applied sciences

Article

Mobility Prediction of Mobile Wireless Nodes

Shatha Abbas [1], Mohammed J. F. Alenazi [1,*] and Amani Samha [2]

[1] Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; 439204371@student.ksu.edu.sa
[2] Department of Management Information Systems, College of Business administration, King Saud University, Riyadh 11451, Saudi Arabia; asamha@ksu.edu.sa
* Correspondence: mjalenazi@ksu.edu.sa

Abstract: Artificial intelligence (AI) is a fundamental part of improving information technology systems. Essential AI techniques have revolutionized communication technology, such as mobility models and machine learning classification. Mobility models use a virtual testing methodology to evaluate new or updated products at a reasonable cost. Classifiers can be used with these models to achieve acceptable predictive accuracy. In this study, we analyzed the behavior of machine learning classification algorithms—more specifically decision tree (DT), logistic regression (LR), k-nearest neighbors (K-NN), latent Dirichlet allocation (LDA), Gaussian naive Bayes (GNB), and support vector machine (SVM)—when using different mobility models, such as random walk, random direction, Gauss–Markov, and recurrent self-similar Gauss–Markov (RSSGM). Subsequently, classifiers were applied in order to detect the most efficient mobility model over wireless nodes. Random mobility models (i.e., random direction and random walk) provided fluctuating accuracy values when machine learning classifiers were applied—resulting values ranged from 39% to 81%. The Gauss–Markov and RSSGM models achieved good prediction accuracy in scenarios using a different number of access points in a defined area. Gauss–Markov reached 89% with the LDA classifier, whereas RSSGM showed the greatest accuracy with all classifiers and through various samples (i.e., 2000, 5000, and 10,000 steps during the whole experiment). Finally, the decision tree classifier obtained better overall results, achieving 98% predictive accuracy for 5000 steps.

Keywords: software defined networking; performance analysis; machine learning; wireless nodes; mobility models; artificial intelligence

Citation: Abbas, S.; Alenazi, M.J.F.; Samha, A. Mobility Prediction of Mobile Wireless Nodes. *Appl. Sci.* **2022**, *12*, 13041. https://doi.org/10.3390/app122413041

Academic Editor: Corrado Santoro

Received: 16 November 2022
Accepted: 15 December 2022
Published: 19 December 2022

1. Introduction and Motivation

Machine learning classification has recently garnered attention from information technology developers. It can indeed enhance many fields by improving infrastructure, protocols, and services, making life easier and more predictable. Consequently, numerous studies on machine learning classification have looked at improving classifier algorithms, how to use them, or even at which algorithm performs best for a particular task. Mobility models are a notable feature in this domain. They simulate the movements of mobile nodes in a typical testing environment in order to evaluate the newest techniques, since examining these techniques in a real network is time-consuming and expensive.

Classification algorithms are useful in various functions, from simple binary to multi-category classifiers. Each algorithm has a unique manner of doing the required job. However, all algorithms need a hypothetical environment in which to evaluate a given task. Mobility models are able to fill such a need by offering an appropriate virtual environment. Therefore, there are basic popular mobility models and novel mobility models. Basic models can implement general applications, such as Gauss–Markov or random models (random waypoint, random direction, and random walk), while novel ones are used for more specific tasks, as it is the case with RSSGM [1].

Many basic mobility models have widespread usage in wireless networks, such as Gauss–Markov, random direction, and random walk. These models mimic the movement of nodes by supposing the position, direction, and speed of mobile nodes in a virtual environment. However, most mobility models simulate user movement in a completely randomized way, as they do not have enough resilience to dominate the randomness transition. In consequence, they cannot simulate recurrent human behavior that has a daily routine (i.e., the movement of human beings is not entirely random). This problem has been solved by the novel recurrent self-similar Gauss–Markov (RSSGM) mobility model, proposed in our previous work. This model executes recurrent node movement with semi-similar routes by controlling the degree of randomness, which helps the model capture repeated visits by a user to a same location. The RSSGM algorithm's performance, which was determined in [1], makes it a more realistic and suitable model for applications with semi-similar routes, such as lifestyle patterns, airplane controls, or public transportation in a wireless network.

Recent studies made use of classification models for the targeted task, whereas others used mobility models to do a specific job. Yet, each of them discussed classifications separately or with a particular mobility model, as in [2], where the authors applied four machine learning (ML) models: DNN, SVM, semi-Markov, and extreme gradient boosting trees simulated by the self-similar least action walk (SLAW) mobility model. In [3], various ML models were used to predict network traffic classification. Results show that the decision tree reached a better average accuracy (i.e., 99.18%) than other models, and higher performance than the port-based method. However, our paper has addressed this gap to comprehensively study the relationship between classifiers and mobility models.

In this paper, we aim to demonstrate the effectiveness of mobility models when using classification algorithms. The best way to achieve this goal is by analyzing factors that directly affect the network performance, that is to say the classifier's behavior over different types of mobility models. Recording the accuracy of prediction for user movement during every experiment is considered the main evaluation parameter in this study. Therefore, the evaluation was carried out by applying six classification machine learning algorithms (i.e., decision tree, logistic regression, k-nearest neighbors, latent Dirichlet allocation, Gaussian naive Bayes, and support-vector machine) that rely on mobility models. We used four types of mobility models, i.e., RSSGM, Gauss–Markov, random direction, and random walk. To analyze mobility models' ability in all cases, we alternately paired the number of centroid nodes (5, 10, 15, and 20) with the number of access points (5, 10, 15, and 20) to cover sixteen diverse scenarios. Then, we compared the accuracy and performance of all types. Finally, we applied the better mobility model's results through a different count of steps to study the classification behavior if the user has to move more or fewer steps than average.

Comparison results of this work depend on the prediction accuracy of user movement. Prediction models based on random mobility models vary in terms of prediction accuracy, starting from 39% and not exceeding 81%. On the contrary, Gauss–Markov's accuracy reached 89% using the LDA classifier. Finally, the RSSGM model was shown to have a very high prediction accuracy in all cases, hitting 98% with the decision tree classifier through the average movement. This result indicates that RSSGM is sufficient for the classification prediction of repeated visits to particular locations.

The remainder of this paper is organized as follows. A brief theoretical background on categories of mobility models and several types of machine learning classification algorithms is introduced in Sections 2 and 3. In Section 4, the related literature is reviewed. The study methods used are described in Section 5. Evaluation and analysis are presented in Section 6, and a discussion is presented in Section 7. Lastly, concluding observations and suggested future research guidelines are discussed in Section 8.

2. Mobility Models

The number of existing mobility models has been growing quickly to meet the needs of research. Mobility models are necessary to evaluate the performance of a new network technique before it is given credence and implemented in a real network [4]. Many mobility models (e.g., random, temporal, and spatial-dependency models) have been created to be used with different network statuses. However, incoming and advanced mobility models are being designed to fit multiple applications. The RSSGM model is among these new mobility models, and is suitable for applications predicting patterns of recurrent visits to specific stations with semi-similar routes [1].

2.1. Gauss–Markov

Gauss–Markov mobility is a well-known model proposed by Liang and Haas [5]. This model applies a Gaussian concept. The future speed and direction of a node depend on its current average speed and direction, as well as on a Gaussian random noise process. The mobile node will move for a fixed period, the the Gauss–Markov model parameters are calculated again [6,7].

