

Article

RSSGM: Recurrent Self-Similar Gauss–Markov Mobility Model

Mohammed J. F. Alenazi * , Shatha O. Abbas, Saleh Almowuena and Maazen Alsabaan

Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; 439204371@student.ksu.edu.sa (S.O.A.); saleh@ksu.edu.sa (S.A.); malsabaan@ksu.edu.sa (M.A.)

* Correspondence: mjalenazi@ksu.edu.sa; Tel.: +966-11-4696259

Received: 6 October 2020; Accepted: 3 December 2020; Published: 7 December 2020



Abstract: Understanding node mobility is critical for the proper simulation of mobile devices in a wireless network. However, current mobility models often do not reflect the realistic movements of users within their environments. They also do not provide the freedom to adjust their degrees of randomness or adequately mimic human movements by injecting possible crossing points and adding recurrent patterns. In this paper, we propose the recurrent self-similar Gauss–Markov mobility (RSSGM) model, a novel mobility model that is suitable for applications in which nodes exhibit recurrent visits to selected locations with semi-similar routes. Examples of such applications include daily human routines, airplane and public transportation routes, and intra-campus student walks. First, we present the proposed algorithm and its assumptions, and then we study its behavior in different scenarios. The study’s results show that different and more realistic mobility traces can be achieved without the need for complex computational models or existing GPS records. Our model can flexibly adjust its behavior to fit any application by carefully tuning and choosing the right values for its parameters.

Keywords: mobility model; wireless networking; Gauss Markov; mobile network; human mobility

1. Introduction and Motivation

Recently, wireless networks have been widely deployed, as they offer high-speed data rates and have become more reliable and resilient against noise and interference. Globally, the total amount of mobile data traffic was around 1.1 Exabytes per day by the end of 2019, and this is expected to increase five times (i.e., 5.5 Exabytes per day) by 2025 [1]. Such popularity has led to continuous improvement and development in every aspect of wireless networking, including its infrastructures, architectures, protocols, and services. However, researchers and developers are still facing significant challenges in the evaluation phase of mobile device simulation algorithms and technologies, since testing them in existing and deployed wireless networks requires a high financial cost and is a time-consuming process. For that reason, several studies have been conducted to offer appropriate environments to facilitate the simulation and evaluation of novel and enhanced network systems. Some of these studies have focused on modeling the mobility of users within the wireless network to predict their movements [2]. Mimicking mobile users, such as pedestrians or drivers, with realistic constraints, will help in appropriately performing handover operations, making scheduling decisions, and obtaining the density of user distributions within their cells. In fact, the understanding of mobility patterns was motivated by its relevance to different real-life applications, such as location recommendation [3], motion estimation and positioning [4,5], safe mobility during crisis [6], and forecasting disturbances in electromobility [7].

Various mobility models have been introduced in the literature, and these models rely on different assumptions for user movement trajectories. For example, both random waypoint (RWP) and random walk (RW) mobility models predict the speed and direction of an object within a specific zone using random processes [8]. Random mobility models are straightforward and simple to apply. However, they are unrealistic and only work when evaluating and simulating algorithms that are entirely independent of user mobility. To improve the predictions of users' movements in wireless networks, the Gauss–Markov (GM) mobility model was proposed to take advantage of the temporal dependencies feature, in addition to random properties [9]. The GM model yields better performance results than other random mobility models, but it is still far from providing realistic patterns. This has caused many researchers to apply complex, application-driven mobility models that are based on the GPS traces of human walks or cellphone location tracking [10–12]. Here, we propose to improve the performance of GM and provide more realistic mobility modeling without introducing additional computational complexity or requiring any traces of human mobility.

It is challenging to predict user movement patterns in a wireless network [13], yet mobile users, despite their random behaviors, sometimes pass particular points due to predetermined and realistic constraints. The number of these points and the amount of randomness in users' behaviors are dependent on scenarios. For instance, the participants in a marathon must run within a given route, so they are likely to cross a high number of known milestones, thereby making their movement patterns predictable. Similarly, the visitors at a shopping mall go through specific walkways and entrances, but with extra degrees of randomness in their behaviors. This uncertainty of human mobility becomes higher in more open places, such as beaches and parks. Most existing mobility models do not provide the flexibility of adjusting their degree of randomness or the option of mimicking human movements by injecting possible crossing points and adding recurrent patterns.

In this paper, we introduce a new mobility model, named the “recurrent self-similar Gauss–Markov mobility (RSSGM)” model that mimics those applications exhibiting recurrent visits to selected locations with self-similar routes. The proposed model captures the random behavior of the GM model. It also uses recurrent user travel patterns of the chosen nodes and self-similarity movement that imitates real human mobility. We will present the proposed algorithm and its assumptions; then we will evaluate its performance under different scenarios. A comparison with other mobility models (e.g., GM, RWP, and random direction (RD)) will also be discussed. We will show that our mobility model provides an enhanced and more realistic prediction of human mobility by applying recurrent and self-similarity features. As a result, the simulation and evaluation of wireless and cellular networks will be improved.

The remainder of this paper is organized as follows. Brief theoretical background of mobility models and their categories is presented in Section 2. In Section 3, related works in the literature are summarized. Our proposed RSSGM model is introduced in Section 4. In Section 5, the obtained evaluation performance results are presented and discussed. Finally, concluding remarks and future research directions are provided in Section 6.

2. Mobility Models in Wireless Networks

Recently, increasingly more research attempts have been made to produce mobility models that can adequately mimic users' movements and behaviors. Hence, it is becoming necessary to provide a more realistic evaluation of new algorithms and protocols. Various mobility models have been proposed to predict the current position, direction, and speed of a mobile terminal at any time instance in the simulation phase. These mobility models differ in their predefined assumptions, the input distributions they use, and the ways they generate route traces [14,15]. As shown in Table 1, these mobility models can be classified based on their characteristics into five categories [16,17]:

- Random mobility models: The movement of nodes in this category is unrestricted, and all assumptions (e.g., speed and direction) are chosen randomly and independently of other mobile users. RWP and RW are examples of such random mobility models.

- Temporal dependency models: The current speed and direction of a moving object in the temporally-dependent models are impacted by their previous values at earlier time instances. The GM mobility model is an example of this category.
- Spatial dependency models: The movement of a node is affected by its adjacent nodes, and the speeds and directions of mobile nodes are associated with each other. The reference point group mobility model is classified as being in this category.
- Mobility with geographic restrictions models: There are geographic restrictions, trajectories, and environment deterrents that must be taken into account in this type of mobility model. The Manhattan grid mobility model falls under this category.
- Hybrid mobility models: If a mobility model has features of at least two of the previous four categories, it will be considered a hybrid model. For instance, the freeway mobility model in [18] belongs to this category.

