ISSN 2616-5775 Vol. 7, Issue 5: 28-36, DOI: 10.25236/AJCIS.2024.070504

A trajectory protection method based on differential privacy and semantic attributes

Langxi Liu[1,a,*]

[1]*School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, 411201, China*
[a]*465730162@qq.com*
[*]*Corresponding author*

Abstract: *With the popularity of smart wearable devices, Location-Based Services (LBS) have been widely applied. However, LBS generates thousands of trajectories, which may potentially leak personal information. To address such privacy concerns, it's often necessary to protect users' trajectory data. Existing differential privacy schemes commonly used for trajectory protection suffer from inefficiency and lack support for semantic attributes. To tackle these issues, this paper proposes a trajectory protection method based on differential privacy and semantic attributes (STrajGAN). Firstly, we employ differential privacy to process the raw data, enhancing efficiency while preserving privacy. Subsequently, the processed data is introduced into a Generative Adversarial Network (GAN) with Gated Recurrent Units (GRU), which generates similar trajectory data while considering the influence of semantic attributes of trajectory points, thus safeguarding user trajectory privacy. Experimental results demonstrate that, compared to other models, STrajGAN can provide better trajectory privacy protection by considering the influence of trajectory point semantic attributes.*

Keywords: *Trajectory Privacy Protection, Differential Privacy, GANs, Semantic attributes, GRU*

1. Introduction

With the rapid development of technologies such as the Internet of Things (IoT), 5G, and cloud computing, smart wearable devices like smartwatches and fitness bands have become increasingly popular. According to IDC forecasts, the global wearable market shipment is expected to reach 519 million units in 2023, with a projected year-on-year growth of 4.7% in 2024[1], indicating a significant market size. However, the widespread adoption of smart wearable devices also poses significant challenges to personal information protection. For instance, Location-Based Services (LBS) are frequently utilized on smart wearable devices[2], where users leave their location records while enjoying convenient services. These records constitute trajectory data, many of which are published and used for various applications such as urban planning, advertising, and store leasing[3]. However, the direct publication and application of trajectory data may lead to the leakage of user information, violating user privacy and posing security risks[4].

To achieve secure publication of trajectory data, existing research mainly adopts methods like obfuscation of locations. Differential privacy[5], due to its rigorous mathematical foundation, has been widely used for handling location information[6]. Hence, a large amount of recent research has focused on mechanisms that achieve differential privacy, making it one of the standards for privacy data publication. For example, Hien et al.[7] proposed a location protection method based on differential privacy and geographical broadcast mechanisms. However, such solutions are designed for centralized differential privacy, which significantly reduces efficiency when deployed on the user side. To enhance the efficiency of differential privacy schemes, subsequent research has proposed locally deployed differential privacy solutions. For instance, Xiong et al.[8] studied random response for continuous location sharing, while Cunningham et al.[9] investigated publishing Point of Interest (POI) sequences conforming to local differential privacy.

However, due to the high-density nature of location data, these types of differential privacy schemes still require significant efficiency reductions to achieve privacy protection[10], and smart wearable devices, designed for portability, typically have small form factors. Additionally, considering battery life and heating issues, they cannot be equipped with chips and memory with excessively high computing capabilities, making it impossible for them to perform complex and massive computations quickly.

ISSN 2616-5775 Vol. 7, Issue 5: 28-36, DOI: 10.25236/AJCIS.2024.070504

Furthermore, Erik et al. [11] found that since differential privacy typically adopts an all-or-nothing approach, requiring perturbation of all data, the random perturbation it imposes does not consider the influence of semantic attributes, resulting in unrealistic trajectories and providing limited protection utility. To address these issues, this paper proposes a trajectory protection solution for smart wearable devices, named STrajGAN. The model is based on differential privacy and generative adversarial networks, ensuring the effectiveness of the privacy protection mechanism while enhancing efficiency and resilience against reconstruction attacks. The contributions of this paper are summarized as follows:

(1) Introducing a label differential privacy method in the model, where users only need to perturb the labels to upload trajectories to the server, improving the efficiency of the trajectory privacy protection mechanism.

(2) Introducing a GAN module in the model, enabling it to synthesize trajectory data.

(3) Introducing a GRU module in the model to ensure that the model considers the influence of trajectory point semantic attributes when generating trajectories.