2.2. Random Waypoint

The random waypoint (RWP) model is used to mimic users' movement in a network. User nodes in an RWP model pick a random and independent location and destination in a specific scope with an assigned speed. Then, the node moves as directed by the previous information until it reaches the target destination, where it pauses for a moment before beginning the new selection [8,9].

2.3. Random Walk

The first mention of the random walk (RW) model occurred in 1926 when Einstein used it to describe nodes' random movements. In this model, the node defines a random speed and direction and then walks a straight line identified by chosen parameters until it reaches its destination. This process is repeated until the goal is achieved [10,11]. Figure 1b illustrates how the RW model works.

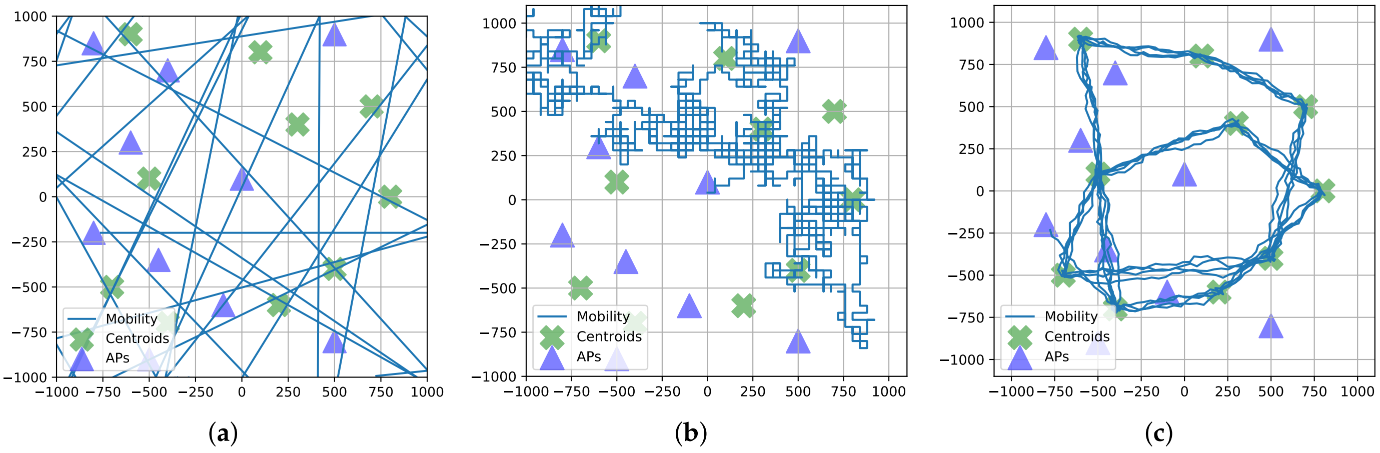


Figure 1. Examples of different mobility models' patterns using five access points (APs) and ten centroids. (**a**) Random direction mobility model. (**b**) Random walk mobility model. (**c**) RSSGM mobility model.

2.4. Random Direction

The execution process of the random direction (RD) model is similar to that of the RWP model, except that, in the case of an unexpected direction, the node picks a destination on the border of a defined region with random speed and direction. The node walks within these values until it reaches the boundary, where it pauses momentarily before selecting a new direction and speed [10,12,13]. Figure 1a represents how the random direction model works.

2.5. RSSGM

The RSSGM model is a relatively new model created to mimic human movement that has a recurrent pattern and self-similarity behavior. This model relies on the Gauss–Markov mobility model to gain temporal dependencies and random characteristics. This addition guides the direction of nodes' travels to predefined places, called centroids, that are regularly visited. Furthermore, the RSSGM model provides self-similarity movement simulating real human mobility [1]. Figure 1c shows the behavior of the RSSGM model.

3. Machine Learning Classifiers

Machine learning (ML) has been having a significant effect on the development of science and technology. ML mechanisms are essential in many fields, such as image classification and machine translation. The working procedure of ML depends on the iterative operations in which historical information is collected for training and evaluation. However, classification is also a significant factor in ML research and data mining, which are valuable for various applications, such as for unmanned driving and for the healthcare sector, e.g., services, facilities, and hospitals [14–17].

There are two types of ML models. The first one corresponds to supervised models, in which a machine learns to predict the outcome variable using labeled training data. The second type is the unsupervised models, which work without labeled data. Instead, they discover the predictor's associations [18]. However, there are various classification algorithms for ML, as they were designed to meet the needs of different researchers. A few are cited below.

3.1. Logistic Regression

Logistic regression (LR) is a supervised ML algorithm that has a reputation in binary classification problems with two classes. There are three categories of LR, i.e., binary, ordinal, and nominal. LR predicts the output as a dependent random variable using independent random variables (continuous or categorical), as well as a constant value. LR is a simple algorithm in which an input number is added to an LR equation, and probability is the result [19,20].

3.2. Decision Tree

The decision tree (DT) algorithm creates a supervised ML model. This model involves a tree structure separated into three parts. The root represents all data, splits are the central nodes, and leaves are the last nodes. This classification model gives an output based on trained input data by splitting complicated decisions into a simple sequence [19,21].

3.3. K-Nearest Neighbors

The k-nearest neighbors (KNN) model is a type of supervised ML model. This algorithm is characterized by simplicity and is widely used in clustering, classification, and many data mining applications. The KNN model is a non-parametric technique that gives weights to neighbors' contributions (the nearest points have a larger effect than far-away points) by calculating the distance between test nodes and training nodes, where training nodes are the neighboring points. In general, KNN offers high performance and is fast because it does not have a separate training phase [21,22].

3.4. Latent Dirichlet Allocation

The latent Dirichlet Allocation (LDA) is an unsupervised ML technique model introduced in 2010. The main application of the LDA is the text field. This model is in fact used to recognize implicit information in a large document by calculating the probability distribution of words from a document's vocabulary list or multiple documents. The LDA model contains three layers, i.e., words, topics, and documents. It is also called the three-layer Bayesian probability model [23,24].

3.5. Gaussian Naive Bayes

The Gaussian naive Bayes (GNB) classifier is a simple probabilistic method that applies the Bayes theorem. In this model, probability depends on the attributes for all variables in the training data set. Therefore, the GNB model is characterized by high performance in supervised learning and in complicated real-world problems. The GNB model also has the advantage of not requiring too many parameters of training data in order to build the classification [25,26].

3.6. Support Vector Machines

The support vector machine (SVM) algorithm is a common supervised ML algorithm that was introduced in [27]. It relies on statistical theory and is mainly used for statistical classification and regression problems. This model tries to determine the optimal hyperplane by maximizing the distance between two boundaries [21,28].

These ML algorithms were used in our study to compare the accuracy performance of classifiers that rely on different mobility models.

4. Related Works

Many studies have been conducted on mobility models and their importance in improving technology. For example, in [8], several mobility models were used to show the performance of an AODV routing protocol over constant bit rate (CBR) and variable bit rate (VBR) data. Furthermore, the authors in [12] used three types of mobility models, taking into account the effects of these mobilities on node density, to study the behavior of various delay-tolerant networking (DTN) routing protocols.

The authors in [9] sought to define the importance of the mobility model for handover rates in heterogeneous wireless networks over urban regions. Moreover, ref. [13] showed that mobility models can enhance the connectivity of a network under the leader–follower model with mobile leaders.