Table 1. Categories of mobility models in wireless networks.

Categories	Examples
Random	RWP RD RW
Temporal Dependency	GM
Spatial Dependency	Reference Point Group
Geographic Dependency	Manhattan Grid
Hybrid	Freeway

Since our proposed mobility model shares some aspects of both the random and the GM mobility models, we will elaborate more on the first two categories in this background section. Examples and more discussion of other classes will be presented in the related work, in Section 3.

2.1. Random Waypoint Mobility Model

The random waypoint (RWP) model is a simple approach in which mobile terminals are randomly distributed within an area of interest. The movement of mobile terminals is independent of each other and there are no fixed routes or restricted boundaries of their movements. A node, such as a pedestrian, vehicular object, or IoT device, starts by selecting a random destination and then moving with speed and direction that are determined randomly from the intervals $[Speed_{min}, Speed_{max}]$ and $[0, 2\pi]$, respectively. When the node reaches the targeted destination, it pauses for a specific time randomly picked from the interval $[0, t_{max}]$. This process is repeated until the simulation phase is terminated. The speed, direction, and pausing time are the key parameters of the mobility behavior of nodes in this approach [8,19].

2.2. Random Direction Mobility Model

The random direction (RD) mobility model is almost identical to the random waypoint approach, as mobile users in both models travel in piece-wise linear segments. However, they differ from each other in the way the next direction is selected and how a user is handled once it hits a boundary. When a mobile terminal reaches the border of the predefined area, it reflects or wraps toward the simulation region with updated values for its speed and directions [20,21]. The distributions for both speed and direction are also much simpler in the random direction approach, making it easier to implement but less physically appealing [22]. Figure 1 illustrates an example of a user movement following the random direction mobility model.

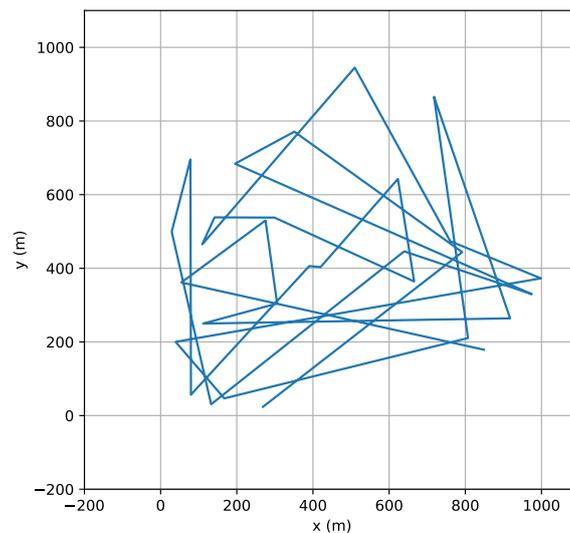


Figure 1. An example of a node's movement using the random direction mobility model.

2.3. Random Walk Mobility Model

Unlike the RWP model, the RW mobility approach tries to imitate human movements in urban streets and walkways, where users tend to move either toward the east-west or north-south directions in concurrent and close steps. The RW mobility model mimics such characteristics and provides more realistic movement instances than other random mobility models [23]. To do that, mobile nodes in the RW approach move from their current positions to the next step by selecting a random speed within a pre-defined range $[Speed_{min}, Speed_{max}]$ and random direction from the interval $[0, 2\pi]$, but every step occurs within a constant time (t) or fixed distance (d). The process of choosing the next direction and speed is repeated in the same way until the simulation process is completed [24,25]. Figure 2 shows an example of the movement of a node following the RW mobility model.

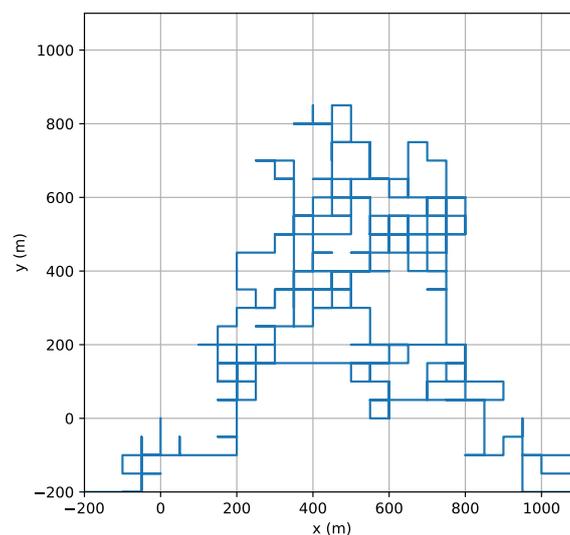


Figure 2. An example of a node's movement using the RW mobility model.

2.4. Gauss–Markov Mobility Model

The Gauss–Markov (GM) mobility model aims at improving previous approaches by exploiting temporal dependency. Here, the speed and direction of a mobile terminal are updated according to their past values at earlier time intervals. Additionally, the calculation of these two values is

done with a certain degree of randomness that can be tuned based on the nature of the simulated wireless network. The GM mobility model is not stateless, as the memories of previous steps are preserved [26,27]. However, the node movement is still independent of other mobile terminals within the same network.

Mobile nodes are allocated in random locations within the wireless network. These nodes will set their speed (S_t) and direction (D_t) for each specific time interval (t). The speed and direction of a node are calculated as follows [28]:

$$S_t = \alpha S_{t-1} + (1 - \alpha)\bar{S} + \sqrt{(1 - \alpha^2)S_{t-1}^G} \quad (1)$$

$$D_t = \alpha D_{t-1} + (1 - \alpha)\bar{D} + \sqrt{(1 - \alpha^2)D_{t-1}^G} \quad (2)$$

where \bar{S} denotes the mean speed and \bar{D} denotes the mean direction. The α parameter is utilized to inject a degree of randomness in the calculation of both speed and direction. The value of α is between 0 and 1, and it has a negative correlation with the degree of randomness; when α increases, the uncertainty of movement will be lower. If α is set close to zero, user mobility tends to follow highly random behavior. Once α is adjusted to be very close to one, the speeds and directions during previous time intervals will have more significant impacts in selecting future values, thereby allowing the mobility model to be more temporally dependent.

At every instance, the next position (X_t, Y_t) of the mobile node depends on the current location (X_{t-1}, Y_{t-1}), speed (S_{t-1}), and direction (D_{t-1}) of the last movement, as shown in the following two equations [28]:

$$X_t = X_{t-1} + S_{t-1} \cos D_{t-1} \quad (3)$$

$$Y_t = Y_{t-1} + S_{t-1} \sin D_{t-1} \quad (4)$$

Figure 3 shows an example of node movement in the GM mobility model with different values of α .

