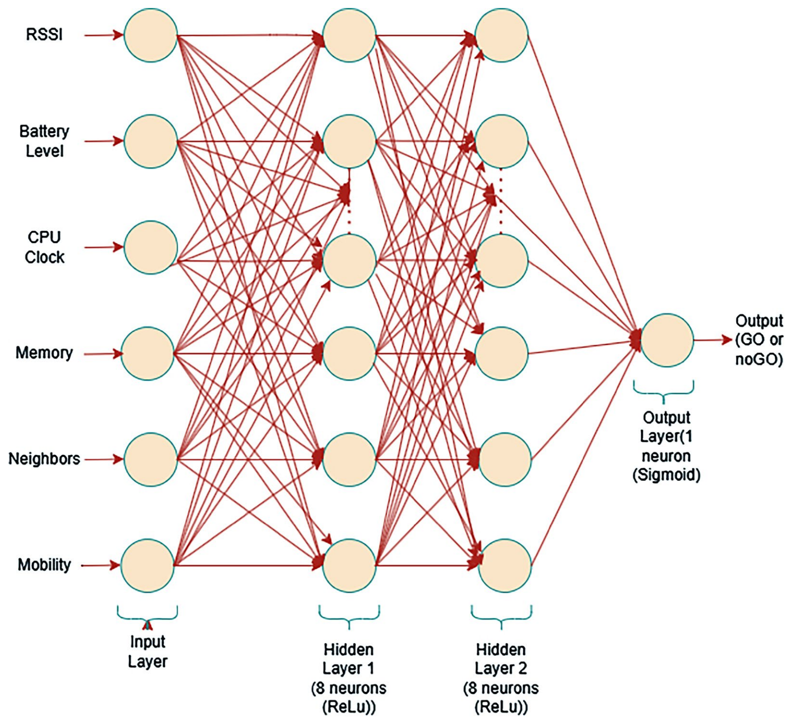

Figure 3. The classification model

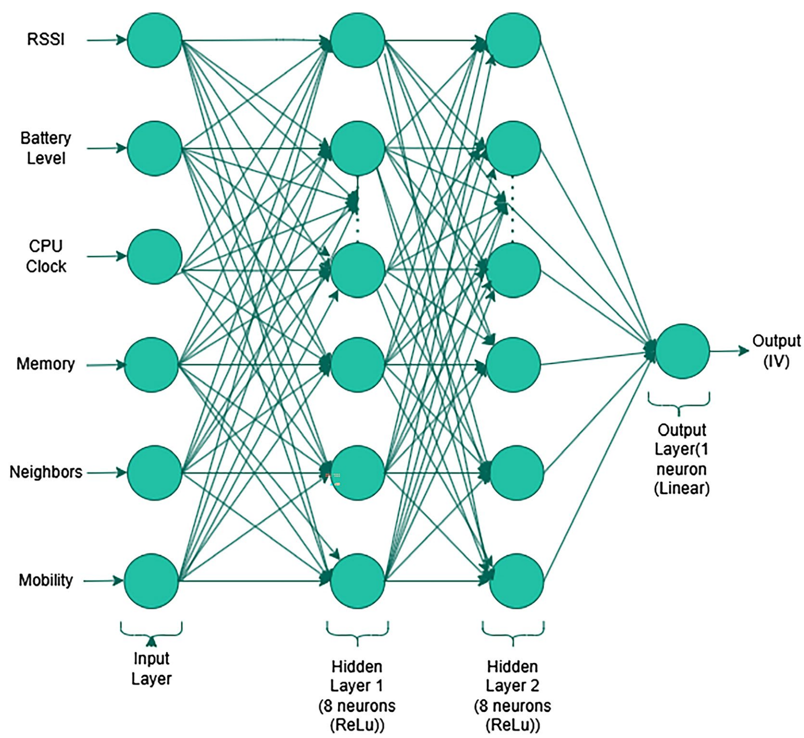

Figure 4. The regression model

4. Models Performance Evaluation

In this section we carry out a theoretical evaluation of our two models for GO selection and Intent Value calculation in Wi-Fi Direct ad hoc networks. These models aim to improve network performance and stability by using more complex and adaptive devices features than traditional methods. In this section we also present the experimental environment and evaluation indicators specific to each of the models.

4.1. Experimental Environment

The hardware and software conditions in the experimental environment of this article are shown in Table 4.

Table 4. Experimental environment.

Software and hardware configuration	Configuration parameter
CPU	AMD Ryzen5-3500U @2.1 GHz
RAM	8G
Operating System	Windows 11 Professional 64-bit
Programming language	Python 3.7.0
Deep learning framework	Tensorflow 1.14
Deep learning library	Keras 2.3.1
Data visualization library	Matplotlib 3.10.1

4.2. Selection of Evaluation Indicators

In order to evaluate the GO selection and Intent Value prediction performance of our models, several evaluation indicators are selected, depending on the type of model.

