Do We Need Real Data? – Testing and Training Algorithms with Artificial Geolocation Data

Jan Kaiser[1] Kai Bavendiek[2] Sibylle Schupp[3]

Abstract: As big data becomes increasingly important, so do algorithms that operate on geolocation data. Privacy requirements and the cost of collecting large sets of geolocation data, however, make it difficult to test those algorithms with real data. Artificially generated data sets therefore present an appealing alternative. This paper explores the use of two types of neural networks as generators of geolocation data and introduces a method based on the Turing Test to determine whether generated geolocation data is indistinguishable from real data. In an extensive evaluation we apply the method to data generated by our own implementation of neural networks as well as the widely used BerlinMOD generator on the one hand, the four most prominent data sets of real geolocation data covering at total of 65 million records on the other hand. The experiments show that in eleven of twelve cases artificial data sets can be told from real ones. We conclude that, at present, the generators we tested provide no safe replacement for real data.

Keywords: geolocation data; artificial data; data generation; neural networks generators; data quality

1 Introduction

With increasing interest in *Big Data* in recent years, interest in geolocation data – collections of coordinates, such as latitudes and longitudes – has increased as well. Geolocation data may find use in areas, such as traffic planning, recommender systems, market research, map creation, location privacy, and autonomous driving. As do algorithms that are capable of analysing geolocation data and drawing meaningful conclusions from it. In order to develop and test such algorithms, developers and researchers, alike, need access to geolocation data sets. Unfortunately, only a very limited amount of real geolocation data sets is freely available because of privacy requirements and other difficulties associated with their collection. Real geolocation data sets are also not very flexible, as their sizes, covered areas, logged entities, etc. cannot easily be changed to meet anyone's needs.

Artificially generated data sets promise to solve the above problems. In theory, it is easily possible to generate data sets off various sizes, for various locations, with various types of entities, all of which could be adjusted to produce exactly the kind of data set a given

[1] Hamburg University of Technology; Hamburg, Germany; jan.kaiser@tuhh.de
[2] Hamburg University of Technology; Hamburg, Germany; kai.bavendiek@tuhh.de
[3] Hamburg University of Technology; Hamburg, Germany; schupp@tuhh.de

doi:10.18420/inf2019_25

application requires. However, the question remains how to generate realistic geolocation data. The other question is how to determine whether generated data is realistic. We propose an approach for determining the realism of geolocation data based on the Turing Test. We use this approach to test the BerlinMOD generator and two Artificial Neural Network generators on the realism of the data they produce.

This paper gives a brief outline of existing work in the field in Section 2. In Section 3, we present two novel methods of generating geolocation data using Artificial Neural Networks. We evaluate[4] both methods and the commonly used BerlinMOD generator using our approach to assessing the quality of generated data based on the Turing Test in Section 4. Lastly in Section 5, we conclude that the presented generators are not fit for use in research. We also conclude that our Turing-based approach for determining the realism of data delivers good results. We finish by making proposals for future work.

2 Related Work

From the previous decade, three algorithmic generators of geolocation data – BerlinMOD [DBG09], Brinkhoff's generator [Br03], and SUMO [Kr12] – have been widely accepted as sources of data for testing algorithms. SUMO and Brinkhoff's generator produce what research refers to as short-term data, i.e. observations of otherwise anonymous entities while they move from one place to another. BerlinMOD produces so-called long-term data in which entities are observed for a longer timeframe, regardless of whether they are currently moving between places or remain stationary. We are not aware of any further developments in the field of geolocation data generators in recent years.

As other means of generating data, Artificial Neural Networks have recently gained in popularity. In the area of sequenced data Recurrent Neural Networks have demonstrated impressive results producing authentic sequences, for example of text [Go17, Te17]. The text generation approach by Gorner [Go17] serves as the basis for our approach to generating sequences of geolocation data. A different type of Neural Network, Generative Adversarial Networks, have been shown to work well for generating realistic images [RMC15, Zh17]. In future work a similar approach could be used to plot realistic geolocation data on an image of a map. Both approaches have also been crossed, for example in the implementation of *C-RNN-GAN* [Mo16], which employs an architecture of recurrent Generative Adversarial Networks to produce realistic music. A similar approach based on recurrent Generative Adversarial Network is pursued in this paper.