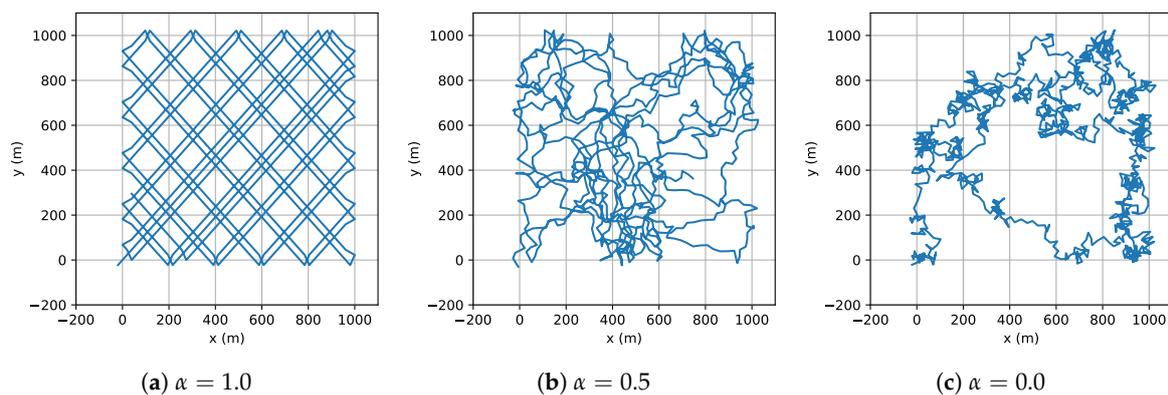


Figure 3. An example of user movement in the GM mobility model at different values of the tuning parameter α .

3. Related Works

There have been several efforts to improve the performance of mobility models in the literature. For instance, He et al. in [29] introduce a modified version of the GM mobility model, in which tuning parameters are utilized to set the degree of randomness in user movements. They also integrate the proposed mobility model with the cluster-based two-ring Machine-to-Machine (M2M) reference model to recognize non-stationary channels. The model is capable of covering other mobility trajectories (including RW mobility model and linear motion).

Wang et al. in [30] present a realistic mobility model that is suitable for environments with irregular barriers and constrained mobility. Their approach is based on Bezier curves to distinguish some control points and to provide smooth user movements between blocks and pathways. The simulation results obtained show that the proposed algorithm outperforms its counterpart mobility models in restrained environments with respect to the smoothness of its paths, the avoidance of collisions, and the capability of broad movements.

Several other studies have proposed mobility models for Unmanned Aerial Vehicles (UAVs) [31–35]. Li et al. [31] took advantage of the temporal dependency of an Unmanned Aerial Vehicle (UAV). They also considered the spatial correlation among a group of UAVs, since these vehicles are usually traveling in an arranged formation. Both temporal dependency and spatial correlation are exploited to propose a particle swarm mobility model for flying ad hoc networks. Sharma and Kim in [32] introduced a mixed mobility model that applies both RWP and uniform mobility models in order to examine the coverage probability of mobile terminals in a three-dimensional network of UAVs. RWP is utilized for vertical motions, whereas the uniform mobility model is followed to choose spatial directions. The authors in [32] found that such a mixed mobility model helps to associate mobile terminals with the closest UAV, thereby enhancing the network's coverage performance.

Jo et al. in [33] proposed a customized mobility model for reconnaissance by dividing the scanning area into zones. UAVs in this model would communicate with each other to record the number of visits in each zone, and then the next movement of a UAV would lean toward the zone with the least number of visits. The same authors extended their work to improve the coverage rate in reconnaissance operations by limiting the overlapping among UAVs and avoiding any simultaneous visits to the same zone [34]. Sayeed et al. in [35] exploited the transmission density of nodes in a wireless sensor network to give a mobility model in which the UAV traversals are determined according to dynamic changes in the underlying network topology.

The performance of mobility models in mobile ad hoc networks (MANETs) has also been investigated [28,36–38]. For instance, Abdullah et al. [37] evaluated the RWP model on various routing protocols: ad hoc on-demand distance vector (AODV), optimized link state routing (OLSR), and Geographic Routing Protocol (GRP). The evaluation was conducted under four different configurations for speed and pausing time: fast car, slow car, race walk, and human walk models. As a result, the OLSR protocol has the best performance among the three routing protocols, as it yields short service delay, low packet retransmission, and better throughput. Appiah [36] analyzed two types of mobility models: RWP and MANET Down Left (MDL), and compared their performance when the OLSR is implemented in MANET. He showed that choosing a suitable speed and pausing time for the RWP mobility model could lead to better performance than MDL. In another performance study, Laqtib et al. [28] investigated the performance of the OLSR under four mobility models: RWP, RD, RW, and GM. The study concluded that the RWP model gives optimal performance results with respect to delay, packet retransmission, and throughput, regardless of the number of mobile nodes in the network.

Lee et al. [39] proposed the self-similar least-action human walk (SLAW) model to capture the statistical characteristics of human movements in highly populated environments, such as university campuses and public places. People in these environments act in similar ways, making the prediction of their movements more manageable.

To mimic human mobility, most related works focus on providing mobility models that are either purely random, rely on real-world traces, or are built based on complex statistical models. Other techniques are also not fully decentralized, since a form of coordination among users is required to determine their movements. In this study, RSSGM provides a random mobility model for network nodes, while exhibiting self-similarity and frequent movements based on user requirements.

4. Recurrent Self-Similar Gauss–Markov Mobility (RSSGM)

The RSSGM model is proposed for applications when nodes exhibit recurrent visits to selected locations with semi-similar routes. For example, a professor at a university might have recurrent visits to a classroom, a laboratory, his/her office, and a coffee shop. In addition, the model provides the flexibility of adjusting the degree of randomness of nodes' mobility. For instance, the mobility of a professor whose movement patterns are predictable might have less randomness than a student who hangs out a lot with friends. Moreover, the model provides the option of mimicking human movements by injecting possible crossing points and adding recurrent patterns.

This section presents and discusses the algorithm and implementation of the proposed RSSGM mobility model, whose pseudo-code is shown in Algorithm 1. The algorithm defines all required functions, such as the *nextCentroid* function that obtains the next centroid point, and a function named "dist" that is used to calculate the Euclidean distance between two points. In addition, all required parameters are defined; for example, speed mean, direction mean, a tuning parameter, a list of centroid points, a distance where the node enters the range of the current centroid, Mobility boundary limits, and the required number of points of a node. The output of the algorithm is to have the mobility traces.

The user provides a set of centroid locations, $centroid_x$, and $centroid_y$, that are used to determine the locations nodes can visit. In other words, a moving node moves towards the first centroid's location, and once it gets in the range of C_{dist} of the centroid location, it starts to move to the next centroid location.