For the classification model, these indicators are:

- **Confusion matrix** compares the model's predictions with the actual results, showing the numbers of correct and incorrect predictions in four categories: True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN), providing an overview of the effectiveness of the model for classifying data.
- **Accuracy and precision** measures the proportion of correct predictions made by the classification model in relation to all predictions made. The accuracy metric (**Equation 6**) is defined as the proportion of true predictions T for each class $C_i \quad \forall i = 1, \cdots, N$ among the total number of predictions.

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} T_{C_i}}{\text{Total Predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

The precision of the model can be formally defined as the frequency of correct predictions for actual positive instances as in **Equation 7**:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

The accuracy curve shows the evolution of the model's accuracy during training, for both the training set and the validation set. It shows whether the model continues to improve its predictions.

- **Recall or True Positive Rate (TPR)**; also known as sensitivity, describes the number of correct predictions is inferred from the confusion matrix as in **Equation 8**.

$$\text{TPR}\left(\text{Recall}\right)=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{8}$$

- **F-measure** is a compromise measure between precision and recall, calculated as the harmonic mean of these two metrics (**Equation 9**).

$$\text{F-measure}=2\times\frac{\text{Precision}\times\text{TPR}}{\text{Precision}+\text{TPR}} \tag{9}$$

- **ROC curve and AUC** is a graphical tool that can be used to evaluate the performance of a classification model by measuring the trade-off between the False Positive Rate (FPR) and the True Positive Rate (TPR) at different decision thresholds. The Area Under the ROC curve (AUC-ROC) is also calculated to quantify the overall performance of the model.
- **Loss curve** shows the evolution of the cost function (or loss) during model training, for both the training set and the validation set. It shows whether the model continues to learn efficiently or whether it starts to overfitting at a certain point. For our model, the loss function used is the Binary Cross-Entropy [36], adapted for binary classification tasks.

The regression model designed to calculate the Intent Value is evaluated using specific metrics that measure the accuracy of the predictions of continuous values. Noting that n is the number of samples, y is the observed Intent Value, $\hat{y}$ is the predicted Intent Value and $\overline{y}$ is the mean, these parameters are:

- **The Median Absolute Error (MedAE)** measures the median of the absolute errors between predicted and actual values. Its formula is given by **Equation 10**.

$$\text{MedAE}\left(y,\hat{y}\right)=\text{med}\left(\left|y_i-\hat{y}_i\right|\right) \tag{10}$$

where med is the median function calculated on absolute errors $\left|y_i-\hat{y}_i\right|$.

- **The Mean Absolute Error (MAE)** is another metric that measures the average of absolute errors, providing a direct view of the average differences between predictions and actual values. MAE is calculated by **Equation 11**.

$$\text{MAE}\left(y,\hat{y}\right)=\frac{1}{n}\sum_{i=0}^{n-1}\left|y_i-\hat{y}_i\right| \tag{11}$$

- **The Mean Square Error (MSE)** calculates the average of the squares of the differences between the predicted values and the actual values. It is defined as follows (**Equation 12**):

$$\text{MSE}\left(y,\hat{y}\right)=\frac{1}{n}\sum_{i=0}^{n-1}\left(y_i-\hat{y}_i\right)^2 \tag{12}$$

- **The R^2 Score** measures the proportion of the variance in the target values that is explained by the model. It is defined by **Equation 13**.

$$R^2\left(y,\hat{y}\right)=1-\frac{\sum_{i=0}^{n-1}\left(y_i-\hat{y}_i\right)^2}{\sum_{i=0}^{n-1}\left(y_i-\overline{y}\right)^2} \tag{13}$$

- **The MAE curves** illustrate how the average error between predicted and actual values changes of the regression model is trained.

4.3. Results and Interpretation

After training our models by applying a weighting to the other features to generate the "Intent Value" column in our dataset, we were able to adjust the target to better reflect the underlying relationships between inputs and outputs (**Equation 1**). This strategy created consistency between the inputs and the target variable, which improved model performance. This weighting process strengthens the links between relevant features and the target, which is often essential in ANN based approaches.

The performance curves obtained from the classification model are shown below:

- In **Figure 5**, which presents the ROC curve and AUC, the ROC curve is close to perfection, with an AUC of 1.00, which is ideal. This indicates that the model has an excellent ability to distinguish between classes 0 and 1 without any error.
- The training loss curve (blue line) and the validation loss (orange line) curve are shown in **Figure 6**. Both curves decrease steadily with the number of epochs. This means that the model is improving with each epochs, reducing the prediction error. What's more, the loss decreases for both training and validation sets, indicating that the model is learning well without noticeable overlearning.
- **Table 5** shows the confusion matrix for both classes. For the Class 0 (noGO), the model correctly predicted 30 instances of noGO (true negatives) but it misclassified 9 instances as GO when they were noGO (false positives). For Class 1 (GO), the model correctly predicted all 38 instances of GO (true positives). No GO node was misclassified as noGO (zero false negatives).