2. Methodology

2.1. Label Differential Privacy and Randomized Response

Differential Privacy (DP) proposes a privacy protection definition for data analysis processes, which aims to reduce the disclosure of personal privacy for any analysis of data and any further merging and processing based on the analysis results with other information.

Label Differential Privacy (LDP)[12]is a relaxed form of DP, where only the privacy of training labels needs to be protected. This privacy protection method assumes that the training data is non-sensitive and public, but the labels are sensitive and need to be kept confidential.

The above definitions theoretically guarantee that algorithms satisfy ε-local differential privacy. However, achieving ε-local differential privacy protection requires the intervention of data perturbation mechanisms. Currently, the Randomized Response[13] technique is the mainstream perturbation mechanism for local differential privacy protection. Its main idea is to use the uncertainty of responses to sensitive questions to protect the original data privacy.

2.2. Generative Adversarial Networks

Generative Adversarial Networks (GANs)[14] are a type of unsupervised deep learning model that consists of a generator model (G) and a discriminator model (D).

The purpose of Generative Adversarial Networks (GANs) is to learn the distribution of given data, represented as $x \sim p_{data}(x)$. To learn this distribution, we first define an input noise variable $P_z(z)$, which is then mapped to the data space, represented as $G(z;\theta_g)$, where G is a generative model consisting of multiple layers of perceptron parameterized by θ_g. The discriminator model, represented as $D(x;\theta_d)$, is used to determine whether the input data comes from the generative model or the training data, where D outputs x to determine if the input is from the training data. Finally, D is trained to accurately judge the source of the data, while G is trained to generate data that closely matches the distribution of the training data.

The training of GANs is a minimax problem, and the corresponding objective function is as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log(1 - D\left(G(z)\right))\right] \quad (1)$$

2.3. GRU

Gate Recurrent Unit (GRU) [15] is a type of Recurrent Neural Network (RNN). Similar to LSTM, it was introduced to address issues such as long-term memory and gradient vanishing during backpropagation. Compared to LSTM, GRU is easier to train and can achieve similar effectiveness. The unit structure of GRU is illustrated in Figure 1.

ISSN 2616-5775 Vol. 7, Issue 5: 28-36, DOI: 10.25236/AJCIS.2024.070504

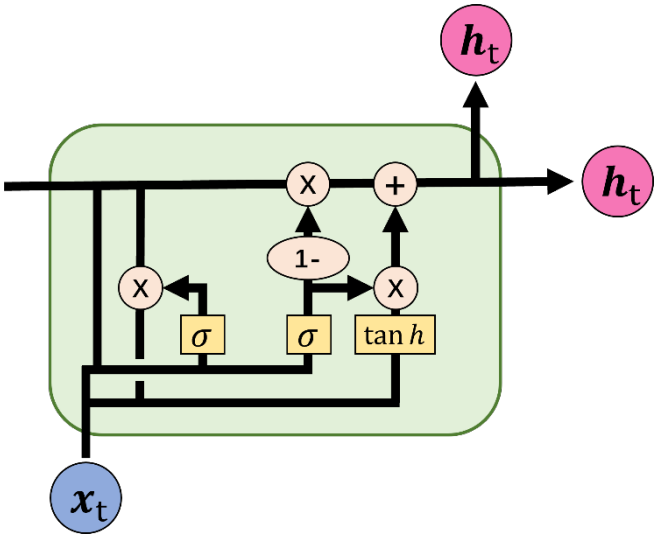

Figure 1: GRU unit.

3. STrajGAN Model

The STrajGAN model, as shown in Figure 2, consists of three main components: data preprocessing, trajectory generator G, and trajectory discriminator D. Data preprocessing encodes the positional coordinates, time, and category attributes of trajectory points, then flips a set of real labels with a certain probability and appends the flipped labels to the sampled real trajectories. The trajectory generator G synthesizes trajectories using random noise, while the trajectory discriminator D distinguishes between real and synthetic trajectory samples. The model is continuously trained on a specific dataset, aiming to make the trajectory discriminator D unable to determine whether the input trajectory is real or synthetic. Eventually, a mature trajectory generator G is obtained, capable of generating synthetic trajectories that meet the requirements of trajectory analysis tasks, possess a certain degree of resilience against attacks, and provide good support for trajectory semantics, thus further protecting user privacy.