The algorithm consists of two main phases: an initialization phase (lines 2–4) and a mobility generation phase (lines 5–32). The first phase starts by initializing all the variables needed for the mobility model. It also obtains the centroid location variables, $centroid_x$ and $centroid_y$, using the function *nextCentroid* function, which retrieves the first element of the given centroids list *centroidList*. Next, the algorithm initializes the location coordinate, x, y , such that the node begins its movement, starting from the first centroid location in the given list.

The second phase of the algorithm aims at generating the mobility patterns of a node, and it consists of three sub-phases: centroid selection, next movement decision, and boundary preservation. In the centroid selection sub-phase, if the current point (x, y) is located near the currently targeted centroid, the centroid coordinate will be updated with the next point on the list of given centroids via the function *nextCentroid*, which returns the following centroid location. On the other hand, if the current point (x, y) is still away from the targeted centroid, the current centroid remains unchanged.

The next movement decision is accomplished in Phase II-b, and is computed as follows. First, the mean direction \bar{D} is updated to a point towards the targeted centroid location, which is calculated using Equation (5):

$$\bar{D} = \arctan \frac{centroid_y}{centroid_x} \quad (5)$$

This equation improves the mobility model compared to other existing approaches, including the GM model, by forcing nodes to come across specific centroid points. Such enforcement is realistic since, although mobile users behave randomly, they sometimes follow particular points because of predetermined constraints in their environments. The next step in the future movement decision sub-phase is updating the two random function variables, S_{t-1}^G and D_{t-1}^G , utilizing the normal distribution function. Then, the current speed and direction are also updated using the two GM Mobility functions shown in lines 15 and 16 of the pseudo-code in Algorithm 1. Using the updated speed and direction, the next location, (x_t, y_t) , is determined using lines 17 and 18.

Finally, our algorithm performs the boundary preservation sub-phase to ensure keeping the mobility pattern within the environment boundary, which is defined by a polygon $[X_{min}, X_{max}, Y_{min}, Y_{max}]$. If the node moves outside the polygon, the mean direction is updated to bounce the node back inside the polygon, using the equations shown in lines 21 and 26 of Algorithm 1. The proposed RSSGM model will continue iterating through Phase II until the number of required points is added to the mobility traces list, which is then returned to the user.

Algorithm 1: Recurrent self-similar gauss–markov mobility model.**Functions:**

$\mathcal{N}(\mu, \sigma^2)$: = a normal distribution function, where μ is the mean and σ^2 is the variance.

nextCentroid(List): = returns the next centroid point in round-robin fashion.

dist($(x_1, y_1), (x_2, y_2)$): = returns the Euclidean distance between two points (x_1, y_1) and (x_2, y_2) .

Input:

\bar{S} : = speed mean;

\bar{D} : = direction mean;

α : = a tuning parameter;

CentroidList: = a list of centroid points;

C_{dist} : = a distance where the node enters the range of the current centroid;

$X_{min}, X_{max}, Y_{min}, Y_{max}$: = Mobility boundary limits;

RequiredPoints: =the required number of points of a node;

Output:

mobilityTraces(x, y): = a list of mobility traces;

```

1 begin
2   Phase I: Initialization:
3   centroidx, centroidy = nextCentroid(CentroidList)
4   xt, yt = centroidx, centroidy
5   Phase II: Mobility Generation:
6   while mobilityTraces.length ≤ RequiredPoints do
7     Phase II-a: Centroid Selection:
8     if dist((xt, yt), (centroidx, centroidy)) < Cdist then
9       | centroidx, centroidy = nextCentroid(CentroidList)
10    end
11    Phase II-b: Next Movement:
12     $\bar{D} = \arctan \frac{\text{centroid}_y}{\text{centroid}_x}$ 
13     $S_{t-1}^G = \mathcal{N}(0, \bar{S})$ 
14     $D_{t-1}^G = \mathcal{N}(0, 2\pi)$ 
15     $S_t = \alpha S_{t-1} + (1 - \alpha)\bar{S} + \sqrt{(1 - \alpha^2)S_{t-1}^G}$ 
16     $D_t = \alpha D_{t-1} + (1 - \alpha)\bar{D} + \sqrt{(1 - \alpha^2)D_{t-1}^G}$ 
17    xt = xt-1 + St cos Dt
18    yt = yt-1 + St sin Dt
19    Phase II-c: Keep Inside Boundary:
20    if xt > Xmin or xt < Xmax then
21      | xt = xt - St * cos(Dt)
22      |  $\bar{D} = \pi - \bar{D}$ 
23      | Dt =  $\bar{D}$ 
24    end
25    if yt > Ymin or yt < Ymax then
26      | yt = yt - St * sin(Dt)
27      |  $\bar{D} = -\bar{D}$ 
28      | Dt =  $\bar{D}$ 
29    end
30    mobilityTraces.append(xt, yt)
31  end
32  return mobilityTraces
33 end

```

5. Experimental Evaluation

The proposed mobility model and other baseline mobility models (GM, RWP, and random direction) are implemented using Python. The NetworkX library is used for the graph and nodes [40]. The Shapely library is used for geometric measurements [41]. The experimental evaluation is conducted based on an area of [1000,1000] with different values of $\alpha = [0, 0.1, 0.5, 0.9, 0.99, 1.0]$ to study its behavior and movement patterns. In addition, we run experiments with different values of mean speed, $\bar{S} = [10, 50, 100]$, to study the impact of changing the mean speed on the behavior of nodes' mobility. During the evaluation, the number of centroids is set to be five with the locations [(150, 150), (500, 500), (800, 800), (150, 800), (150, 800)]. All parameters used in the simulation are shown in Table 2, and the results of our evaluation are shown in Figure 4.