Several metrics for comparing data sets and their quality exist, for example in the area of location privacy [MBT11]. However we are not aware of any works that devise means of evaluating the quality of artificial data in a general scenario.

[4] Implementations available at `https://github.com/LordHelmchen324/real-vs-synthetic-geospatial`

While our work explores the possibilities of generating raw data that is indistinguishable from real recorded data, there has been other work investigating formal models to describe real-world patterns found in geolocation data [Zh16]. Rhee et al., for example, have found that the movement of people in a city can be modelled mathematically as Levy walks [Rh11]. Models like this hold great potential for understanding the behaviour of people in a city, and they could also find use in generators of geolocation data.

3 Generators

Two different kinds of generators are distinguished for this paper: algorithmic generators and Artificial Neural Networks. Algorithmic generators use assumptions about the real-world behaviour of entities to produce realistic spatio-temporal data. The assumptions commonly include concepts such as starts and destinations of trips, commuting, and the dependency on a road network. Popular algorithmic generators in research are BerlinMOD, Brinkhoff's generator, and SUMO. As the latter two do not model entities over more than the duration of a single trip, the nature of the data they produce is not the same as that of real data. For the remainder of the paper, we therefore focus on BerlinMOD as the algorithmic generator.

As a second and novel approach to generating geolocation data, Artificial Neural Networks (ANN) are considered in this paper. ANNs have grown increasingly popular over the past years and they have been successfully applied to various kinds of problems. Two types of ANN are evaluated for generating geolocation data in this paper: Recurrent Neural Neural Networks because of the demonstrated ability to generated sequence data and Generative Adversarial Networks because they were specifically developed to generate data that is indistinguishable from real data. The particular architectures developed for this paper are introduced in sections 3.1 and 3.2, respectively.

3.1 Recurrent Neural Networks

The particular Recurrent Neural Networks (RNN) architecture used in this paper builds on an architecture for text generation presented by Martin Gorner of *Google* [Go17, Te17]. The architecture consists of a stack of Gated Recurrent Unit (GRU) cells, followed by time-distributed fully-connected layers (see Figure 1). Time-distributed here means that there are separate fully-connected layers following each of the GRU cells at each time step. Therefore, the output of the network is a sequence of 3-tuples. Different numbers of stacked GRU cells, fully-connected layers, and both their sizes may be chosen.

The given RNN architecture requires the training data to be fed into it in a way that respects that training is run in batches and on sequences of geolocation data, that a validation is run at the end of each epoch of training, and that the RNN's state remains persistent. For the specifics and a formal description of this formatting of the data please refer to the

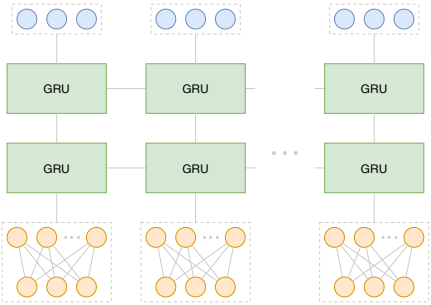


Fig. 1: RNN architecture

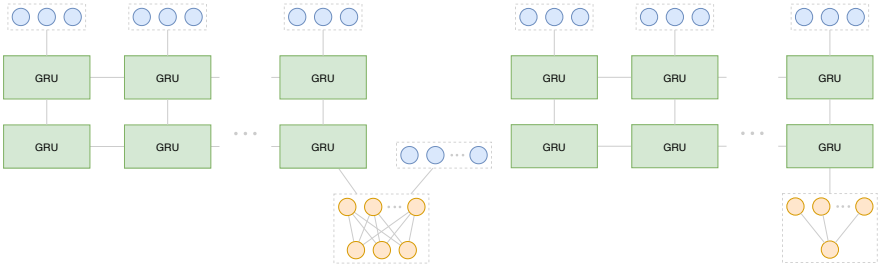
thesis [Ka19] this paper is based on. This thesis also goes into more details on the exact parameters, such as loss functions and optimisers, we used.