In [10], two novel random models were offered as alternatives to random entity mobility models. Alenazi et al. described a new model, RSSGM, that is useful with nodes that have semi-similar routes [1].

ML is a popular technique used in AI technology and has considerable ability to enhance different systems. Consequently, there is a considerable amount of research investigating ML and its implementation in several applications. Ref. [15] summarized the performance of eight ML models (i.e., LR, KNN, SVM, DT, random forest, gradient boosting method, and naive Bayes). The authors examined models' ability to predict breast cancer in women, determining that the best ML model for this purpose is the gradient boosting classifier, with an accuracy of 74.14%. However, ref. [25] found that the GNB algorithm successfully predicted breast and lung cancer with an accuracy of 98% and 90%, respectively.

As for [16], the authors presented a new ML model for predicting the quality of energy model systems. Moreover, ref. [18] proposed a new ML model, called stepwise-support vector machine (StepSVM), based on the SVM model and used for stepwise model selection. In [19], the authors presented a model that works by tuning the current XGBoost model. Their new model was shown to perform better than several other models (LR, DT, random forest, AdaBoost, and XGBoost) using four datasets (NASA-KC2, PC3, JM1, and CM1). In [22], the k-NN classifier was used, after applying three feature selection algorithms, i.e., CBFS, FPRS, and KFRS, to a leukemia dataset in order to improve the model's cancer classification ability.

The authors in [2] applied four ML models: a deep neural network, extreme gradient boosting trees, semi-Markov, and SVM to investigate the performance in terms of energy and accuracy. They evaluated the models' performance using 84 users, simulated by the self-similar least action walk (SLAW) mobility model. They found that the XGBoost was the most accurate, achieving 90% accuracy and exceeding 80% on energy saving. Ref. [29] compared variation logit models and ML in terms of model growth, evaluation, and behavioral mode-choice modeling, providing an understanding of both models' strengths and shortcomings. Results of the analysis suggest that ML models are preferable to logit models. Ref. [30] provided a complete image of ML and deep learning (DL), including definitions, the generation of a dataset in an ad-hoc network, and where ML can be used, extracting valuable information from several research articles. They defined the issues and challenges that face ML and DL in a wireless network. In [23], the authors presented a weighted latent Dirichlet allocation (W-LDA) that overcomes the weaknesses of a classic LDA. Another work, ref. [24], suggested an LDA-based data augmentation algorithm that provides a large training dataset to reveal key audio events in a recording, and further creates new recordings based on that key. Additionally, many articles have discussed and applied machine learning algorithms to improve various applications and systems such as [3,31,32].

The authors in [33] suggested a method for non-orthogonal multiple access (NOMA) depending on the satellite-terrestrial integrated network (STIN). This method was compared with recent works, and it proved the significance and importance of groups of users in a cluster manipulated with NOMA technology. In [34], the grounded multi-antenna was suggested to work as a green interference to improve the transmission security of the physical layer in the satellite network. Results showed high performance and attitude. As for [35], the authors suggested a strategy to reduce satellites' transmission power and BS while keeping users' rate requirements. This method was designed using singular value decomposition and uplink–downlink duality. Finally, the authors in [36] aimed to increase security and decrease the network's consumed power by using SLNR and SCA while applying a new iterative algorithm.

5. Methodology

The current study was concerned with mobility models and how ML algorithms are affected by these models. We used four mobility models, i.e., GM, RD, RW, and RSSGM, to perform mobility node movement, generating a dataset of user locations, as well as current and previously connected access points. Then, different classifiers were applied, i.e., DT, LR, KNN, LDA, GNB, and SVM, using the previously produced dataset, to predict user activity. Finally, we evaluated each classifier's accuracy to determine which one could best predict mobility behavior. Figure 2 briefly illustrates the steps taken in this study.

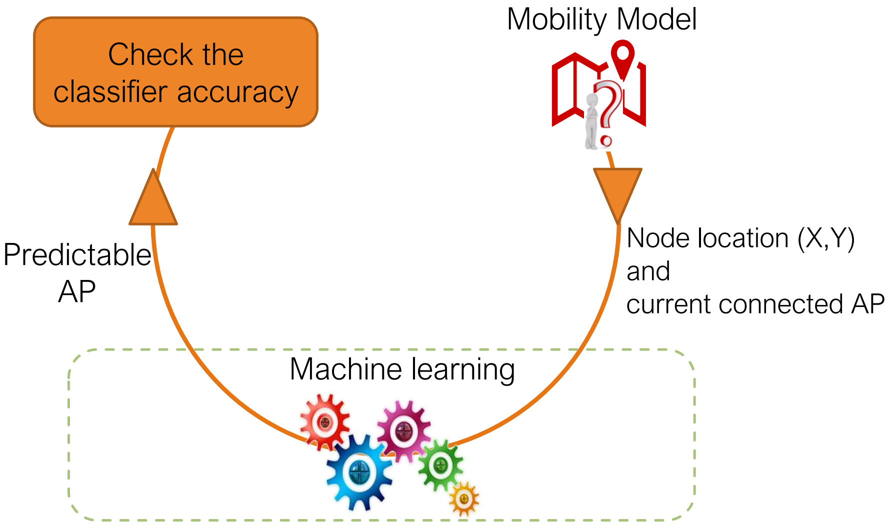


Figure 2. The method used in our study.

In the first phase of this work, mobility models were implemented using Python in sixteen scenarios using a variable number of access points (5, 10, 15, and 20) alternately with a different number (5, 10, 15, and 20) of nodes that have recurrent visits—referred to as centroids. All of the nodes (i.e., access points and centroids) had fixed locations in a predefined domain with an area of [−1000, 1000] meters. Table 1 shows the locations of chosen nodes. The positions of the nodes were randomly distributed to cover the entire region. Figure 3 shows an example of how nodes extended into a certain area, using fifteen access points and ten centroids as an example. We applied the mobility models using 5000 steps, which represents the average amount an adult walks in a day [37]. The execution of the mobility model provides a dataset of user positions that the node will visit before reaching its target location. The connected access point is determined using the shortest-path method to identify the nearest access point (AP). Table 2 shows some of the data generated using the RSSGM model.

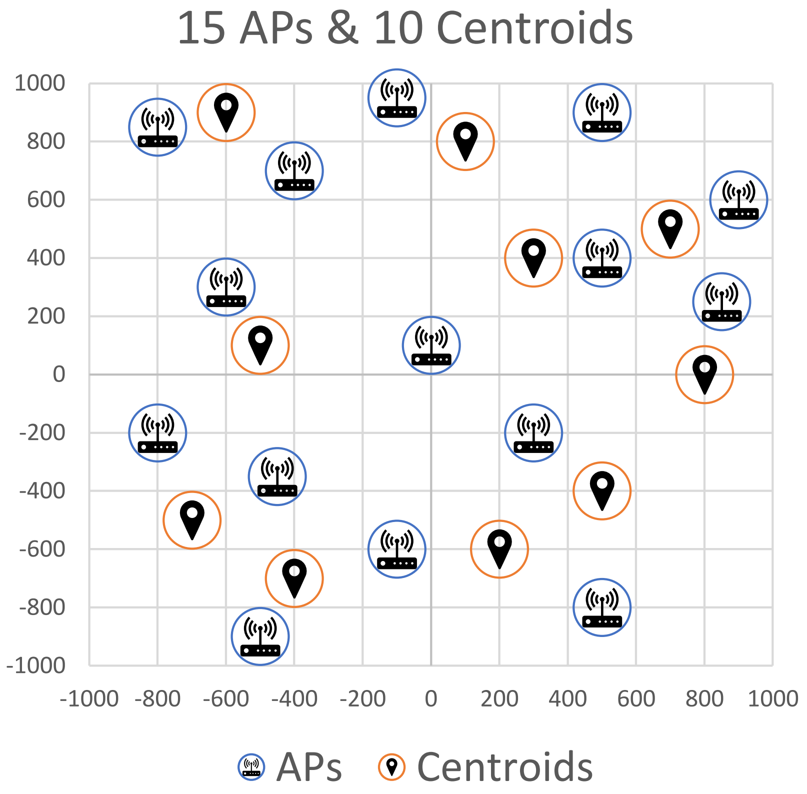


Figure 3. Example of how nodes extend over the predefined area using 15 access points and 10 centroids.