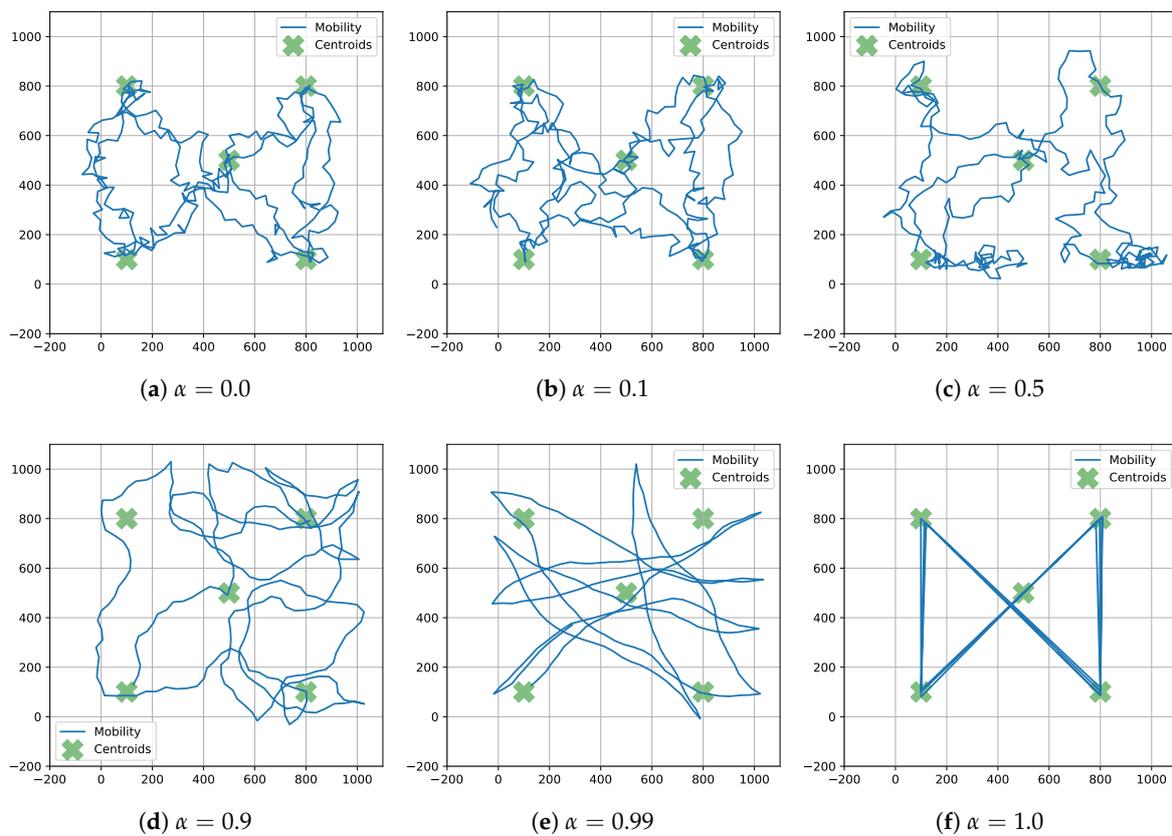


Figure 4. Varying α values for recurrent self-similar Gauss–Markov mobility.

Table 2. Simulation parameters.

Parameter	Value
\bar{S}	40 m/s
\bar{D}	0°
α	[0, 0.1, 0.5, 0.9, 0.99, 1.0]
CentroidList	[(150, 150), (500, 500), (800, 800), (150, 800), (150, 800)]
C_{dist}	20 m
$X_{min}, X_{max}, Y_{min}, Y_{max}$	[0, 1000, 0, 1000]
RequiredPoints	10,000

5.1. Effect of Varying α on Mobility

The effect of varying α is that the random movement of the mobile nodes in terms of speed and direction changes based on Equations (1) and (2), respectively. The speed equation is invoked at line 15 in Algorithm 1, while the direction equation is invoked in line 16 of the same algorithm. The changes in the speed value are not reflected in the movements in Figure 4. Thus, we will only focus on the direction equation. When $\alpha = 0.0$, the direction equation is as follows:

$$D_t = \bar{D} + \sqrt{D_{t-1}^G} \quad (6)$$

As a result, the movement is entirely affected by the sum of the median direction \bar{D} and the randomness term $\sqrt{D_{t-1}^G}$. The median direction always points at the next centroid position, which is computed using Equation (5), whereas the random value is drawn from a normal distribution with a mean of 0. In the case when the drawn random value is zero at any given time, the node moves purely towards the centroid position. However, when the random value is not zero, the resulting direction is toward the centroid's position. As a consequence, the node could move in all directions locally, but it moves globally as it gets closer towards the next centroid, as shown in Figure 4a. We clearly see that the node starts at point (150, 150), and it moves towards the point (500, 500), while making turns in all directions. Then, the node gets closer to the center point by a distance of 20, and then it continues to the next centroid. This process is repeated by selecting the next centroids in a round-robin fashion until 10,000 points are generated.

We observe that there is no significant change in the node movements between when $\alpha = 0.0$ and $\alpha = 0.1$, as shown in Figure 4a,b. For $\alpha = 0.1$, the effect of the randomness dominates as the term $1 - \alpha$ increases slightly to 0.9. In fact, we can clearly see that the node starts to move in all directions randomly. However, the node eventually reaches the centroid's position, since the direction is the sum of the random value and the median direction. Therefore, the node moves mostly in a random way, as shown in its movement, but it purely points toward the centroid location. When the random value is a non-zero term, the resulting direction will be toward the centroid's position and deviated by adding the amount of uncertainty. The node's mobile movement will also be entirely affected by the random term, as shown in Figure 4b.

For $\alpha = 0.5$, the effect of the randomness is still clearly observed, and the node movement is equally affected by both the mean direction and the random term. Therefore, the node moves mostly in a random way, as shown in Figure 4c. Although the value of $\alpha = 0.5$ implies that the randomness should be about %50 between straight lines and random movements, we still see more patterns in U-turn shapes during the node movements. Such patterns are triggered by the random behavior of the GM mobility approach. This behavior also indicates that the effect of α value is not uniform, and the α should be higher than 0.5 to move in a more consistent way.

Once α is tuned to 0.9, less randomness is noticed in the movement of nodes, and the node is profoundly affected by the mean direction in its mobility. Still, since the randomness value, $1 - \alpha$, is 0.1, the node moves slightly randomly while targeting the next centroid location, as shown in Figure 4d. We also notice that setting the tuning parameter $\alpha = 0.9$ in the proposed mobility model yields an interesting movement pattern that mimics human mobility in open areas.

For $\alpha = 0.99$, the effect of the randomness starts to vanish as the term $1 - \alpha$ decreases to almost 0.01. The movement is profoundly affected by the mean direction. Therefore, the node moves in a semi-straight line, as shown in Figure 4e. We observe that the node visits all centroids according to their order in the given list. However, the main challenge with $\alpha = 0.99$ is that a node might need several steps to get closer to the centroid position. This is because the node mainly moves with semi-straight lines, which in turn deviate it from the targeted centroid. As a result, it does not get closer to the centroid and might hit the boundary, as shown in Figure 4e.

When $\alpha = 1.0$, the amount of uncertainty in our mobility model is zero, since both Equations (1) and (2) have only one term. Hence, the movement here is purely motivated by the mean

direction, which lets the mobile node move in straight lines, as shown in Figure 4f. As seen in the figure, the node starts from the position of the first centroid at (150, 150). Then, it moves toward the second centroid located at (500, 500). The node keeps visiting all five centroids in a round-robin fashion.