3.2 Generative Adversarial Networks

Generative Adversarial Networks (GAN) [Go14] use an *adversarial model* – a setup where two ANNs compete against each other in becoming very good at their respective tasks – in order to train an ANN that can generate data that, in the ideal scenario, is new but indistinguishable from real data. The general idea is a setup of two ANNs, a *generator* and a *discriminator*. The discriminator's task is to tell generated fake data from real data, and the generator's task is to generate fake data based on a random seed that the discriminator cannot tell apart from real data.

The particular GAN architecture used in this paper consists of both a recurrent generator and a recurrent discriminator. RNN were chosen for the GAN architecture as well because the task is to generate sequences. The generator uses a sequence of stacked GRU cells where only the output of the last stack is concatenated with a random input. The concatenated vectors are fed forward to a stack of fully-connected layers, the last of which must always have three neurons. The height of both the GRU cell stack and the fully-connected stack may be chosen freely. The generator's architecture is illustrated in Figure 2a. A similar architecture is used for the discriminator. Again, a stacked sequence of GRU cells is used and only the last cell's output is fed forward into a stack of fully-connected layers. The last of the fully-connected layers of the discriminator must have only a single neuron. See Figure 2b for an illustration of the discriminator network used in this paper.

For training the GAN, the real data needs to be provided in the correct format to ensure the coherence of the networks' states. For details on the particular formatting of the data, including formal descriptions, we, again, refer to the thesis [Ka19] this paper is based on. The latter thesis also gives information on the training process, loss functions, and optimisers we used.



(a) Generator (b) Discriminator

Fig. 2: RNN architectures used in the GAN setup

4 Interrogations

The purpose of generated geolocation data is to be used instead of real recorded geolocation data when no suitable real data is available. Ideally, generated data should therefore be indistinguishable from actual real data, i.e. it should be fully realistic. However, defining realism for geolocation data is not trivial. A multitude of complex factors influences real geolocation data and the general definition of realism on geolocation data is equally ill defined as the concept of intelligence. To the best of our knowledge, there exists no formal way of fully grasping either of these matters in its entirety.

The determination of intelligence has famously been approached using the Turing Test [Tu50]. The idea is that a machine is considered intelligent, if an interrogator cannot point out the machine after asking questions in a conversation with both a human and the machine. A common setup for the test is to measure the proportion of interrogators who were unable to tell which conversational partner is the machine. Thereby, this approach circumvents the need for a thorough definition of intelligence. Setups similar to the Turing Test have been proven to work well in other fields such as distinguishing paintings created by GAN architectures from those created by human artists [El17].

Mapping the Turing Test approach to geolocation data sets, the interrogator is given a real and a generated data set – not knowing which of the two is which – and the task to figure out which of the two data sets is generated. In place of a conversation, the interrogator may perform different kinds of data analysis, referred to as *questions*, on both data sets. While we as the interrogators do know which data sets are real or fake, we attempt to recreate the way an uninformed interrogator would argue.

4.1 Real Data Sets

For the purpose of a Turing Test-like interrogation we compared four real recorded data sets to data sets generated by BerlinMOD and our ANN generators. We used the same

data sets to train the ANN generators. All real data sets and their properties are given in Table 1. The chosen data sets are to our knowledge the largest freely available geolocation data sets currently used in research. The Mobile Data Challenge [Ki10, La12] (MDC) and GeoLife [ZXM09, ZXM08, ZXM10] data sets are both based on people. Because data sets based on people are rare, we also include the Cabspotting [PSDG09] and T-Drive [Yu11, Yu10] data sets, which are taxi-based.

	MDC	**GeoLife**	**Cabspotting**	**T-Drive**
Location	Lake Geneva region	Beijing & world	San Francisco	Beijing
Entities	185 users	182 users	536 taxis	10,357 taxis
Duration	1.5 years	5 years	30 days	6 days
Records	13,678,618	24,876,978	11,219,955	17,662,984

Tab. 1: Properties of the real data sets

4.2 Data Set Generation Setup

We generated the artificial data sets to have the same basic properties, e.g. city, number of users, and duration, as their real counterparts. For generating data using BerlinMOD we used data from OpenStreetMap [Op17] as the input for the road network.

	rnn	***gan_gen***	***gan_dis***
GRU	32	32	16
Fully-connected	3	3	1

Tab. 2: ANN configurations with each number in the GRU and Fully-connected row representing a layer and giving its size