In the second phase, we used six ML classifiers (i.e., DT, LR, KNN, LDA, GNB, and SVM) based on the dataset of locations generated by different mobility models, current and previous APs. The ML classifier was also used to predict the future connected AP, that is to say the AP that the node will connect with. Finally, the classifier accuracy metrics' performance was evaluated using the Scikit-learn library. Scikit-learn is a well-known library dealing with issues associated with data mining and data analysis [38]. Subsequently, we used this information to determine the best classifier algorithm with multiple mobility models to provide better movement prediction based on classifier accuracy results.

Table 1. Access points and centroid locations.

APs	X (m)	Y (m)	Centroids	X (m)	Y (m)
1	−800	−200	1	−400	−700
2	−400	700	2	−600	900
3	0	100	3	100	800
4	500	900	4	700	500
5	500	−800	5	500	−400
6	−500	−900	6	−700	−500
7	−450	−350	7	−500	100
8	−100	−600	8	300	400
9	−600	300	9	800	0
10	−800	850	10	200	−600
11	−100	950	11	−200	−200
12	900	600	12	−800	400
13	500	400	13	800	−800
14	850	250	14	−100	500
15	300	−200	15	200	0
16	200	−850	16	−900	−900
17	700	−500	17	−100	−900
18	−300	0	18	500	100
19	100	550	19	900	−400
20	−900	−700	20	−900	700

Table 2. Example of a dataset generated using the RSSGM model.

Time (s)	X (m)	Y (m)	Previous AP	Current AP
32	−522.039	234.0297	0	1
33	−534.758	282.532	1	1
34	−545.7	331.5814	1	1
35	−551.134	381.5525	1	1
36	−560.975	430.9473	1	1
		. . .		
467	149.2532	660.1039	2	3
468	114.1719	625.27	3	3
469	76.80538	593.0601	3	1
470	39.04904	561.3788	1	1
471	5.846693	524.811	1	2
472	−25.0253	486.2131	2	2
473	−18.7087	535.4928	2	1
474	−1.68525	582.249	1	1
475	23.00982	625.3539	1	1
476	45.7389	669.5808	1	1
477	58.65014	717.8236	1	1
478	75.00265	764.9371	1	3
479	86.98774	813.4452	3	3
480	132.711	793.1574	3	3
		. . .		
799	680.0722	395.8155	2	3
800	628.7113	401.9032	3	3

In the last phase of the methodology, we experience the performance of classifications that rely on the better mobility model's results (collected in the second phase) by using 2000, 5000, and 10,000 steps to compare the classification behavior when the user had to move more or fewer steps than the average.

6. Results and Evaluation

This section evaluates the accuracy of the predictions of ML classifiers using a dataset generated by different mobility models. The assessment was made using a classifier accuracy performance metric, i.e., the score function in Scikit-learn library [39]. Results were extracted using six types of classifiers (DT, LR, KNN, LDA, GNB, and SVM) and four types of mobility models (Gauss–Markov, random direction, RW, and RSSGM) in different scenarios by changing the numbers of APs in a predefined area.

6.1. Evaluation of the RD Mobility Model

The first mobility model applied in this experiment was an RD model. The accuracy of the DT classifier was consistent around 79% in all scenarios, but the accuracy of other classifiers decreased as the number of APs increased. For instance, the SVM classifier was 85% accurate with five APs and 57% accurate with twenty APs. GNB, KNN, LDA, and LR presented a similar pattern. LR had lower average accuracy than other classifiers, i.e., 39%. The accuracy of the classifiers using a different number of APs is presented in Figure 4.

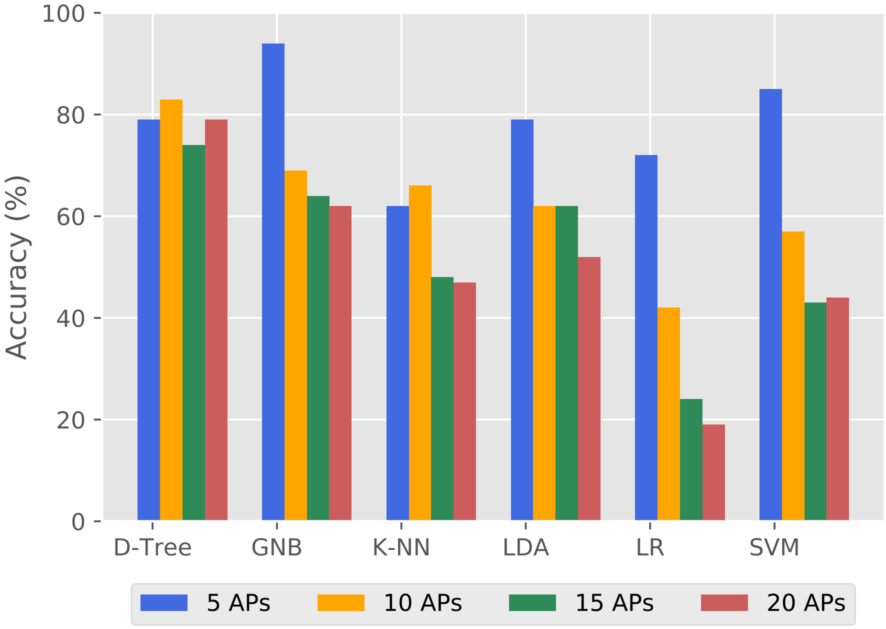


Figure 4. Average accuracy with different number of APs using RD mobility.

6.2. Evaluation of the RW Mobility Model

The second mobility model tested was the RW model. Figure 5 represents the results of the RW model's accuracy. As noted previously, accuracy values dropped as the number of access points increased in all classifier algorithms, indicating that algorithms had difficulty predicting random movement, especially when there were many transitions between APs. K-NN showed the lowest accuracy of 58%, while GNB, LR, DT, SVM, and LDA reached 59%, 61%, 65%, 68%, and 81%, respectively.