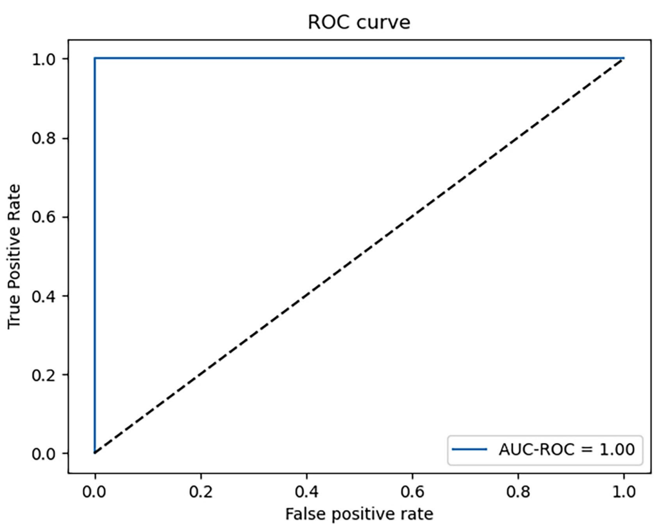

Figure 5. ROC curve and AUC.

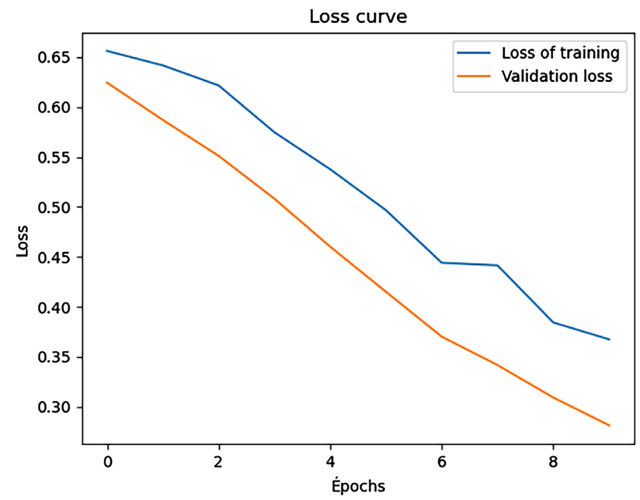

Figure 6. Loss curve of the classification model.

Table 5. Confusion matrix for classification model.

		Actual instance	
		Positive (P)	**Negative (N)**
Predicted Values	**Positive (P)**	38	9
	Negative(N)	0	30

- The curves presented in **Figure 7** show the evolution of training accuracy (blue line) and validation accuracy (red line). Accuracy increases steadily for both sets (training and validation). The red curve reaches a high accuracy (0.90) after a few epochs, while the training curve gradually catches up, suggesting that the model has a good generalization capacity and a very good validation accuracy (approx. 88% - 90%), demonstrating that it is still effective in distinguishing classes and is not overlearned, as the curves follow a similar trajectory without significant deviation.

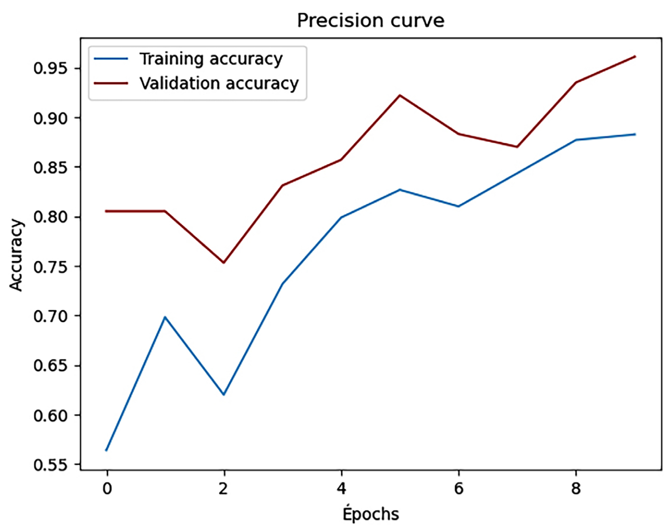

Figure 7. Accuracy curves of the classification model.

A summary of the performance of our classification model is presented in Table 6.

Table 6. Summary of the performance of the classification model.

Metrics	Class 0 (NoGO)	Class 1 (GO)	Global	Interpretation
Overall loss	-	-	0.3308	The overall loss is relatively small, suggesting that the model fits the data well.
Accuracy	1.00	0.81	0.8831	100% of predictions for noGO are correct, and 81% for GO. The model correctly classifies around 88% of nodes, indicating good overall accuracy.
Recall	0.77	1.00	-	The model detects 77% of noGO nodes, but captures all GO nodes.
F1-Score	0.87	0.89	-	The balance between precision and recall is excellent for both classes, indicating a good compromise for each class but slightly higher for class 1 (GO).

