On-Device Deep Learning Location Category Inference Model*

Gadzhi Musaev[1,3][0000−0001−9067−4495], Kevin Mets[2][0000−0002−4812−4841], Rokas Tamošiūnas[3][0000−0002−4325−5818], Vadim Uvarov[3][0000−0001−5885−0567], Tom De Schepper[2][0000−0002−2969−3133], and Peter Hellinckx[4][0000−0001−8029−4720]

University of Antwerp - imec
IDLab - Faculty of Applied Engineering[1],
Department of Computer Science[2]
Sint-Pietersvliet 7, 2000 Antwerp, Belgium
Sentiance[3]
Korte Lozanastraat, 20-26, Antwerp, Belgium
University of Antwerp, Faculty of Applied Engineering[4]
{kevin.mets, tom.deschepper, peter.hellinckx}@uantwerpen.be,
{gadzhi.musaev, rokas.tamosiunas, vadim.uvarov}@sentiance.com

Abstract. We define Location Category Inference (LCI) as a task of predicting the category of a visited venue, such as *bar*, *restaurant* or *university*, given user location GPS coordinates and a set of venue candidates. LCI is an essential part of the hyper-personalization systems as its output provides deep insights into user lifestyle (has children, owns a dog) and behavioral patterns (regularly exercises, visits museums). Due to such factors as signal obstruction, especially in urban canyons, the GPS positioning is inaccurate. The noise in the GPS signal makes the problem of LCI challenging and requires researchers to explore models that incorporate additional information such as the time of day, duration of stay or user lifestyle in order to overcome the noise-induced errors. In this paper we propose an embeddable on-device LCI model which fuses spatial and temporal features. We discuss how initial clustering of locations helps limiting the GPS noise. Then, we propose a multi-modal architecture that incorporates socio-cultural information on when and for how long people typically visit venues of different categories. Finally, we compare our model with one nearest neighbor, a simple fully connected neural network and a random forest model and show that the multi-modal neural network achieves f1 score of 73.2% which is 6.6% better than the best of benchmark models. Our model outperforms benchmark models while being almost 180 times smaller in size at around 1.9Mb.

Keywords: Location Category Inference · Deep Learning · On-device Machine Learning.

---

* Supported by Sentiance.

1 Introduction

A system that maps raw mobile user location coordinates to a semantic meaning of that location is a rich source of knowledge and insights for a variety of applications. We can broadly define two categories of insights that an LCI system might provide: user profiles and contextual moments. The main function of a user profiling model is to get insights into *who* the users are: parents, students, office workers, etc. Contextual moment inference models are designed to give insights into *what they are busy with at certain moments*: commute, lunch, shopping, leisure, vacation, etc. Context-aware advertisement [14], personalized next point-of-interest recommendation [6], behavior change [10] and other applications rely explicitly or implicitly on such models in order to deliver most appropriate messages to right users in timely manner.

Despite the undeniable progress in both mobile hardware and software during the last years, this problem is still challenging due to such technical problems as signal obstruction, signal reflection in densely built-up areas such as city centers, the trade off between GPS accuracy and battery consumption, etc. The uncertainty in the GPS location estimation leads researchers to explore additional sources of information. It has been shown that incorporating time of events and socio-cultural clues on how venues of different categories are typically used has significant impact on the accuracy of the LCI predictions [3, 4]. Other studies also show that the impact of the GPS noise can be reduced by applying clustering on the set of all locations visited by a user and grouping the visits that correspond to the same location together [12, 13].

Additional challenges arose recently with the growing concerns about privacy in location-based services. A number of recent studies explore geoprivacy issues and potential solutions [7, 16, 17]. This problem is not merely of academic interest, but is also critical for the industry. Today, businesses are expected to handle their users location-based data with full responsibility and limit the data storage and processing to the minimum required to provide the service. Thanks to the recent advances in mobile hardware and emergence of on-device deep learning frameworks such as tensorflow-lite, one of the practical steps to improve user experience with location-based services is to build on-device models and to guarantee that the location data never leave user devices. Unfortunately, this requires new research since many models proposed earlier were not designed to run in restricted mobile environments.

In this paper we address the problem of building an LCI model that satisfies both requirements: it must be lightweight and embeddable into mobile applications while incorporating additional temporal features and ensuring best performance in terms of accuracy metrics. We use one nearest neighbour model as a baseline that shows the lower bound of an LCI model performance. We also compare our model against a random forest which has been successfully applied to the same task [2, 9]. Finally, we train a simple fully connected neural network and show that such a model would not achieve performance of the random forest.