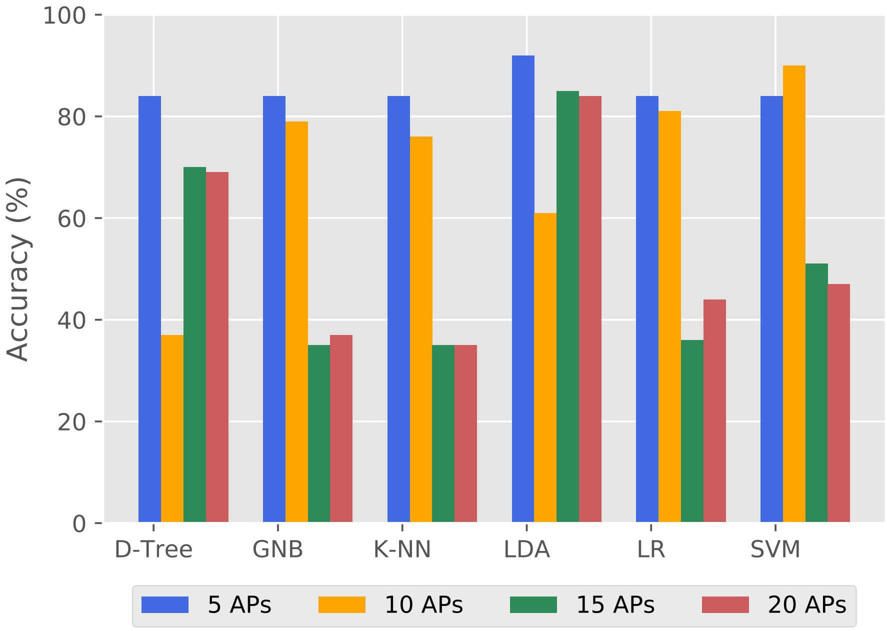


Figure 5. Average accuracy with a different number of APs using the RW mobility model.

6.3. Evaluation of the Gauss–Markov Mobility Model

Gauss–Markov is the third mobility model used in this study. Figure 6 represents the model's accuracy in all scenarios. The accuracy of LR was significantly affected by an increase in the number of APs, decreasing from 84% to 49%. The other classifiers had good results in all scenarios, with the average prediction accuracy reaching 89% with LDA, 88% with GNB and DT, 71% with SVM, and 66% with k-NN.

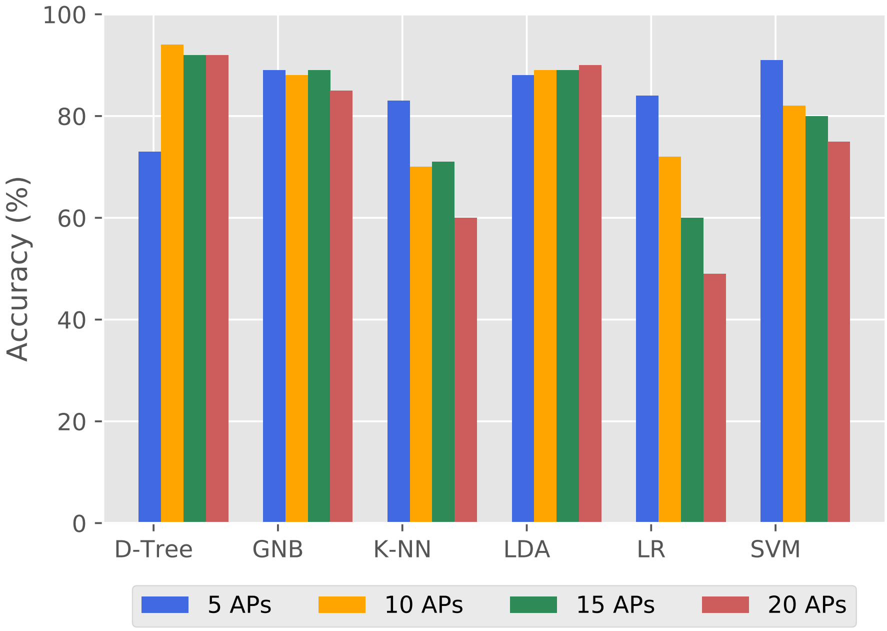


Figure 6. Average accuracy with a different number of APs using the Gauss–Markov mobility model.

6.4. Evaluation of the RSSGM Mobility Model

The RSSGM mobility model was the last model considered. This model is used to simulate user movement that has a self-similarity routine. RSSGM achieved very high prediction accuracy in all scenarios and with all classifiers. The LR algorithm had the lowest accuracy (80% on average) when compared with other classifiers and, as noted in LR, the number of APs affected the accuracy in inverse proportion. When five APs were used, the accuracy value was 91%; when twenty APs were used, the accuracy decreased to 67%. Generally, all classifiers had the same behavior of an inverse relationship between accuracy and the number of APs. However, SVM, LDA, and KNN presented a good prediction accuracy (i.e., an average of 92%). The DT classifier was the most accurate, with 98% on average. Such a result indicates that the DT classifier is the best classifier to use with the RSSGM model. Classifier accuracy is shown in Figures 7 and 8, illustrating the behavior of classifiers when the numbers of APs and centroids are changed.

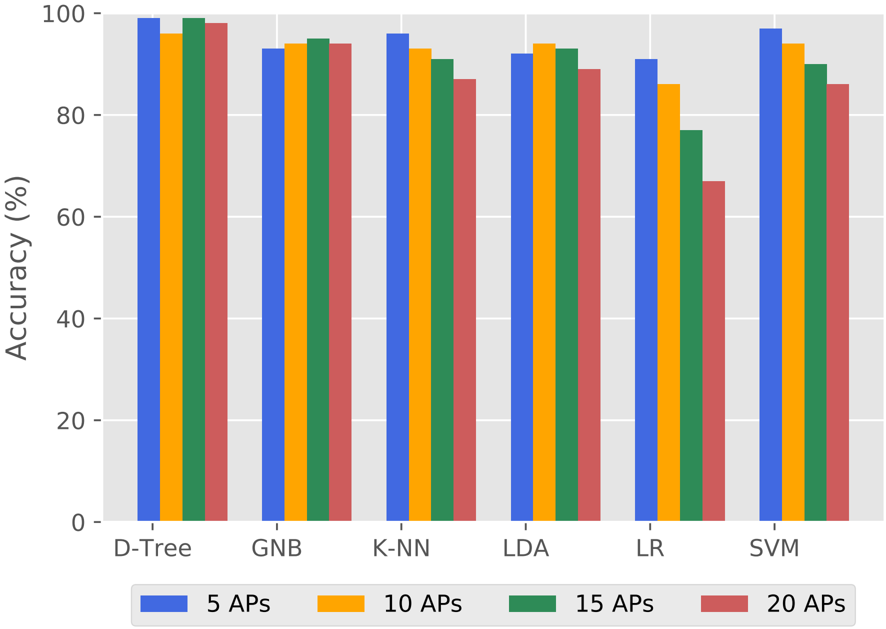


Figure 7. Average accuracy with a different number of APs using the RSSGM model.