The classification model used to elect the Group Owner (GO) in a Wi-Fi Direct network showed solid performance with an overall accuracy of 88%, an AUC of 1.00, and a minimum loss of 0.33. The use of class weighting better managed the potential imbalance between GO and noGO classes, improving the model's ability to correctly identify nodes that could be GO. This ensures a more accurate selection of the GO, resulting in smaller architecture and very good performance.

The performance of our regression model was obtained by calculating the metrics presented in section 4 (**Equations 10, 11, 12, and 13**), that measure the accuracy of continuous value predictions. These performances are summarised as follows:

- In Loss curves for Regression presented in Figure 8, the loss drops rapidly during the first epochs and then stabilizes, with a slight variation. This shows that the model has quickly learned to reduce the error. In the same figure, the training curve (in blue) and validation curve (in orange) are very close to each other, which means that there is no overlearning. The model is able to generalize well on data not seen during training.
- In MAE curves (Figure 9), the MAE falls in a similar way to the loss during the first epochs (around 10 epochs). After this initial drop, the error stabilizes at around 1 for training and slightly below 1 for validation, showing that the model predicts value intents relatively well, with an average error of around 1 unit.
- The Prediction curve presented in Figure 10, shows a strong correlation between actual and predicted values, i.e. the points are close to the red diagonal line (representing a perfect match), indicating a good match between model

predictions and actual values. There is also a slight deviation in some places, with a few points deviating slightly from the diagonal line, particularly for higher values (around 14 or 15). This suggests that the model is slightly less accurate for extreme values, but overall, it is able to predict the majority of values with acceptable accuracy.

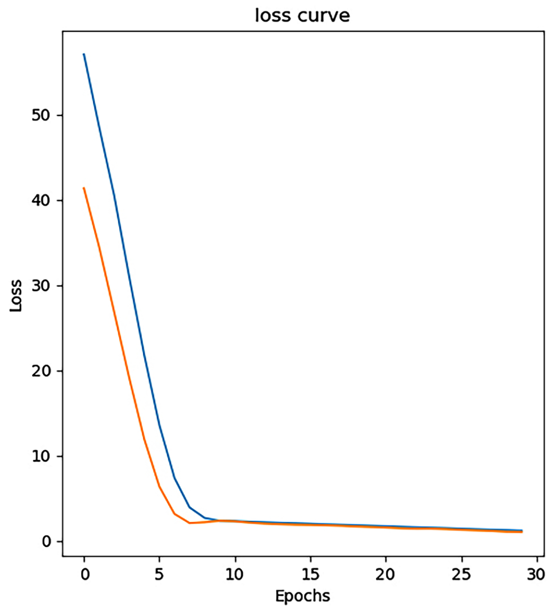

Figure 8. Loss curves of the regression model.

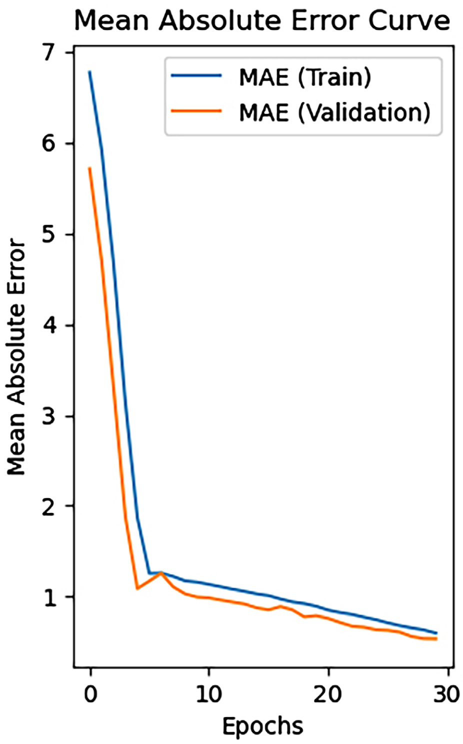

Figure 9. MAE curves.

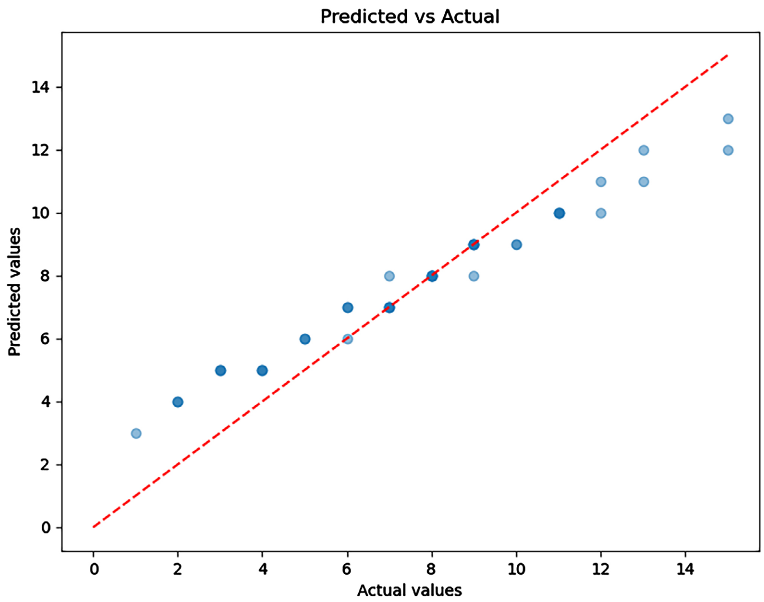

Figure 10. Prediction vs. actual curve.