After evaluating the proposed mobility model with several values of the tuning parameter α , we conclude that selecting the appropriate value of α is crucial to mimic the intended movements within the chosen application. For example, with $\alpha = 1$ or $\alpha = 0.99$, nodes tend to move in straight lines among the given centroid points. Such behavior emulates airplane movements between a set of airport locations. With $\alpha = 0.9$ or $\alpha = 0.5$, nodes tend to mimic human-like walks in open areas, such as parks and sports fields. Moreover, with α values closer to zero, nodes tend to move randomly, turning to all directions while reaching the next centroid eventually. Such movement can mimic delivery robots when they deliver packages to stores, or ants when they search for food.

5.2. Effect of Varying Mean Speed \bar{S} on Mobility

This section studies the effect of varying mean speed \bar{S} on the RSSGM mobility model. We run three scenarios with different values of $\bar{S} = [10, 50, 100] \text{ m/s}$ as shown in Figure 5. We start our study by setting the mean speed to a low-speed value of 10 m/s. The results of RSSGM mobility with $\bar{S} = 10$ are shown in Figure 5a. We visually observe that the node moves with inconsiderable variations in terms of direction. The mean speed keeps the instantaneous speed low on average. Since the speed is low, we observe that the node does not get far away from the mean direction. Next, we increase the mean speed to 50 m/s to study the effect of a medium speed. In this scenario, we see that the node starts to move far away from the mean direction, as shown in Figure 5b. Finally, the mean speed is set to 100 m/s to study the effect of such a high speed. The results of RSSGM mobility with $\bar{S} = 100$ are shown in Figure 5c. With high speed, we observe that the node cannot reach the targeted centroid easily. It misses the centroid's center several times before moving toward the next centroid. This is why the node hits the boundaries many times, as shown in Figure 5c. On the other hand, with low speed, $\bar{S} = 10$, the node clearly crosses the center of the centroid on the first try as shown in Figure 5a. In this figure, we see that the node never hits the boundaries.

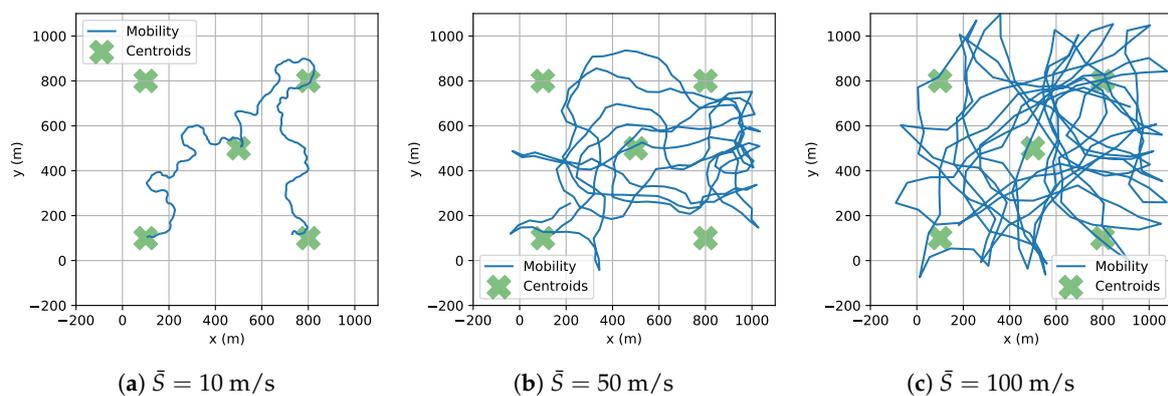


Figure 5. Varying α values for RSSGM Mobility.

5.3. Results and Discussions

This section compares the proposed mobility model to baseline models, such as GM, RWP, and random direction. The evaluation focuses on spatial and temporal randomness and self-similarity. For evaluation purposes, we define a new metric called spatial and temporal standard deviation (STSD), which measures the level of change in mobility with different random seeds. The STSD is defined as follows:

$$STSD(T) = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (P_i - \bar{P})^2} \quad (7)$$

where $T = \{P_1, P_2, P_3, \dots, P_N\}$ represents the set of mobility traces and N represents the number of sample traces. $P_i = \{p_1, p_2, p_3, \dots, p_t, \dots, p_n\}$ represents the set of points for a given mobility trace, where p_t represents the point at time t . STSD metric determines the standard deviation of multiple samples of a given mobility model. Hence, if all samples are identical, STSD is zero, which indicates that there is no deviation for the given set of samples. However, STSD value increases as the sample mobility traces follow different routes. The larger the distance is for a particular time, the higher STSD value is indicated.

5.3.1. Impact of α on Spatial and Temporal

This section studies the impact of changing the value of α on the spatial and temporal standard deviation (STSD) for RSSGM and GM mobility models. RWP and random direction are not included in this comparison, since they do not depend on the value of α . For this comparison, the value of α is increased from 0 to 1 incrementally to study how it affects STSD. The evaluation results are shown in Figure 6. We observe that the STSD value starts high for both mobility models when α is 0. However, RSSGM provides significantly lower, almost 40%, STSD values than GM. This indicates that RSSGM exhibits similar routes with different random seeds. Moreover, as we increase α to 0.9, the STSD provides semi-uniform values for both models. When α gets closer to 1, the STSD values increase slightly (by 20%) for GM, which indicates a slight deviation, because nodes tend to move in semi-straight lines but in different directions with every random seed. As a result, the STSD value increases as these lines travel in a different direction every time. For RSSGM, this phenomenon is seen more clearly. In fact, when α gets closer to 1, the STSD values increase for RSSGM by 100%. Finally, when α reaches 1, we observe that the STSD value drops to zero for both mobility models since the movements of the nodes are identical for all random seeds.

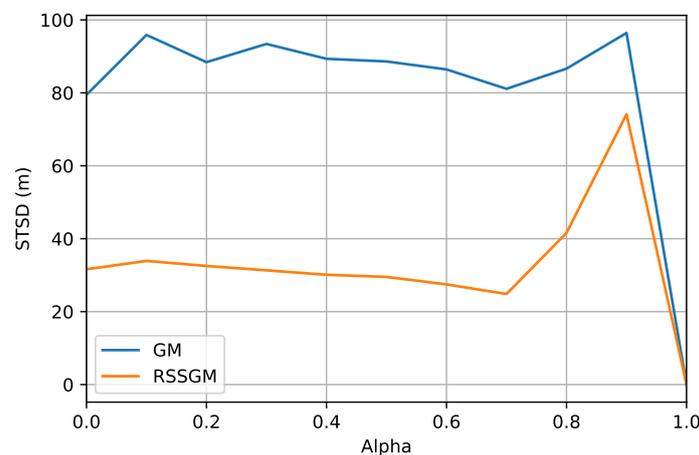


Figure 6. Impact of α on spatial and temporal standard deviation (STSD).