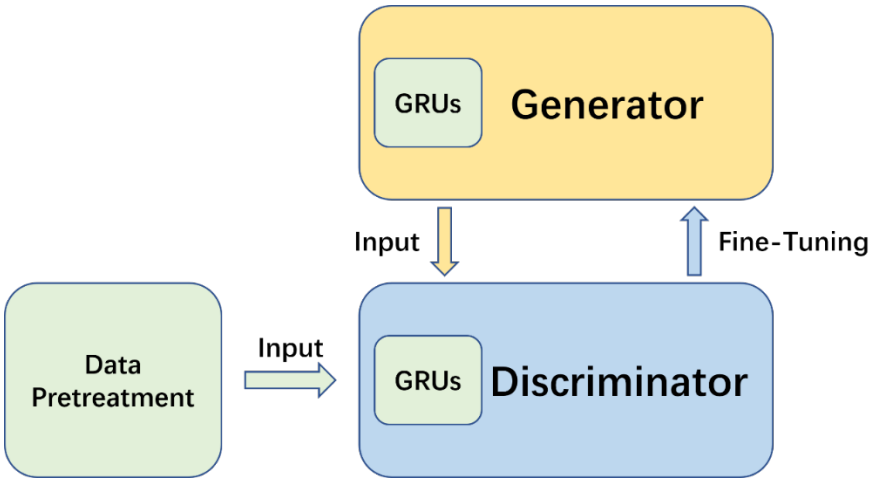

Figure 2: STrajGAN Model.

3.1. Data Pretreatment

Since semantic attributes need to be incorporated into trajectory points, it's necessary to convert semantic attribute data into effective numerical representations to facilitate model training. Therefore, the main content of the data preprocessing module is a Trajectory Point Encoding and Padding module. The primary purpose of this module is to convert the original trajectory data into a specific format suitable for input into the STrajGAN model.

The semantic attributes of encoded trajectory points include the following properties: trajectory ID, user ID, location, day count, hour, and category.

ISSN 2616-5775 Vol. 7, Issue 5: 28-36, DOI: 10.25236/AJCIS.2024.070504

Since trajectory ID and user ID are only used to indicate the user and the trajectory to which the point belongs, they do not need to be transformed throughout the process.

For the location attribute, we standardize all longitude and latitude coordinates based on the centroids of all trajectories in the dataset. This is done to obtain the deviations of longitude and latitude from the centroid. By doing this, the model can better learn spatial deviation patterns between different trajectory points. These deviation values will be used as numerical representations of trajectory points to construct spatial embeddings.

The attributes of day count, hour, and category are encoded using OneHotEncoder, which is a representation process commonly used in machine learning to encode attributes into multidimensional binary vectors, also known as dummy variables.

After encoding, since the number of trajectory points in each trajectory is not fixed, it's necessary to pad all trajectories to reach the length of the longest trajectory in the dataset. This helps speed up the training process. Specifically, in the context of this research, all attributes of the empty trajectory points in each trajectory are set to 0. This ensures that all trajectories have the same length as the longest trajectory in the dataset. These padded trajectory points do not affect the final training results.

3.2. Trajectory Generator G

To adapt to the working environment of trajectory generation, some adjustments have been made to the trajectory generator G compared to the original model. Firstly, a GRU network module has been added, which receives batches of noise vectors and passes them to the PointNet inside the generator. Leveraging its transformation invariance property and input-output alignment characteristics, the input random noise is projected into an m-dimensional feature vector x_t. Subsequently, x_t is passed through one linear layer to obtain m-dimensional output points, and through four linear layers with sigmoid activation functions to obtain output labels.

The Generator's workflow is as follows:

1) Process a batch of k-dimensional random noise vectors S_z using PointNet to obtain an m-dimensional feature vector x_t.

2) Perform linear feature extraction on x_t and apply regularization and ReLU activation to obtain the m-dimensional synthesized point $\hat{x}$.

3) Perform linear feature extraction on $\hat{x}$ and apply regularization and ReLU activation to obtain the m-dimensional feature vector x_{t1}.

4) Perform linear feature extraction on x_{t1} and apply regularization and ReLU activation to obtain the $\frac{1}{2}m$ m-dimensional feature vector x_{t2}.

5) Perform linear feature extraction on x_{t2} to obtain a 1-dimensional feature value x.