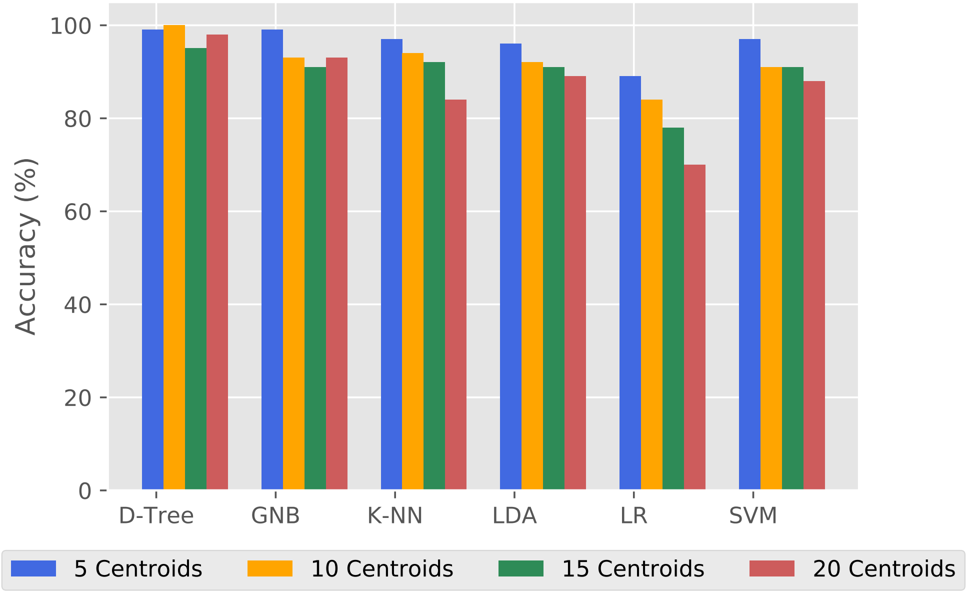


Figure 8. Average accuracy with a different number of centroids using the RSSGM model.

7. Discussion and Analysis

This section presents a discussion of the results of the implementation of ML classifiers using datasets generated by different mobility models. The datasets indicate node location as well as previous and current APs. Current APs were determined by choosing APs nearest to the mobile node until the target location was reached. This experiment was evaluated using a classifier accuracy performance metric, that is to say the score function in the Scikit-learn library.

Figure 9 summarizes the total average accuracy of all six classifiers in all sixteen scenarios using four mobility models. As previously noted, the prediction accuracy in random mobility models (i.e., RW and RD) fluctuated when the number of APs increased, with accuracy values starting from 39% and never exceeding 81%. On the contrary, the patterns of the GM and RSSGM models were constant and had high ranges of accuracy. The accuracy of the GM model reached 89% with the LDA classifier and fell to 66% with the LR model. However, the last mobility model tested, i.e., RSSGM, achieved the highest accuracy, hitting 98% with the DT classifier, 94% with GNB, 92% with KNN, SVM, and LDA, and 80% with the LR algorithm.

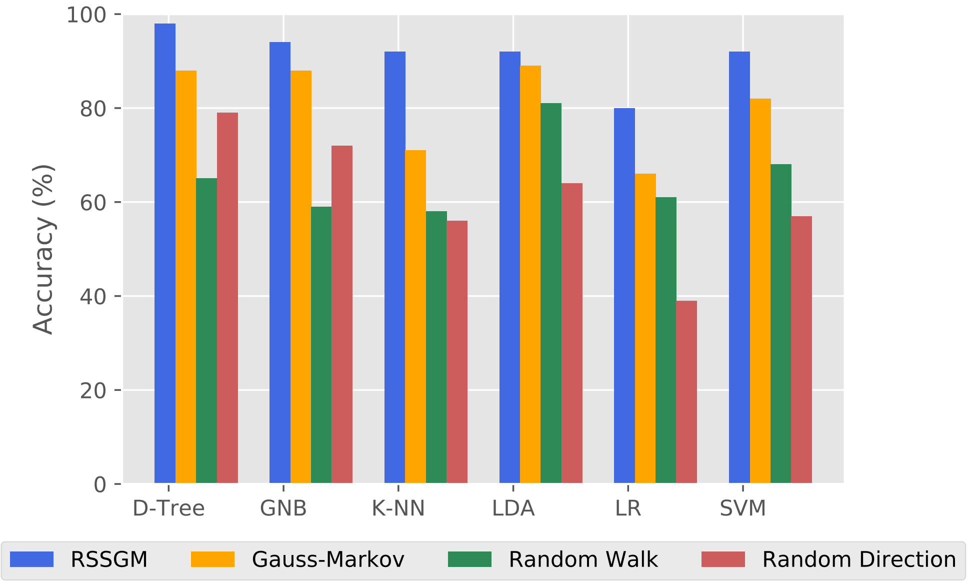
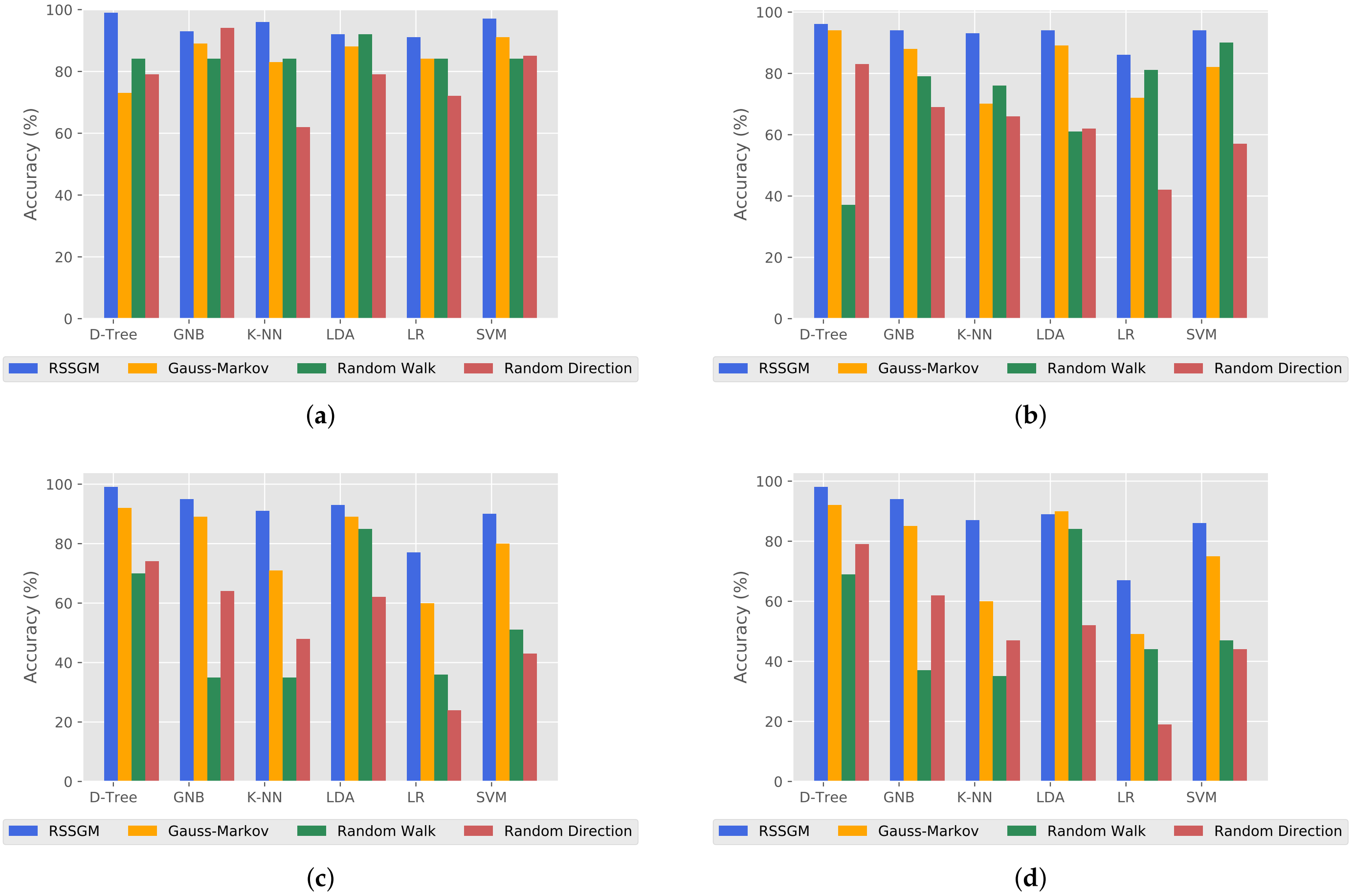


Figure 9. Total average accuracy.