In order to visualize the intent values prediction results, we have represented the 10 first instances in Table 7.

Table 7. Actual and predicted values of the regression model.

Nodes	0	1	2	3	4	5	6	7	8	9
Actual Value	3	8	9	9	2	6	9	5	6	11
Predicted Value	4	8	8	9	3	6	9	5	6	10

Table 8. Summary of the performance of the regression model.

Metrics	Value	Interpretation
MedAE	1.00	Half of the predictions have an absolute error of less than or equal to 1 unit, indicating stability in the model's predictions.
MAE	0.6104	A MAE of 0.61 indicates that, on average, the error between predicted and actual values is around 0.61 units, which indicates good overall accuracy.
MSE	0.7403	A relatively low MSE shows that the squared error between the actual and predicted values is moderate, indicating a good predictive capacity for the model.
R^2-Score	0.9416	The high R^2-Score means that the model captures 94% of the variance in the data, which shows that it explains the relationship between the input variables and the output well.

With an MSE of 0.74, a MAE of 0.61 and an R^2 of 0.94, our regression model with the weighted dataset demonstrated solid performance, indicating that it explained 94% of the variance in the data. The error and loss curves show fast learning and efficient generalisation, without overlearning. The majority of the predictions match the actual values, although differences are apparent for the extreme values of Intent-Value (14-15). Overall, the model is accurate and stable. **Table 8** summarises the performance of the Regression model.

5. Group Owner Selection

In this section, we present a new GO selection approach (GO-ANN), that incorporates the artificial neural network models presented in Section 3. This mechanism, shown in **Algorithm 1**, allows a device to be selected to act as the GO from a set of devices that are all within direct communication range of each other.

In fact, the Wi-Fi Direct specification restricts the GO negociation procedure to two devices i.e. only two interested devices can form a P2P group where one becomes GO and then the GO will announce its presence by sending beacons like an access point. Other P2P devices and also legacy Wi-Fi stations can join the group later as clients. This limitation has several implications for the performance for example, in several applications where more than two devices are in a shared wireless range and need to from a group to communicate efficiently and quickly. In this case, the role of GO must be negotiated automaticaly from all devices in the same wireless range and mutually neighbors.

Our GO election process, called GO-ANN and presented in **Algorithm 1**, will elect a GO from among several neighbouring devices in the same coverage area. It is detailed as follows: In the beginning, each node discovers its neighbors. Each P2P device computes its Intent Value (IV_C) based on our Regression Model. It also generates another Intent Value (IV_G) by using the classic Wi-Fi Direct method. These parameters are sent to a remote P2P device via a packet (*LocalDeviceFrame*) embedded in the GO Negotiation Request. Upon receipt of the GO Negotiation Response, the parameters of the remote P2P device, contained in the *RemoteDeviceFrame* paquet are obtained. The classification model is then activated to identify the most suitable node to act as GO between the two pieces of devices. If the result of the classification is 0, the local device can be the GO, therefore the remote device is disqualified to the selection process. On the contrary, if the result of the classification is 1, the remote device can be the GO and the local device disqualifies itself from the selection process.

More specifically, let us assume three P2P devices: A, B and C. Node A sends a GO Negotiation Request message to Node C. Node C activates the classification model and finds that Node A is the most suitable node to act as GO. Node C sends its parameters through GO Negotiation Response frame and disqualifies itself for the election. Node A is still active, but does not announce itself as GO, in contrast to the standard protocol operation. Node A shares its parameters with Node B and finds that Node B is the most suitable node. Node A disqualifies itself and stops participating in the election. Node B is still active in the election. Similarly, each node continues to participate in the election until it finds another node most suitable than itself or if it has performed GO Negotiation with all discovered devices.

The election process ends with a single node elected as Group Owner.