6) Apply the sigmoid function to x to obtain synthesized labels $\hat{l}$, which take values of 0 or 1.

7) Return the synthesized point $\hat{x}$ and synthesized labels $\hat{l}$.

3.3. Trajectory Discriminator D

To adapt to the label differential privacy mechanism, the trajectory discriminator D has undergone some adjustments compared to the original model. Firstly, a GRU network module has been added to receive batch noise vectors and pass them to the PointNet inside the generator. This balances the ability to learn point set representations between the generator and discriminator, thereby achieving uniform matching in the minimax game. Algorithm 3 describes the working process of the discriminator.

The Discriminator's workflow is as follows:

1) Process the batch input points and labels with PointNet, resulting in an m-dimensional feature vector x_t.

2) Perform linear feature extraction on x_t, followed by regularization and ReLU processing, yielding an m-dimensional feature vector x_{t1}.

3) Further perform linear feature extraction on x_{t1} to obtain a 1-dimensional feature value x.

ISSN 2616-5775 Vol. 7, Issue 5: 28-36, DOI: 10.25236/AJCIS.2024.070504

4) Apply the sigmoid function to x to obtain 0/1 (True/False).

5) Return 0/1.

4. Experiments and Analysis

4.1. Model Training and Dataset

The entire model training process is as follows:

1) Assign true labels to all real points in the dataset.

2) Flip the labels of all points with a probability of $q = \frac{1}{e^{\varepsilon}+1}$.

3) For each iteration from 1 to T:

4) Sample B points from all uploaded points.

5) Sample B random coordinate points with labels from the noise vector P_z.

6) Flip the labels of random coordinate points with a probability of $q = \frac{1}{e^{\varepsilon}+1}$.

7) Update the discriminator D after discriminating between real and fake.

8) Sample B random coordinate points with labels from the noise vector P_z.

9) Flip the labels of random coordinate points with a probability of $q = \frac{1}{e^{\varepsilon}+1}$.

10) Update the generator G after discriminating between real and fake.

11) Return G.

The experiment was implemented using the Python programming language on a Windows 10 operating system. The hardware setup includes an AMD Ryzen 9 5900X 12-Core Processor with a clock speed of 3.70 GHz, 32GB of memory, and an RTX 3090 GPU.

The experiment utilized the New York City trajectory dataset provided by May et al.[16], retaining only the following attributes: user ID, trajectory ID, coordinates, time, and category. Two-thirds of the trajectories were used for training the STrajGAN model, while the remaining one-third was used for testing.

The model was trained on the training set for 2000 epochs using several default training hyperparameters. After training completion, the generator was fed with trajectory data from the test set along with random noise to generate synthetic trajectory data. The generated data was then compared with two other commonly used location privacy protection methods, namely random perturbation and Gaussian geometric queries.

4.2. The privacy of the model's output trajectories

The Trajectory-User Linking (TUL)[17] is a classical trajectory analysis task commonly used to assess the privacy of trajectories. The TUL results indicate the model's ability to predict whether a trajectory belongs to the original user. A higher TUL score suggests a higher success rate in prediction, indicating poorer privacy of the model.

In this study, we employ the recent TUL algorithm, MARC, to perform the TUL task on both the test and synthesized data. Five common metrics are used to evaluate TUL accuracy in this task: ACC@1, ACC@5, Macro-P, Macro-R, and Macro-F1.

ACC@K measures the model's ability to select the correct label among the top K most probable label candidates. Macro-P and Macro-R represent the average precision and recall across all categories, respectively, while Macro-F1 is the harmonic mean of Macro-P and Macro-R.

Lower values of ACC@1, ACC@5, Macro-P, Macro-R, and Macro-F1 indicate higher accuracy in generating trajectories by the model.

The experimental results are depicted in Figures 3, 4, and 5. In these figures, the vertical axis represents the TUL accuracy, where lower values indicate better privacy of the generated trajectories.

ISSN 2616-5775 Vol. 7, Issue 5: 28-36, DOI: 10.25236/AJCIS.2024.070504

From the results, it can be observed that compared to random perturbation and Gaussian geometric queries, the trajectories generated by STrajGAN exhibit better privacy.