All ANNs were trained on the respective real data sets. We tried different configurations of the presented architectures, but all produced similar results. We therefore only present the results of the configurations from Table 2 using a batch size of 64, a sequence length of 100, *mean absolute error* (RNN) and *binary cross entropy* (GAN) loss functions, and the *Adam* optimiser [KB14]. At the start of the generation we use *Kernel Density Estimation* (KDE) [Ro56, Pa62] to create distributions from the real data sets, from which we sample latitude and longitude of the initial records as well as relative start and end times of each user's data. The start position KDE is fitted on all positions in the full original data set using a bandwidth of 0.00003. The KDE for the relative start time is fitted only on a reduced data set of 50 users and 1 week (see Section 4.3). A bandwidth of 0.03 is used for fitting the KDEs to relative start times and durations. Sampling of relative start times and durations is repeated until the relative start time lies within the interval $[0, 1[$ and the sum of relative start time and duration is within the interval $]0, 1]$, in order to ensure that only data within the given week is generated. We then convert relative times to real times before feeding them to the ANNs to generate new records starting from the initial record for as long as the generated records' timestamps are within the sampled duration. Because the ANNs generate time deltas δt instead of absolute times, we constrain each time delta to be at least 1 second

for RNN and 90 seconds for GAN in order to make sure that the generation eventually terminates. A larger lower bound is needed on the GAN generation because the trained GANs tend to produce small δt.

4.3 Interrogation Setup

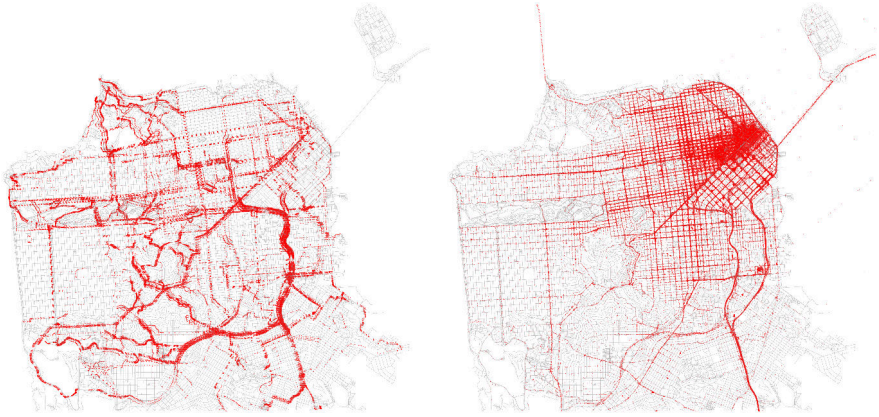
For this evaluation, all data sets – real and generated ones – are limited to 50 randomly chosen users and the time frame of the busiest week (Monday to Sunday for all but T-Drive) in order to keep computation times manageable. An exception is made on the T-Drive data set. Because this data set does not cover a full Monday to Sunday week, its entire time span is considered.

For the sake of brevity, we present only 4 of the 12 interrogations we conducted, two on data sets generated by BerlinMOD and one on each of the ANNs we presented (see [Ka19] for the complete observations of all interrogations). We assume that the intent is to generate a data set of the same nature – people- or taxi-based – as the real counterpart and that the interrogator is aware of this information. We also devised four questions our interrogator will ask. However, our interrogator is free to make a final decision after any number of questions.

The first question we devised is a simple plot of all records as red dots on a map of the respective city's streets, allowing the interrogator to quickly spot irregular behaviour. We call this a *map overview*. The second question our interrogator can ask is a histogram of the number of records over bins of all 168 hours of the relevant week. We hope to observe behaviour such as commuting in this histogram we refer to as *traffic*. Our third question we call *speeds*. This question constitutes a histogram of the speeds measured between two consecutive records for speeds from 0 to 150 km/h. We hope to be able to spot unlikely speeds in a data set this way. As the fourth and last question we devised, we plot the records of a single user coloured according to their time of recording on a map. This question may give clearer insights into user behaviour than the map overview question. We refer to this question as *single user*.