Input: SetOfDevice; Local Device Parameters (RSSI, Battery Level, CPU Clock, Memory, Neighbors, Mobility)
Output: Elected GO

foreach *(Device $\in$ SetOfDevice)* **do**
    $IV_C \longleftarrow Regression(Parameters)$; /* *Compute the Intent Value using the Regression Model with Local Device Parameters.* */;
    $IV_G \longleftarrow Generate(IV)$; /* *Generate the Intent Value using the native Wi-Fi Direct method.* */;
    if $(IV_C <= IV_G)$ **then**
        $LocalDeviceFrame \longleftarrow Parameters$; /* *Form the package with Local Device Parameters.* */;
    else
        $LocalDeviceFrame \longleftarrow IV_G$;
    end
    Send *Group Owner Negotiation Request* to the Remote Device with the $LocalDeviceFrame$;
    Receive *Group Owner Negotiation Response* from the Remote Device with the $RemoteDeviceFrame$;
    $Decision \longleftarrow Classification(LocalDeviceFrame, RemoteDeviceFrame)$; /* *Identification of the most suitable node to act as GO.* */;
    if *(Decision = 0)* **then**
        The Remote Device is disqualified;
    end
    while *(Decision = 1)* **do**
        The Local Device is disqualified;
    end
end

Algorithm 1. Group owner selection (GO-ANN).

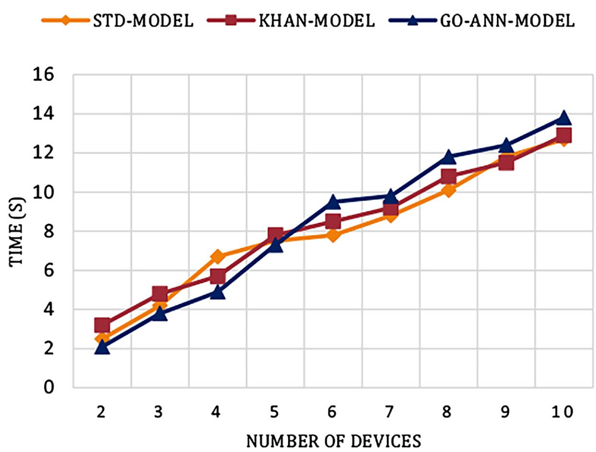

Figure 11. GO election time.

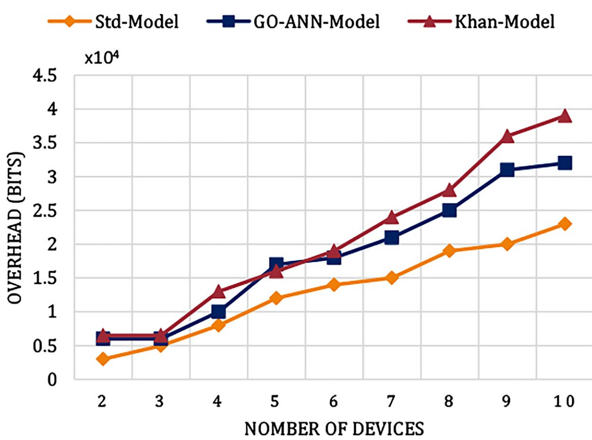

Figure 12. GO election overhead.

This algorithm was simulated on OMNET++ to evaluate the GO election time and the overhead level based on the number of devices discovered. The simulations were carried out considering 10 P2P devices whose parameters were randomly generated in accordance with the values defined in the dataset. The random nature of these parameters allows us to better evaluate the effectiveness of our algorithm, since the selected GO can change from one simulation to another. For each number of devices (ranging from 2 to 10) in the group, the simulation was run 50 times, and each time the time taken and the level of network overload due to message exchange were recorded.

Figure 11 shows the average GO selection time based on the number of devices. It can be seen that the average time taken to elect the GO increases with the number of devices discovered in the group. But when the number of devices is low ($\leq$ 5), the average election time of the GO with our approach is lower than with the standard election model and the model proposed by Khan *et al.* in [18]. This time begins to exceed other methods when the number of devices exceeds 5. This is certainly due to the calculations performed by the regression and classification models; these calculations become significant as the number of devices increases.

For the overhead shown in **Figure 12**, it can be seen that the network overload increases with the number of devices. This is because the number of messages exchanged increases as the number of devices increases. Nevertheless, although our approach generates more overhead than the standard method, it remains better than the approach used by Khan *et al.* in [18].

6. Conclusion and Future Works

In sum, this work has demonstrated that Artificial Neural Networks (ANN) represent an innovative and effective solution for improving Group Owner (GO) selection and Intent Value calculation in Wi-Fi Direct mobile ad hoc networks. Ad hoc networks, by their very nature, are decentralized and dynamic, and present numerous challenges in terms of stability and resource management. GO selection is a critical task in these networks, as the proper functioning of the network relies heavily on the optimal choice of this central node. The models developed in this work offer a smarter approach that is better adapted to varying network conditions, compared with conventional GO selection methods based on simple criteria such as RSSI or battery life. The classification model enables GO selection based on multiple dynamic characteristics, while the regression model provides a quantitative measure of a node's propensity to become a GO by calculating its Intent Value. This combination of the two models enables more robust and flexible decision making, adapted to constantly evolving network environments. The proposed election technique, as compared to other group formation introduces more overhead than the standard method and longer time for group formation, however it ensures that the most capable node is elected as group owner. Empirical tests will help confirm the applicability of these models in real-life scenarios, and assess their performance in the face of complex and dynamic and hoc networks.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Demir, U., Tapparello, C. and Heinzelman, W. (2017) Maintaining Connectivity in Ad Hoc Networks through WiFi Direct. 2017 *IEEE* 14*th International Conference on Mobile Ad Hoc and Sensor Systems* (*MASS*), Orlando, 22-25 October 2017, 308-312. https://doi.org/10.1109/mass.2017.60