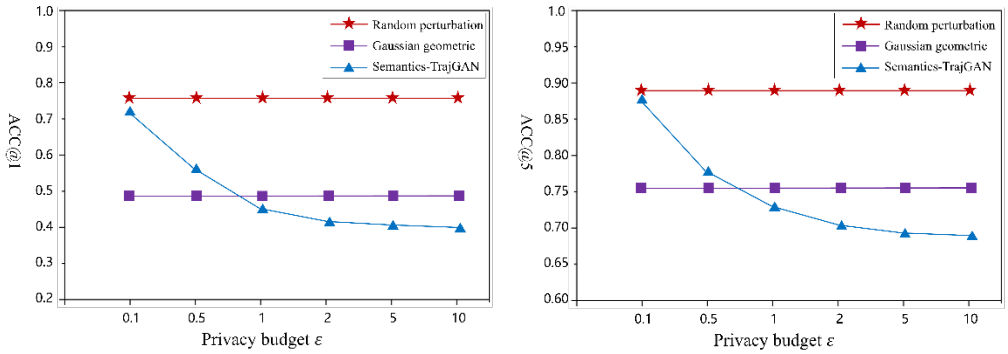

Figure 3: Results of different methods under the TUL task-ACC.

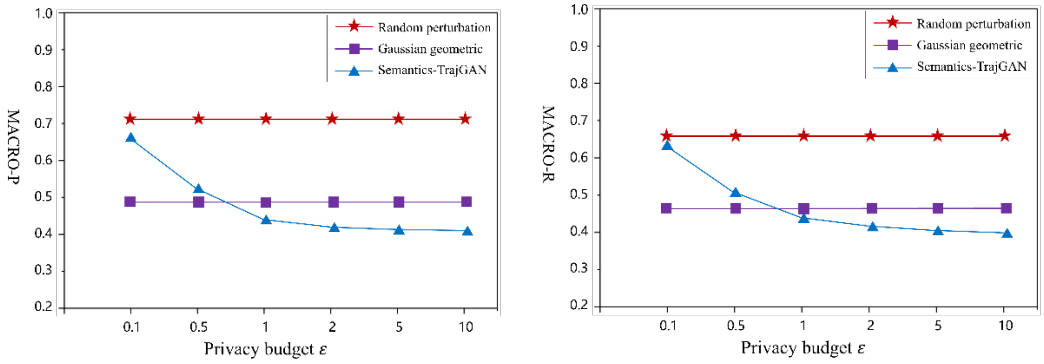

Figure 4: Results of different methods under the TUL task Macro-P and Macro-R.

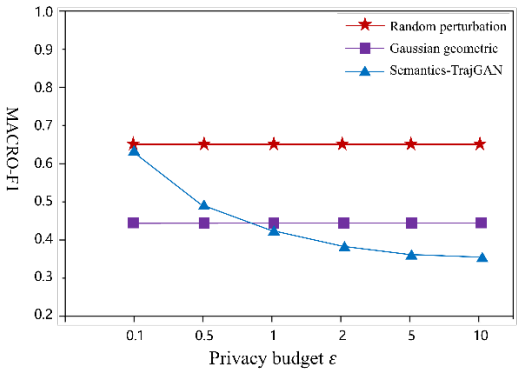

Figure 5: Results of different methods under the TUL task-MACRO-F1.

4.3. The authenticity of the model's output trajectories

The authenticity of the model's output trajectories consists of two components: temporal accuracy and spatial accuracy.

The temporal accuracy of the model's output trajectories can be determined using the Pearson product-moment correlation coefficient [18]. It measures the linear correlation between two sets of data variables X and Y by calculating the covariance of the variables divided by the product of their standard deviations.

In the context of this research, the temporal feature capability of different models can be compared by plotting the frequency of trajectory points appearing at different hours of the day and in different categories on the same day. Then, the overall Pearson correlation coefficient between the trajectories generated by different models and the original trajectories can be calculated. A higher Pearson correlation coefficient indicates better temporal similarity capability of the model, suggesting better temporal accuracy of the model's output trajectories.

ISSN 2616-5775 Vol. 7, Issue 5: 28-36, DOI: 10.25236/AJCIS.2024.070504

The spatial accuracy of the model's output trajectories can be determined using the Jaccard index[19], also known as the intersection over union. It is a statistical measure used to compare the similarity and diversity of sample sets. The Jaccard index measures the similarity between two sets by comparing the size of their intersection to the size of their union.

In the context of this research, a higher Jaccard index between two trajectory segments indicates higher spatial similarity, suggesting better spatial accuracy of the model's output trajectories.