4.4 BerlinMOD versus Cabspotting

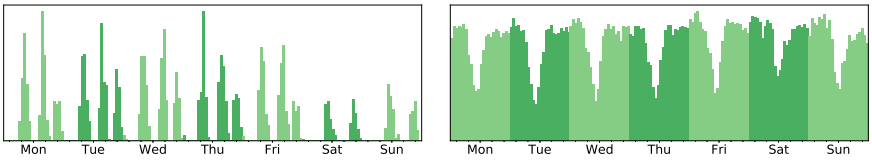
Starting with the data set generated by BerlinMOD based on the Cabspotting data set, we first consider a map overview (see figures 3a and 3b). Both data sets mostly follow the street network. Only a few of the records are not on the streets, which can be explained as either disturbance caused by the GPS positioning systems or users entering a building. The records which leave the streets display different behaviours in both data sets. In data set 2 those records are placed arbitrarily, with some even being in the sea. In data set 1, on the other hand, off-street records remain close to the streets. This may be a soft argument to consider



(a) BerlinMOD (data set 1) (b) Real (data set 2)

Fig. 3: Map overview of data sets on San Francisco, USA

data set 1 to be artificial as it gives the impression of being perfect data that was simply disturbed slightly after it was generated. Furthermore, in data set 2 there is a single area, where records are concentrated densely, whereas data set 1 is very focused on the street network only. Given that Cabspotting is a taxi-based data set, it can be argued that data set 1 is more realistic because taxis are not likely to leave the streets. However, the argument could also be made that most cities have a main area of interest, where traffic accumulates, and that such an area can only be observed in data set 2. Going by this question, no clear decision can be made regarding which of data sets is the artificial one.

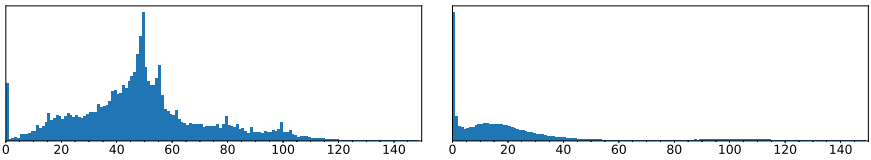


(a) BerlinMOD (data set 1) (b) Real (data set 2)

Fig. 4: Number of records during each hour in data sets on San Francisco, USA

Next, we ask the question of traffic (see figures 4a and 4b). data set 2 shows that there is less traffic during the midday on each day. However, one would intuitively expect there to be the least amount of traffic at night. Furthermore, all days of the week show the same behaviour and, except for Sunday, similar amounts of traffic. Intuitively, one would expect the days of the weekend to behave differently from work days and be less busy altogether. The first immediately visible property of data set 1 is that there are periods of no traffic. This may be possible in data sets of few users, but one would intuit that in 50 taxis, there is always at least one moving. Otherwise, traffic seems to follow a commuting pattern. Taxis travel in the morning hours and in the afternoon, and smaller spikes occur in the evenings. On

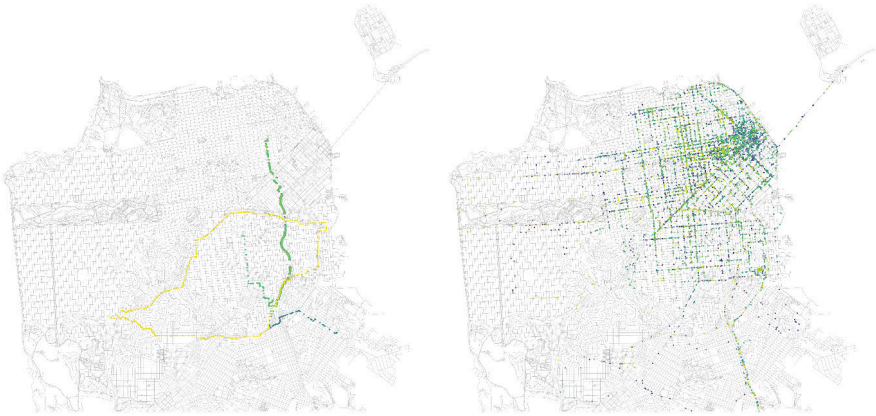
the weekend, taxis log fewer records than on weekdays and at different times of the day. Assuming that taxis are used by people for their commute and travels to free times activities, these patterns make sense. Based on these observations, it is hard to make out the artificial data set. Only the times of no traffic in data set 1 make that data set seem slightly strange and possibly artificial.