[2] Seng, L.T., Gardner-Stephen, P., Mohamad Ali, N. and Sulaiman, R. (2024) Exploring the Role of Wi-Fi Direct's Service Discovery Protocol in Forming Wireless Collaboration Networks: A Scoping Review. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, **39**, 204-230. https://doi.org/10.37934/araset.39.2.204230

[3] Camps-Mur, D., Garcia-Saavedra, A. and Serrano, P. (2013) Device-to-Device Communications with Wi-Fi Direct: Overview and Experimentation. *IEEE Wireless Communications*, **20**, 96-104. https://doi.org/10.1109/mwc.2013.6549288

[4] Conti, M., Delmastro, F., Minutiello, G. and Paris, R. (2013) Experimenting Opportunistic Networks with WiFi Direct. 2013 *IFIP Wireless Days* (*WD*), Valencia, 13-15 November 2013, 1-6. https://doi.org/10.1109/wd.2013.6686501

[5] Botrel Menegato, U., Souza Cimino, L., Delabrida Silva, S.E., Medeiros Silva, F.A., Castro Lima, J. and Oliveira, R.A.R. (2014) Dynamic Clustering in WiFi Direct Technology. *Proceedings of the* 12*th ACM International Symposium on Mobility Management and Wireless Access*, Montreal, 21-26 September 2014, 25-29. https://doi.org/10.1145/2642668.2642682

[6] Cherif, W., Khan, M.A., Filali, F., Sharafeddine, S. and Dawy, Z. (2017) P2P Group Formation Enhancement for Opportunistic Networks with Wi-Fi Direct. 2017 *IEEE Wireless Communications and Networking Conference* (*WCNC*), San Francisco, 19-22 March 2017, 1-6. https://doi.org/10.1109/wcnc.2017.7925840

[7] Mbala, R.M., Nlong, J.M. and Kamdjoug, J.K. (2021) A Framework for Multi-Hop Ad-Hoc Networking over Wi-Fi Direct with Android Smart Devices. *Communications and Network*, **13**, 143-158. https://doi.org/10.4236/cn.2021.134011

[8] Hande, J.Y. and Sadiwala, R. (2024) Optimization of Energy Consumption and Routing in MANET Using Artificial Neural Network. *Journal of Integrated Science and Technology*, **12**, 718-718.

[9] Vikkurty, S. and Setty, P. (2022) Artificial Neural Network Based Optimized Link State Routing Protocol in MANET. *International Journal of Intelligent Engineering and Systems*, **15**, 65-73.

[10] Ramya, R. and Brindha, D.T. (2022) A Comprehensive Review on Optimal Cluster Head Selection in WSN-IoT. *Advances in Engineering Software*, **171**, Article ID: 103170. https://doi.org/10.1016/j.advengsoft.2022.103170

[11] Gangal, V., Sesli, E. and Hacioglu, G. (2024) ANN Based Cluster-Head Selection Process in WSN. https://assets-eu.researchsquare.com/files/rs-4109504/v1_covered_fc57ef14-2f24-48c0-b8f3-6c4fed113b84.pdf

[12] Senturk, A. (2025) Artificial Neural Networks-Based LEACH Algorithm for Fast and Efficient Cluster Head Selection in Wireless Sensor Networks. *International Journal of Communication Systems*, **38**, e6127. https://doi.org/10.1002/dac.6127

[13] Kovendan, A.K.P., Divya, R. and Sridharan, D. (2018) Dynamic Distance-Based Cluster Head Election for Maximizing Efficiency in Wireless Sensor Networks Using Artificial Neural Networks. In: *Recent Findings in Intelligent Computing Techniques*, Springer, 129-136. https://doi.org/10.1007/978-981-10-8636-6_14

[14] Gurumoorthy, S., Subhash, P., Pérez de Prado, R. and Wozniak, M. (2022) Optimal Cluster Head Selection in WSN with Convolutional Neural Network-Based Energy Level Prediction. *Sensors*, **22**, Article 9921. https://doi.org/10.3390/s22249921

[15] Khan, M.A., Cherif, W., Filali, F. and Hamila, R. (2017) Wi-Fi Direct Research—Current Status and Future Perspectives. *Journal of Network and Computer Applications*, **93**, 245-258. https://doi.org/10.1016/j.jnca.2017.06.004

[16] Khan, M.A., Hamila, R. and Hasna, M.O. (2019) Optimal Group Formation in Dense Wi-Fi Direct Networks for Content Distribution. *IEEE Access*, **7**, 161231-161245. https://doi.org/10.1109/access.2019.2951832

[17] Alliance, Wi-Fi (2016) Wi-Fi Peer-to-Peer (P2P) Technical Specification V1.7.