Figure 10 illustrates the mobility models' accuracy when the number of APs increased to indicate the classifier's behavior in a high-scale motility network. RSSGM and GM had high prediction accuracy in all scenarios, reaching 98% and 89%, respectively, whereas the RD and RW models had low accuracy at 39% and 58% on average and going to 79% and 82%, respectively. However, the DT model had an excellent level of prediction accuracy with different numbers of APs, especially when using the RSSGM model, where it was 99% accurate with five APs. The RD had insignificant values, especially with the LR classifier, where it had only 39% accuracy.

To increase the confidentiality of the results, we reclassified the RSSGM model for all algorithms with a different number of steps to compare the classification behavior when the user had to move more or fewer steps than average. Therefore, we added two experiments (i.e., 2000 and 10,000 steps) over the previous scenarios, and then collected and compared obtained results with 5000 steps. The results show a positive correlation between step count and classification accuracy, based on expanding the learning and understanding between user movement and classifier when the number of trials increases. Moreover, all results of applying classifiers based on the RSSGM model are considered to have very high accuracy performance. Figure 11 shows the comparison between all scenarios.



(**a**) (**b**)

(**c**) (**d**)

Figure 10. Example of different mobility model patterns using different number of APs and ten centroids. (**a**) Classifier accuracy with 5 APs. (**b**) Classifier accuracy with 10 APs. (**c**) Classifier accuracy with 15 APs. (**d**) Classifier accuracy with 20 APs.

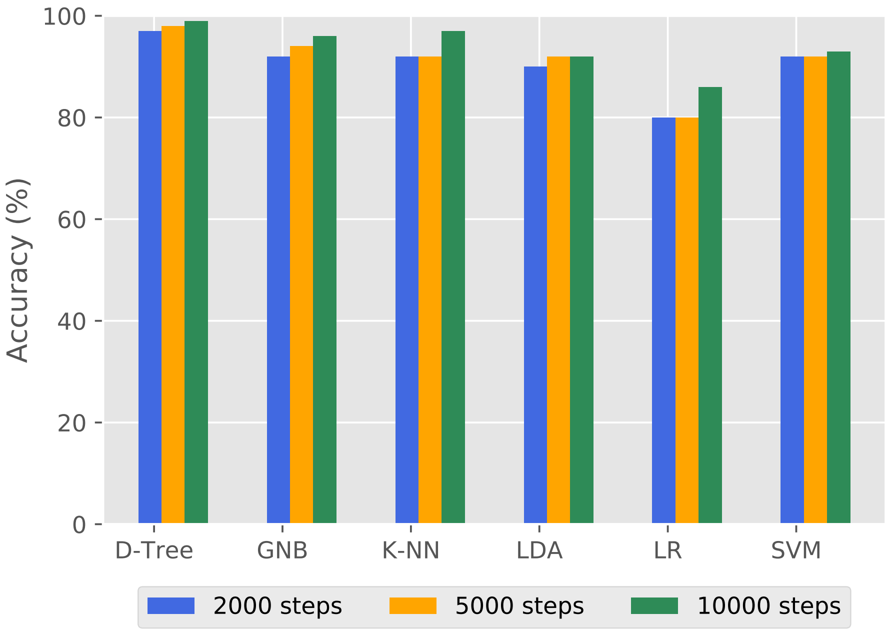


Figure 11. Average accuracy with a different number of Steps.

8. Conclusions and Future Work

In this study, we evaluated the performance of classifiers and mobility models using sixteen distinct scenarios. Four mobility models (i.e., RW, RD, Gauss–Markov, and RSSGM) and six classifiers (i.e., DT, LR, KNN, LDA, GNB, and SVM) were used. We also explored the effectiveness of mobility models by comparing the prediction accuracy of these classifiers. The results show that random models (i.e., RD and RW) varied in their prediction accuracy, with a minimum of 39% and a maximum of 81%. The Gauss–Markov model had good accuracy, reaching 89% with the LDA classifier. The RSSGM model was found to have very high prediction accuracy, achieving 98% accuracy with the DT classifier. Moreover, RSSGM reached a high accuracy and a positive correlation between step count and classification accuracy, based on expanding the learning and understanding between user movement and classifier. Therefore, RSSGM is ideal for the classification prediction of nodes with repeated visits to particular locations.

This experiment is a modeling type that requires less time and can implement many scenarios to provide an initial impression of the classification behavior over different mobility models. However, as a next step, the experiment shall capture more accurate behavior, and future works should include other ML classifiers (e.g., XGboost) to pick the best classifier for use with mobility models. Furthermore, the simulation should consider more parameters to provide more accurate results, such as an RSSGM to evaluate the theoretical work.

Author Contributions: S.A. performed the experiments, analyzed the data, and wrote the manuscript. M.J.F.A. supervised the research and critically revised the manuscript. A.S. critically revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to Researcher Supporting Project number (RSPD2023R582), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Alenazi, M.J.F.; Abbas, S.O.; Almowuena, S.; Alsabaan, M. RSSGM: Recurrent Self-Similar Gauss–Markov Mobility Model. *Electronics* **2020**, *9*, 89. [CrossRef]
2. Gebrie, H.; Farooq, H.; Imran, A. What Machine Learning Predictor Performs Best for Mobility Prediction in Cellular Networks? In Proceedings of the 2019 IEEE International Conference on Communications Workshops (ICC Workshops), Shanghai, China, 20–24 May 2019; pp. 1–6. [CrossRef]
3. Alzoman, R.; Alenazi, M. A Comparative Study of Traffic Classification Techniques for Smart City Networks. *Sensors* **2021**, *21*, 4677. [CrossRef] [PubMed]
4. Gharib, M.; Foroozani, A.; Rezaei, S.; Hemmatyar, A.; Movaghar, A. An Area-Scalable Human-Based Mobility Model. *Comput. Netw.* **2020**, *177*, 107300. [CrossRef]
5. Liang, B.; Haas, Z. Predictive distance-based mobility management for PCS networks. In Proceedings of the IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320), New York, NY, USA, 21–25 March 1999; Volume 3, pp. 1377–1384. [CrossRef]
6. Biomo, J.D.M.M.; Kunz, T.; St-Hilaire, M. An enhanced Gauss–Markov mobility model for simulations of unmanned aerial ad hoc networks. In Proceedings of the 2014 7th IFIP Wireless and Mobile Networking Conference (WMNC), Vilamoura, Portugal, 20–22 May 2014; pp. 1–8. [CrossRef]
7. Ariyakhajorn, J.; Wannawilai, P.; Sathitwiriyawong, C. A Comparative Study of Random Waypoint and Gauss–Markov Mobility Models in the Performance Evaluation of MANET. In Proceedings of the 2006 International Symposium on Communications and Information Technologies, Bangkok, Thailand, 18–20 October 2006; pp. 894–899. [CrossRef]
8. Gowda, S.G.; Jacob, S. Network Mobile Topology Impact QOS in Multiservice Manet. *Acta Tech. Corvininesis-Bull. Eng.* **2020**, *13*, 79–85.