(a) BerlinMOD (data set 1) (b) Real (data set 2)

Fig. 5: Histograms of user's speeds in data sets on San Francisco, USA

So far a strong decision is not possible, so we move on to the question of speeds (see figures 5a and 5b). In data set 2 users move at less than 40 km/h most of the time, which seems about right for taxis in city traffic, but there is also a smaller cluster surrounding 100 km/h, presumably caused by taxis travelling on motorways that cut through the city. There is also a spike near 0 km/h which we argue could be caused by taxis waiting for passengers. In data set 1 speeds peak at around 50 km/h – a common speed in cities – and drop towards 0 km/h and 120 km/h, meaning there are taxis that travel slowly, for example in traffic jams or on small streets, and some that travel fast, for example on motorways. The smaller spike at 0 km/h can be explained as taxis waiting for passengers. Summarising the question of speeds, neither of the two data sets can be pointed out as generated.



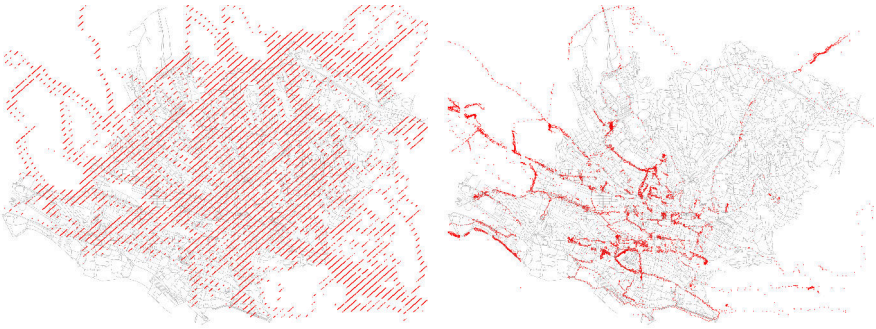
(a) BerlinMOD (data set 1) (b) Real (data set 2)

Fig. 6: Data of single users on San Francisco, USA

Because the previous question did not yield a clear indication, we decide to look at single trajectories (see figures 6a and 6b). An immediately obvious difference between both data sets is that the user in data set 2 went to many different places, whereas the user from

data set 1 looks orderly and only visited a handful of places mostly travelling along the same routes. In a people-based data set the latter behaviour would make sense with people frequently commuting between their homes and work places on the same routes, but given that Cabspotting is a taxi-based data set, this last question gives another soft indication towards data set 1. As taxis transport many different passengers to and from many different locations, they are likely to visit many different places over the course of one week.

4.5 BerlinMOD versus MDC



(a) BerlinMOD (data set 3) (b) Real (data set 4)

Fig. 7: Map overview of data sets on Lausanne, Switzerland

Comparing a data set generated by BerlinMOD to the MDC data set, we start with a map overview (see figures 7a and 7b). In both data sets, users follow the roads most of the time and both data sets appear to have a certain degree of disturbance in them, something that is natural to GPS positioning systems. However, the disturbance in data set 3 occurs along the same axis for all records, which we do not expect to observe in real data. We therefore conclude that data set 3 is generated and do not proceed to the next question.

4.6 RNN versus Cabspotting

Comparing the data set generated by our RNN architecture to the Cabspotting data set, we once again first ask the map overview question (see figures 8a and 4b). For the reasons given in Section 4.4, data set 2 does not give any indicators that it is artificial. The data in data set 5, however, appears to evolve only around a single point on the map, and when moving away or towards this point the users appear to do so with absolute disregard for the streets. This feature clearly does not resemble what one might expect a real data set to look like, hence we conclude already that data set 5 was artificially generated. We ask no further questions.