5.3.2. Comparative Evaluation

Based on STSD, this section compares RSSGM, GM, RWP, and random direction (RD) mobility models. The comparative evaluation results are shown in Figure 7. We observe that the RSSGM provides the lowest STSD values, with an average of around 30 m standard deviation. This indicates that RSSGM provides the most similar routes with different random seeds. Next, GM shows the second-lowest variation in terms of mobility, with an STSD value of around 80 m. Moreover, we observe that RD shows the third-lowest variation in terms of average mobility, with an STSD value of

around 270 m. However, RD provides high variance and range in terms of variation, starting from 0 to up to 410 m. This indicates that RD is highly random in terms of mobility with different seeds. Finally, RW provides the third-highest variation in terms of average mobility, with an STSD value of around 280 m. By observing all STSD values, we conclude that our RSSGM showed the best similarity when running different random seeds. This similarity is preferable when simulating some applications, such as human walk, which exhibit similar routes to routinely visited places at homes or workplaces.

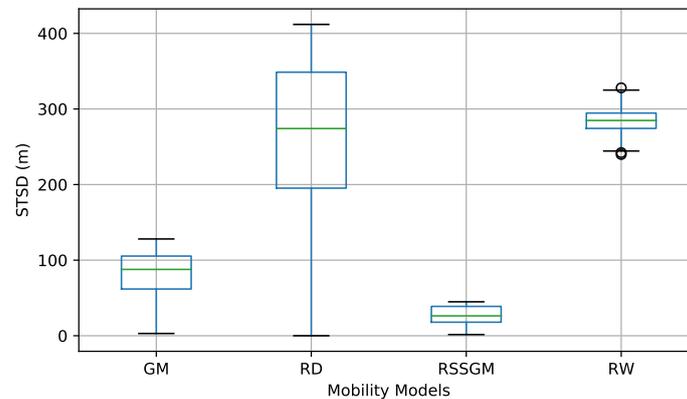


Figure 7. STSD values with different mobility models.

6. Conclusions and Future Work

In this paper, we presented a novel mobility model, based on the GM model, to consider such movements in a predefined set of places, named centroids. We studied the behavior of our proposed mobility approach in several scenarios to evaluate the impacts of changing its parameters, such as α and mean speed \bar{v} . Our evaluation results showed that different mobility behaviors could be achieved by carefully selecting the right parameters. The most dominant parameter is α , which controls the randomness in our mobility. With α values closer to 1, nodes move in semi-straight lines and STSD reaches zero, whereas these nodes tend to move in more random patterns (high STSD) when α is closer to 0. Moreover, with relatively low speed, nodes tend to cross the center of each centroid on the first attempt. However, with high speed, nodes' mobility is significantly affected by the randomness of speed and direction, which in turn yields high randomness in displacement. As a result, nodes cannot reach the targeted centroid easily; they miss the centroid's center several times before moving toward the next centroid. In addition, compared to GM, RW, and RD, the study shows that RSSGM achieves the best similarity when running different seeds. For future work, we plan to build a 3D version of our mobility model to emulate airplane and drone routes. In addition, we plan to compare our mobility model to real traces, such as human walking traces and airplane routes. Moreover, we plan to implement our model in an ns-3 network simulator to study its effect in data traffic scenarios.

Author Contributions: All authors have contributed in all parts of this work. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group number RG-1441-512.

Acknowledgments: The authors thank the Deanship of Scientific Research and RSSU at King Saud University for their technical support.

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

References

- Ericsson Mobility Report. June 2020. Available online: <http://tiny.cc/Ericsson20> (access on 12 August 2020).
- Wang, H.; Yang, Z.; Shi, Y. Next location prediction based on an Adaboost-Markov model of mobile users. *Sensors* **2019**, *19*, 1475. [[CrossRef](#)] [[PubMed](#)]

3. Li, W.; Liu, X.; Yan, C.; Ding, G.; Sun, Y.; Zhang, J. STS: Spatial–Temporal–Semantic Personalized Location Recommendation. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 538. [[CrossRef](#)]
4. Luo, A.; Chen, S.; Xu, B. Enhanced map-matching algorithm with a hidden Markov model for mobile phone positioning. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 327. [[CrossRef](#)]
5. Pham, T.T.; Suh, Y.S. Spline function simulation data generation for walking motion using foot-mounted inertial sensors. *Electronics* **2019**, *8*, 18. [[CrossRef](#)]
6. Moslem, S.; Campisi, T.; Szmelter-Jarosz, A.; Duleba, S.; Nahiduzzaman, K.M.; Tesoriere, G. Best–Worst Method for Modelling Mobility Choice after COVID-19: Evidence from Italy. *Sustainability* **2020**, *12*, 6824. [[CrossRef](#)]
7. Müller, M.; Biedenbach, F.; Reinhard, J. Development of an Integrated Simulation Model for Load and Mobility Profiles of Private Households. *Energies* **2020**, *13*, 3843. [[CrossRef](#)]
8. Keerthi, D.S.; Chaithanyaranga, H.V.; Shreedevi, P. Study and performance evaluation of mobility models in MANET and WSN. In Proceedings of the International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), Mysuru, India, 9–10 December 2016; pp. 297–301. [[CrossRef](#)]
9. He, D.; Sun, W.; Shi, L. The Novel Mobility Models Based on Spiral Line for Aerial Backbone Networks. *IEEE Access* **2020**, *8*, 11297–11314. [[CrossRef](#)]
10. Megyesi, D.; Matis, M.; Breda, R. Evaluation Tool for Group Detection in UAV Mobility Model. In Proceedings of the 2019 New Trends in Aviation Development (NTAD), Chlumec nad Cidlinou, Czech Republic, 26–27 September 2019; pp. 120–124. [[CrossRef](#)]
11. Paola, A.D.; Giammanco, A.; Re, G.L.; Morana, M. Human Mobility Simulator for Smart Applications. In Proceedings of the IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Cosenza, Italy, 7–9 October 2019; pp. 1–8. [[CrossRef](#)]
12. Banagar, M.; Dhillon, H.S. 3GPP-Inspired Stochastic Geometry-Based Mobility Model for a Drone Cellular Network. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [[CrossRef](#)]
13. Kulkarni, V.; Mahalunkar, A.; Garbinato, B.; Kelleher, J.D. Examining the limits of predictability of human mobility. *Entropy* **2019**, *21*, 432. [[CrossRef](#)]
14. Aschenbruck, N.; Gerhards-Padilla, E.; Martini, P. A survey on mobility models for performance analysis in tactical mobile networks. *J. Telecommun. Inf. Technol.* **2008**, *2*, 54–61.
15. Tuduca, C.; Gross, T. A mobility model based on WLAN traces and its validation. In Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA, 13–17 March 2005; Volume 1, pp. 664–674. [[CrossRef](#)]
16. Dobhal, D.C.; Dimri, S.C. The impact of the mobility of nodes on performance of TCP in MANET. In Proceedings of the International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 20–22 April 2017; Volume 1, pp. 149–155. [[CrossRef](#)]
17. Dorge, P.D.; Meshram, S.L. Design and Performance Analysis of Reference Point Group Mobility Model for Mobile Ad hoc Network. In Proceedings of the First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 15–17 December 2018; pp. 51–56. [[CrossRef](#)]
18. Xie, J.; Wan, Y.; Kim, J.H.; Fu, S.; Namuduri, K. A Survey and Analysis of Mobility Models for Airborne Networks. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1221–1238. [[CrossRef](#)]
19. Meghanathan, N. Impact of the Gauss-Markov Mobility Model on Network Connectivity, Lifetime and Hop Count of Routes for Mobile Ad hoc Networks. *J. Netw.* **2010**, *5*, 509–516. [[CrossRef](#)]
20. Gloss, B.; Scharf, M.; Neubauer, D. Location-Dependent Parameterization of a Random Direction Mobility Model. In Proceedings of the IEEE 63rd Vehicular Technology Conference, Melbourne, Australia, 7–10 May 2006; Volume 3, pp. 1068–1072. [[CrossRef](#)]
21. Bilgin, M. Novel random models of entity mobility models and performance analysis of random entity mobility models. *Turk. J. Electr. Eng. Comput. Sci.* **2020**, *28*, 708–726. [[CrossRef](#)]
22. Nain, P.; Towsley, D.; Benyuan Liu.; Zhen Liu. Properties of random direction models. In Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA, 13–17 March 2005; Volume 3, pp. 1897–1907. [[CrossRef](#)]
23. Cai, Y.; Wang, X.; Li, Z.; Fang, Y. Delay and capacity in MANETs under random walk mobility model. *Wirel. Netw.* **2014**, *20*. [[CrossRef](#)]