9. Khaki, M.; Ghasemi, A. The impact of mobility model on handover rate in heterogeneous multi-tier wireless networks. *Comput. Netw.* **2020**, *182*, 107454. .: 10.1016/j.comnet.2020.107454. [CrossRef]
10. 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]
11. Banagar, M.; Dhillon, H.S. Performance Characterization of Canonical Mobility Models in Drone Cellular Networks. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 4994–5009. [CrossRef]
12. Hossen, M.S.; Rahim, M. Analysis of Delay-Tolerant Routing Protocols using the Impact of Mobility Models. *Scalable Comput.* **2019**, *20*, 17–26. [CrossRef]
13. Norouzi Kandalan, R.; Alla, S.; Rezaeian, N. Impact of Mobility on Consensus Building in the Leader-Follower Model. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–6. [CrossRef]
14. Ferreira, L.A.; Guimarães, F.G.; Silva, R. Applying Genetic Programming to Improve Interpretability in Machine Learning Models. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8. [CrossRef]
15. Austria, Y.; Goh, M.; Jr, L.; Lalata, J.A.; Goh, J.; Vicente, H. Comparison of Machine Learning Algorithms in Breast Cancer Prediction Using the Coimbra Dataset. *Int. J. Simul. Syst. Sci. Technol.* **2019**. [CrossRef]
16. Gu, Z.; Wang, J.; Luo, S. Investigation on the quality assurance procedure and evaluation methodology of machine learning building energy model systems. In Proceedings of the 2020 International Conference on Urban Engineering and Management Science (ICUEMS), Zhuhai, China, 24–26 April 2020; pp. 96–99. [CrossRef]
17. Luo, Y. Uncertainty of the Classification Result from a Linear Discriminant Analysis. In Proceedings of the 2020 IEEE International Workshop on Metrology for Industry 4.0 IoT, Roma, Italy, 3–5 June 2020; pp. 101–105. . MetroInd4.0IoT48571.2020.9138186. [CrossRef]
18. Guo, C.Y.; Chou, Y.C. A novel machine learning strategy for model selections—Stepwise Support Vector Machine (StepSVM). *PLoS ONE* **2020**, *15*, e0238384. [CrossRef]
19. Gupta, A.; Sharma, S.; Goyal, S.; Rashid, M. Novel XGBoost Tuned Machine Learning Model for Software Bug Prediction. In Proceedings of the 2020 International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 17–19 June 2020; pp. 376–380. [CrossRef]
20. Isak-Zatega, S.; Lipovac, A.; Lipovac, V. Logistic regression based in-service assessment of mobile web browsing service quality acceptability. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 1–21. [CrossRef]
21. Saber, M.; El Rharras, A.; Saadane, R.; Kharraz, A.H.; Chehri, A. An Optimized Spectrum Sensing Implementation Based on SVM, KNN and TREE Algorithms. In Proceedings of the 2019 15th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), Sorrento, Italy, 26–29 November 2019; pp. 383–389. [CrossRef]
22. Begum, S.; Chakraborty, D.; Sarkar, R. Data Classification Using Feature Selection and kNN Machine Learning Approach. In Proceedings of the 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India, 12–14 December 2015; pp. 811–814.
23. Xiaodong, T. Topic Extraction and Classification Method Based on Comment Sets. **2020**, *16*, 329–342. . JIPS.04.0165. [CrossRef]
24. Leng, Y.; Zhao, W.; Lin, C.; Sun, C.; Wang, R.; Yuan, Q.; Li, D. LDA-based data augmentation algorithm for acoustic scene classification. *Knowl.-Based Syst.* **2020**, *195*, 105600. .: 10.1016/j.knosys.2020.105600. [CrossRef]
25. Kamel, H.; Abdulah, D.; Al-Tuwaijari, J.M. Cancer Classification Using Gaussian Naive Bayes Algorithm. In Proceedings of the 2019 International Engineering Conference (IEC), Erbil, Iraq, 23–25 June 2019; pp. 165–170.
26. Tzanos, G.; Kachris, C.; Soudris, D. Hardware Acceleration on Gaussian Naive Bayes Machine Learning Algorithm. In Proceedings of the 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, Greece, 13–15 May 2019; pp. 1–5.
27. Vapnik, V.N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 1999.
28. Shao, Y.; Yuan, X.; Zhang, C.; Liu, C. Rolling Bearing Fault Diagnosis Based on Wavelet Package Transform and IPSO Optimized SVM. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 2758–2763.
29. Zhao, X.; Yan, X.; Yu, A.; Van Hentenryck, P. Prediction and Behavioral Analysis of Travel Mode Choice: A Comparison of Machine Learning and Logit Models. *Travel Behav. Soc.* **2020**, *20*, 22–35. [CrossRef]
30. Sarao, P. Machine learning and deep learning techniques on wireless networks. *Int. J. Eng. Res. Technol.* **2019**, *12*, 311–320.
31. Alzahrani, A.; Alenazi, M. Designing a Network Intrusion Detection System Based on Machine Learning for Software Defined Networks. *Future Internet* **2021**, *13*, 111. [CrossRef]
32. Alzahrani, A.; Alenazi, M. ML-IDSDN: Machine learning based intrusion detection system for software-defined network. *Concurrency and Computation: Practice and Experience* **2023**, *35*, e7438. [CrossRef]
33. Lin, Z.; Lin, M.; Wang, J.B.; de Cola, T.; Wang, J. Joint Beamforming and Power Allocation for Satellite-Terrestrial Integrated Networks With Non-Orthogonal Multiple Access. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 657–670. [CrossRef]
34. An, K.; Lin, M.; Ouyang, J.; Zhu, W.P. Secure Transmission in Cognitive Satellite Terrestrial Networks. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3025–3037. [CrossRef]
35. Lin, Z.; Niu, H.; An, K.; Wang, Y.; Zheng, G.; Chatzinotas, S.; Hu, Y. Refracting RIS-Aided Hybrid Satellite-Terrestrial Relay Networks: Joint Beamforming Design and Optimization. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3717–3724. [CrossRef]

36. Lin, Z.; An, K.; Niu, H.; Hu, Y.; Chatzinotas, S.; Zheng, G.; Wang, J. SLNR-based Secure Energy Efficient Beamforming in Multibeam Satellite Systems. *IEEE Trans. Aerosp. Electron. Syst.* 2022, 1–4. [CrossRef]
37. Lubans, D.R.; Plotnikoff, R.C.; Miller, A.; Scott, J.J.; Thompson, D.; Tudor-Locke, C. Using Pedometers for Measuring and Increasing Physical Activity in Children and Adolescents: The Next Step. *Am. J. Lifestyle Med.* **2015**, *9*, 418–427. [CrossRef]
38. Hishamuddin, M.N.F.; Hassan, M.F.; Tran, D.C.; Mokhtar, A.A. Improving Classification Accuracy of Scikit-learn Classifiers with Discrete Fuzzy Interval Values. In Proceedings of the 2020 International Conference on Computational Intelligence (ICCI), Bandar Seri Iskandar, Malaysia, 8–9 October 2020; pp. 163–166. [CrossRef]
39. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.