(a) RNN (data set 5) (b) GAN (data set 6)

Fig. 8: Map overview of data sets on San Francisco, USA

4.7 GAN versus Cabspotting

Moving on to the data set generated by our GAN architecture based on the Cabspotting data set, we start with the map overview question (see figures 8b and 4b). We immediately decide that data set 6 is artificial because the features of data set 2 can be argued to be realistic as already done in Section 4.4 and data set 6 displays a similar strange behaviour as data set 5 in Section 4.6. Taxis in data set 6 do not move along the streets but instead follow a circular path into a small area, where they remain mostly stationary. This area is shared among all taxis. We would not expect such behaviour to be present in real geolocation data and therefore decide that data set 6 is artificial without asking further questions.

4.8 Conclusion of Experiments

From the above experiments as well as the other experiments we conducted, we conclude that in only 1 out of 12 cases a generator was able to fool the interrogator. In this particular case of BerlinMOD generating a data set based on GeoLife, the sparse nature of the real GeoLife data set made the latter seem artificial in comparison to the dense data set generated by BerlinMOD. Other data sets generated by BerlinMOD also did well. Most times only details revealed their artificial nature after asking all four of our questions. Data sets generated by either of our ANN architectures did badly. In every one of the interrogations conducted on them, they could be pointed out as artificial immediately after asking the very first question of a map overview. On all four surveyed real data sets we observed the problems to be the same. Road networks appear to be ignored by the ANN generators in all cases, always producing similar circular movement which accumulates on one or a few positions on the map.

5 Conclusion

Concluding this study, we find that data generated by any of the three surveyed generators cannot replace real data. In all data we generated there remains enough evidence within recorded and generated data sets to tell them apart, which means that some of their properties regarding realism, which potentially have an influence on a prospective use case, do differ. Of the presented generators, BerlinMOD came the closest to producing a realistic data set. In most questions of the interrogation, data generated by BerlinMOD looked realistic at first glance, often only giving away its generated nature through detailed features of the data. Our generators based on ANNs did not do well. The data they generate is profoundly unrealistic and not at all suitable for testing algorithms. We therefore advise against using either one for testing and training algorithms that work on geolocation data. When the results are not required to be fully reliable, data generated by BerlinMOD may be used.

One might argue, as we have seen in the experiments ourselves, that a lack of quality in the real data can cause this method to fail as features of realistic data, such as sparsity or outliers, can fool interrogators depending on their expertise. However, because such effects are commonplace in real data, all discrimination methods do have to deal with the effects. Our Turing-based method of discriminating between real and generated data sets has otherwise proven to work well. We showed that our method functions well as a means of evaluating the realism of generated data while eliminating the need for a thorough definition of realism. The method we presented has potential for use on other kinds of data as it is domain-independent.

Opportunities for future work lie in devising further questions to ask on a data set. Such questions may be of use for evaluating more powerful generators. As an alternative to the presented approach based on the Turing Test, future work might also consider using ANNs as the means of evaluating the realism of generated geolocation data sets. Furthermore, our method based on the Turing Test may be applied to other types of data, of which in the age of *Big Data* there are certainly many. Training more capable ANN generators for geolocation data is also an intriguing idea for future work. ANNs could potentially also be used in the field of Trajectory Anonymisation by training them on a single real data set and then using them to produce a new data set with the same general features, but with users that behave slightly differently from the real users, thereby protecting the privacy of the real users.

Acknowledgements

(Portions of) the research in this paper used the MDC Database made available by Idiap Research Institute, Switzerland and owned by Nokia. Map data copyrighted OpenStreetMap contributors and available from `https://www.openstreetmap.org`.

References

[Br03] Brinkhoff, T.: Generating traffic data. IEEE Data Engineering Bulletin, 26:19–25, 2003.

[DBG09] Düntgen, C.; Behr, T.; Güting, R. H.: BerlinMOD: A benchmark for moving object databases. The VLDB Journal, 18:1335–1368, 2009.

[El17] Elgammal, A.; Liu, B.; Elhoseiny, M.; Mazzone, M.: CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms. arXiv preprint, arXiv:1706.07068, 2017.

[Go14] Goodfellow, I. et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680, 2014.

[Go17] Gorner, M.: TensorFlow and Deep Learning without a PhD. https://www.youtube.com/watch?v=fTUwdXUFfI8 (Accessed: 12th April 2019), 2017.