The results obtained by plotting the frequency of trajectory points appearing at different hours of the day and different categories on the same day, and calculating the overall Pearson correlation coefficient between the trajectories generated by different models and the original trajectories, are shown in Figure 6. A higher value in the table indicates better temporal similarity performance of the method.

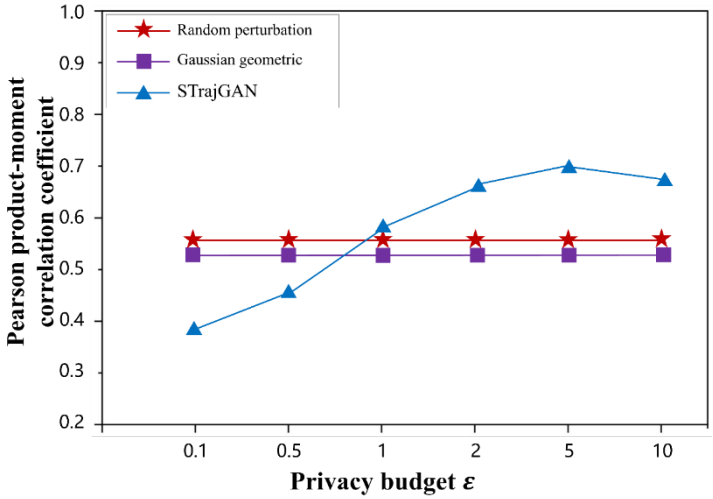

Figure 6: Comparison of Pearson correlation coefficients between different methods.

Based on the results above, it can be concluded that when the privacy budget $\varepsilon \geq 1$, compared to random perturbation and Gaussian geometric queries, the STrajGAN model exhibits better temporal accuracy, indicating higher authenticity of the generated trajectories.

The spatial accuracy of model output trajectories can be determined using the Jaccard index[18]. The Jaccard index, also known as the intersection over union, is a statistical measure used to compare the similarity and diversity of sample sets. It quantifies the similarity of finite sample sets and is defined as the ratio of the size of the intersection of two sets to the size of their union.

In the specific context of this study, a higher Jaccard index between two trajectory segments indicates greater spatial similarity, thus implying better spatial accuracy of the model output trajectories.

The Jaccard index between the original trajectories and synthesized trajectories is calculated, and the comparison between different methods is shown in Figure 7.

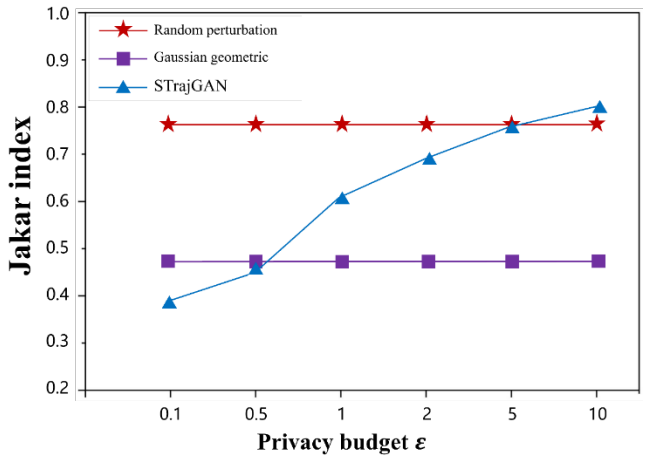

Figure 7: Comparison of the Jakar index between different methods.

ISSN 2616-5775 Vol. 7, Issue 5: 28-36, DOI: 10.25236/AJCIS.2024.070504

Based on the results above, it can be observed that when the privacy budget $\epsilon \leq 5$, random perturbation exhibits the smallest Jaccard index. This is because this method has a relatively low impact on the spatial dimension of trajectories. Although random perturbation can effectively maintain spatial similarity, it does so at the expense of sacrificing privacy. However, when the privacy budget $\epsilon \geq 1$, STrajGAN outperforms Gaussian Geometric Perturbation in terms of the TUL metric, indicating better privacy. Combining experimental analysis, it can be concluded that the STrajGAN model achieves a better balance between authenticity and privacy.