24. Aslam, M.; Khan, A.R. Comparison of Random Waypoint and Random Walk Mobility Model under DSR, AODV and DSDV MANET Routing Protocols. *arXiv* **2011**, arXiv:1104.2368.
25. Alshanyour, A.; Baroudi, U. Random and realistic mobility models impact on the performance of bypass-AODV routing protocol. In Proceedings of the 1st IFIP Wireless Days, Dubai, United Arab Emirates, 24–27 November 2008; pp. 1–5. [[CrossRef](#)]
26. Bugarčić, P.D.; Malnar, M.Z.; Jevtić, N.J. Performance Analysis of MANET Networks Based on AODV Protocol in NS-3 Simulator. In Proceedings of the 26th Telecommunications Forum (TELFOR), Belgrade, Serbia, 20–21 November 2018; pp. 1–4. [[CrossRef](#)]
27. Alenazi, M.J.; Sahin, C.; Sterbenz, J.P.G. Design Improvement and Implementation of 3D Gauss-Markov Mobility Model. In Proceedings of the International Telemetering Conference (ITC), San Diego, CA, USA, 22–25 October 2012.
28. Laqtib, S.; El Yassini, K.; Houmer, M.; El Ouadghiri, M.D.; Hasnaoui, M.L. Impact of mobility models on Optimized Link State Routing Protocol in MANET. In Proceedings of the International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, 26–29 October 2016; pp. 104–109. [[CrossRef](#)]
29. He, R.; Ai, B.; Stuber, G.L.; Zhong, Z. Mobility model-based non-stationary mobile-to-mobile channel modeling. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 4388–4400. [[CrossRef](#)]
30. Wang, W.; Wang, J.; Wang, M.; Wang, B.; Zhang, W. A realistic mobility model with irregular obstacle constraints for mobile ad hoc networks. *Wirel. Netw.* **2019**, *25*, 487–506. [[CrossRef](#)]
31. Li, X.; Zhang, T.; Li, J. A particle swarm mobility model for flying ad hoc networks. In Proceedings of the GLOBECOM-IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
32. Sharma, P.K.; Kim, D.I. Random 3D Mobile UAV Networks: Mobility Modeling and Coverage Probability. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2527–2538. [[CrossRef](#)]
33. Jo, Y.I.; Fathoni, M.F.; Kim, K. A New Mobility Model for Multi-UAVs Reconnaissance Based on Partitioned Zone. *Appl. Sci.* **2019**, *9*, 3810. [[CrossRef](#)]
34. Jo, Y.I.; Lee, S.; Kim, K.H. Overlap Avoidance of Mobility Models for Multi-UAVs Reconnaissance. *Appl. Sci.* **2020**, *10*, 4051. [[CrossRef](#)]
35. Sayeed, M.; Kumar, R. An efficient mobility model for improving transmissions in multi-UAVs enabled WSNs. *Drones* **2018**, *2*, 31. [[CrossRef](#)]
36. Appiah, M. Performance comparison of mobility models in Mobile Ad Hoc Network (MANET). In Proceedings of the 1st International Conference on Next Generation Computing Applications (NextComp), Moka, Mauritius, 19–21 July 2017; pp. 47–53. [[CrossRef](#)]
37. Abdullah, A.; Ozen, E.; Bayramoglu, H. Investigating the Impact of Mobility Models on MANET Routing Protocols. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 25–35. [[CrossRef](#)]
38. Alenazi, M.J.F.; Cheng, Y.; Zhang, D.; Sterbenz, J.P.G. Epidemic Routing Protocol Implementation in Ns-3. In Proceedings of the 2015 Workshop on Ns-3, Barcelona, Spain, 13–14 May 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 83–90. [[CrossRef](#)]
39. Lee, K.; Hong, S.; Kim, S.J.; Rhee, I.; Chong, S. SLAW: Self-Similar Least-Action Human Walk. *IEEE/ACM Trans. Netw.* **2012**, *20*, 515–529. [[CrossRef](#)]
40. Hagberg, A.A.; Schult, D.A.; Swart, P.J. Exploring Network Structure, Dynamics, and Function using NetworkX. In Proceedings of the 7th Python in Science Conference, Pasadena, CA, USA, 19–24 August 2008; Varoquaux, G., Vaught, T., Millman, J., Eds.; Los Alamos National Lab. (LANL): Los Alamos, NM, USA, 2008; pp. 11–15.
41. Gillies, S.; Bierbaum, A.; Lautaportti, K.; Tonnhofer, O. *Shapely: Manipulation and Analysis of Geometric Objects*; GitHub: San Francisco, CA, USA, 2007.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).