[Ka19] Kaiser, J.: A study of generated versus recorded geolocation data. Research project thesis, Hamburg University of Technology, 2019. Available at https://www.sts.tuhh.de/pw-and-m-theses/2019/kaiser19.pdf.

[KB14] Kingma, D. P.; Ba, J.: Adam: A Method for stochastic optimization. arXiv preprint, arXiv:1412.6980, 2014.

[Ki10] Kiukkonen, N.; Blom, J.; Dousse, O.; Gatica-Perez, D.; Laurila, J.: Towards rich mobile phone datasets: Lausanne data collection campaign. In: Proceedings of the ACM Int. Conf. on Pervasive Services. 2010.

[Kr12] Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L.: Recent development and applications of SUMO - Simulation of Urban MObility. Int. Journal On Advances in Systems and Measurements, 5:128–138, 2012.

[La12] Laurila, J. et al.: The mobile data challenge: Big data for mobile computing research. In: Proceedings of the Workshop on the Nokia Mobile Data Challenge, in Conjunction with the 10th Int. Conf. on Pervasive Computing. pp. 1–8, 2012.

[MBT11] Martinez-Bea, S.; Torra, V.: Trajectory anonymization from a time series perspective. In: IEEE Int. Conf. on Fuzzy Systems. pp. 401–408, 2011.

[Mo16] Mogren, O.: C-RNN-GAN: Continuous recurrent neural networks with adversarial training. arXiv preprint, arXiv:1611.09904, 2016.

[Op17] OpenStreetMap contributors: Planet dump retrieved from https://planet.osm.org. https://www.openstreetmap.org, 2017.

[Pa62] Parzen, E.: On estimation of a probability density function and mode. The Annals of Mathematical Statistics, 33:1065–1076, 1962.

[PSDG09] Piorkowski, M.; Sarafijanovoc-Djukic, N.; Grossglauser, M.: A parsimonious model of mobile partitioned networks with clustering. In: The First Int. Conf. on COMmunication Systems and NETworkS. pp. 1–10, 2009.

[Rh11] Rhee, I.; Shin, M.; Hong, S.; Lee, K.; Kim, S. J.; Chong, S.: On the Levy-walk nature of human mobility. IEEE/ACM Transactions on Networking, 19:630–643, 2011.

[RMC15] Radford, A.; Metz, L.; Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint, arXiv:1511.06434, 2015.

[Ro56] Rosenblatt, M.: Remarks on some nonparametric estimates of a density function. The Annals of Mathematical Statistics, 27:832–837, 1956.

[Te17] TensorFlow: Text generation using a RNN with eager execution. `https://www.tensorflow.org/tutorials/sequences/text_generation` (Accessed: 12th April 2019), 2017.

[Tu50] Turing, A. M.: Computing Machinery and Intelligence. Mind, 59:433–460, 1950.

[Yu10] Yuan, J. et al.: T-Drive: Driving directions based on taxi trajectories. In: Proceedings of 18th ACM SIGSPATIAL Conf. on Advances in Geographical Information Systems. pp. 99–108, 2010.

[Yu11] Yuan, J.; Zheng, Y.; Xie, X.; Sun, G.: Driving with knowledge from the physical world. In: Proceedings of the 17th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. pp. 316–324, 2011.

[Zh16] Zhao, K.; Tarkoma, S.; Liu, S.; Vo, H.: Urban human mobility data mining: An overview. In: 2016 IEEE Int. Conf. on Big Data. pp. 1911–1920, 2016.

[Zh17] Zhu, J. Y.; Park, T.; Isola, P.; Efros, A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE Int. Conf. on Computer Vision. pp. 2223–2232, 2017.

[ZXM08] Zheng, Y.; Xie, X.; Ma, W. Y.: Understanding mobility based on GPS data. In: Proceedings of the 10th Int. Conf. on Ubiquitous Computing. pp. 312–321, 2008.

[ZXM09] Zheng, Y.; Xie, X.; Ma, W. Y.: Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th Int. Conf. on World Wide Web. pp. 791–800, 2009.

[ZXM10] Zheng, Y.; Xie, X.; Ma, W. Y.: GeoLife: A collaborative social networking service among user, location and trajectory. IEEE Data Engineering Bulletin, 33:32–39, 2010.