5. Conclusions

To effectively enhance the efficiency of trajectory privacy protection mechanisms and increase support for semantic attributes of trajectory points, this paper proposes the STrajGAN model. The model introduces the label differential privacy method to improve the operational efficiency of trajectory privacy protection mechanisms. Additionally, by incorporating a GAN with a GRU module to generate trajectories, the model considers the influence of semantic attributes of trajectory points, thereby achieving the goal of privacy protection. The trajectory generation of the model relies on the training of the dataset and exhibits some regional characteristics. In future work, we plan to extend the model to larger-scale trajectory datasets, generate customized variable-length synthetic trajectory data, explore potential privacy attacks and defense strategies, and assess the effectiveness and practicality of the model in other trajectory data mining and analysis tasks.

References

[1] Fernández-Caramés, T. M., & Fraga-Lamas, P. (2018). Towards the Internet of smart clothing: A review on IoT wearables and garments for creating intelligent connected e-textiles. Electronics, 7(12), 405.

[2] Mendes, R., & Vilela, J. P. (2017). Privacy-preserving data mining: methods, metrics, and applications. IEEE Access, 5, 10562-10582.

[3] Yang, Z., Wang, R., Wu, D., Wang, H., Song, H., & Ma, X. (2021). Local trajectory privacy protection in 5G enabled industrial intelligent logistics. IEEE Transactions on Industrial Informatics, 18(4), 2868-2876.

[4] Jin, F., Hua, W., Francia, M., Chao, P., Orlowska, M., & Zhou, X. (2022). A survey and experimental study on privacy-preserving trajectory data publishing. IEEE Transactions on Knowledge and Data Engineering.

[5] Dwork, C. (2006, July). Differential privacy. In International colloquium on automata, languages, and programming (pp. 1-12). Berlin, Heidelberg: Springer Berlin Heidelberg.

[6] Andrés, M. E., Bordenabe, N. E., Chatzikokolakis, K., & Palamidessi, C. (2013, November). Geo-indistinguishability: Differential privacy for location-based systems. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (pp. 901-914).

[7] To, H., Ghinita, G., Fan, L., & Shahabi, C. (2016). Differentially private location protection for worker datasets in spatial crowdsourcing. IEEE Transactions on Mobile Computing, 16(4), 934-949.

[8] Xiong, X., Liu, S., Li, D., Wang, J., & Niu, X. (2019). Locally differentially private continuous location sharing with randomized response. International Journal of Distributed Sensor Networks, 15(8), 1550147719870379.

[9] Cunningham, T., Cormode, G., Ferhatosmanoglu, H., & Srivastava, D. (2021). Real-world trajectory sharing with local differential privacy. arXiv preprint arXiv:2108.02084.

[10] Rao, J., Gao, S., Kang, Y., & Huang, Q. (2020). Lstm-trajgan: A deep learning approach to trajectory privacy protection. arXiv preprint arXiv:2006.10521..

[11] Buchholz, E., Abuadbba, A., Wang, S., Nepal, S., & Kanhere, S. S. (2022, December). Reconstruction attack on differential private trajectory protection mechanisms. In Proceedings of the 38th Annual Computer Security Applications Conference (pp. 279-292).

[12] Ghazi, B., Golowich, N., Kumar, R., Manurangsi, P., & Zhang, C. (2021). Deep learning with label differential privacy. Advances in neural information processing systems, 34, 27131-27145.

[13] Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association, 60(309), 63-69.

[14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.

[15] Dey, R., & Salem, F. M. (2017, August). Gate-variants of gated recurrent unit (GRU) neural networks. In 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS) (pp.

ISSN 2616-5775 Vol. 7, Issue 5: 28-36, DOI: 10.25236/AJCIS.2024.070504

1597-1600). IEEE.
[16] May Petry, Lucas, et al. "MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings." International Journal of Geographical Information Science 34.7 (2020): 1428-1450.
[17] Zhou, F., Gao, Q., Trajcevski, G., Zhang, K., Zhong, T., & Zhang, F. (2018, July). Trajectory-User Linking via Variational AutoEncoder. In IJCAI (pp. 3212-3218).
[18] Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation coefficients: appropriate use and interpretation. Anesthesia & analgesia, 126(5), 1763-1768.
[19] Werner, M., Schauer, L., & Scharf, A. (2014, May). Reliable trajectory classification using Wi-Fi signal strength in indoor scenarios. In 2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014 (pp. 663-670). IEEE.