[18] Khan, M.A., Cherif, W. and Filali, F. (2016) Group Owner Election in Wi-Fi Direct. 2016 *IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, 20-22 October 2016, 1-9. https://doi.org/10.1109/uemcon.2016.7777908

[19] IEEE (2012) 802.11-2012—IEEE Standard for Information Technology-Telecommunications and Information Exchange between Systems LOCAL and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[20] IEEE (2011) 802.11u-IEEE Standard for Information Technology-Telecommunications and Information Exchange between Systems-Local and Metropolitan Networks-Specific Requirements-Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 9: Interworking with External Networks. IEEE Std. 802.11u-2011.

[21] Wi-Fi Alliance (2007) Wi-Fi Protected Setup Specification. Wi-Fi Alliance Std.

[22] Liu, K., Shen, W., Yin, B., Cao, X., Cai, L.X. and Cheng, Y. (2016) Development of Mobile Ad-Hoc Networks over Wi-Fi Direct with Off-The-Shelf Android Phones. 2016 *IEEE International Conference on Communications (ICC)*, Kuala Lumpur, 22-27 May 2016, 1-6. https://doi.org/10.1109/icc.2016.7511190

[23] Perkins, C., Belding-Royer, E. and Das, S. (2003) Ad Hoc On-Demand Distance Vector (AODV) Routing. No. RFC 3561.

[24] Shahin, A.A. and Younis, M. (2015) Efficient Multi-Group Formation and Communication Protocol for Wi-Fi Direct. 2015 *IEEE 40th Conference on Local Computer Networks (LCN)*, Clearwater Beach, 26-29 October 2015, 233-236. https://doi.org/10.1109/lcn.2015.7366314

[25] Chaki, P., Yasuda, M. and Fujita, N. (2015) Seamless Group Reformation in WiFi Peer to Peer Network Using Dormant Backend Links. 2015 12*th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, 9-12 January 2015, 773-778. https://doi.org/10.1109/ccnc.2015.7158075

[26] Jahed, K., Farhat, O., Al-Jurdi, G. and Sharafeddine, S. (2016) Optimized Group Owner Selection in WiFi Direct Networks. 2016 24*th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, 22-24 September 2016, 1-5. https://doi.org/10.1109/softcom.2016.7772169

[27] Zhang, H., Wang, Y. and Tan, C.C. (2014) WD2: An Improved WiFi-Direct Group Formation Protocol. *Proceedings of the 9th ACM MobiCom Workshop on Chal-*

lenged Networks, Maui, 7 September 2014, 55-60. https://doi.org/10.1145/2645672.2645674

[28] Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., *et al.* (2018) A Comprehensive Survey on Machine Learning for Networking: Evolution, Applications and Research Opportunities. *Journal of Internet Services and Applications*, **9**, Article No. 16. https://doi.org/10.1186/s13174-018-0087-2

[29] Gifford, J. (2023) Smartphone Processors Ranking & Scores. https://www.kaggle.com/datasets/alanjo/smartphone-processors-ranking

[30] Jo, A. (2023) Battery Charging Data. https://www.kaggle.com/datasets/josephgifford/battery-charge-feb-2022?resource=download

[31] (2017) UCI KDD Archive. https://kdd.ics.uci.edu/

[32] WAND Network Research Group (2017) WITS: Waikato Internet Traffic Storage. https://wand.net.nz/wits

[33] Abbas, S., Alenazi, M.J.F. and Samha, A. (2022) Mobility Prediction of Mobile Wireless Nodes. *Applied Sciences*, **12**, Article 13041. https://doi.org/10.3390/app122413041

[34] Henderi, H. (2021) Comparison of Min-Max Normalization and Z-Score Normalization in the K-Nearest Neighbor (KNN) Algorithm to Test the Accuracy of Types of Breast Cancer. *IJIIS*: *International Journal of Informatics and Information Systems*, **4**, 13-20. https://doi.org/10.47738/ijiis.v4i1.73

[35] Hara, K., Saito, D. and Shouno, H. (2015) Analysis of Function of Rectified Linear Unit Used in Deep Learning. 2015 *International Joint Conference on Neural Networks* (*IJCNN*), Killarney, 12-17 July 2015, 1-8. https://doi.org/10.1109/ijcnn.2015.7280578

[36] Ruby, U. and Yendapalli, V. (2020) Binary Cross Entropy with Deep Learning Technique for Image Classification. *International Journal of Advanced Trends in Computer Science and Engineering*, **9**, 5393-5397. https://doi.org/10.30534/ijatcse/2020/175942020