Main contributions of this paper are following:

– We present a lightweight on-device deep learning LCI model embeddable into mobile applications that outperforms larger cloud-based models.
– We present a novel method of encoding spatial data on positions of an arbitrary number of venues in the neighbourhood as a spatial histogram of a fixed size. In order to combine the spatial histogram with temporal features we propose a multi-modal architecture.
– The data set we collected for our model is unique. Next to well-known temporal models based on time of day, we also introduce duration-based models.

2 Background

The model we propose is intended to work as a component of mobile applications that require insights into the lifestyle of a user. Since the runtime of the inference is significantly different from the training runtime, we need to discuss these additional details and also establish terminology used in the following sections. In order to facilitate the discussion, the Journeys application is used as an example. Figure 1 presents a screen of the Journeys application which demonstrates the timeline of a user.

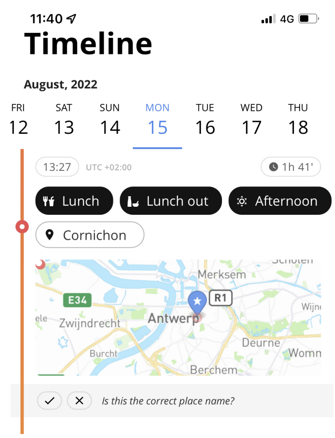


Fig. 1. The Journeys app. The timeline screen shows the timeline for an entire day of the user including each location they visited and transport modes they used to move around. Here we show a correctly identified visit to a lunch place in Antwerp. Using the insights from the LCI model we can compute the 'moments' of the user (*Lunch* in this case)

As a user moves around their area of living while carrying a mobile device, each moment of time can be considered as either a moment of staying at a fixed location or moving from one location to another. Thus, the timeline of the user unfolds into a sequence of being still connected with transportations between them. For example, if one leaves their home in the morning, takes a bus to the office, works till noon and then goes to a cafe nearby, their timeline can be represented as a sequence $S(home) \rightarrow T(bus) \rightarrow S(work) \rightarrow T(walk) \rightarrow S(cafe)$. We call each such event S a *stationary* and each T a *transport*. Describing specific algorithms that detect stationaries and transports are out of scope of this paper. What is important is that a stationary contains all spatial and temporal variables our model needs to run the inference: $S = (lat, lon, start, stop)$. For each stationary, the LCI model is applied in order to classify the stationary into one of the supported categories, such as *drink-day*, *drink-evening*, etc. We derive the categories based on the OSM tags attached to each venue. The full list of supported categories together with a short description and an example of a corresponding OSM tag is given in the table 3.

In order to infer a category of a location, we also need to obtain a set of candidate venues in a certain radius around the location. We call a component that provides a service for executing such queries a *venue provider*. The representation of the set of venue candidates in a form of features is called the *environment fingerprint*. Describing an implementation of a venue provider is out of scope of this paper, but it is important to mention that n order to guarantee that the location data never leave the user device, the venue provider should also be on-device.

3 Methodology

We propose an end-to-end system that receives raw stationary events, runs clustering on the entire timeline, builds spatial histogram of venues, incorporates temporal models and finally outputs the category of the visited location. In this section we describe each of the steps and models used in them. A high-level overview of the entire pipeline is presented in Figure 2.

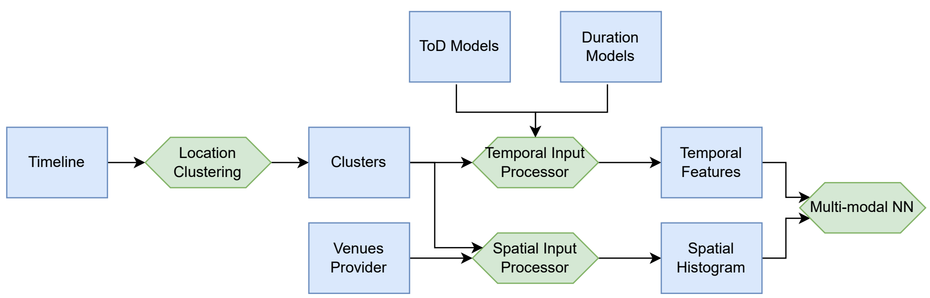


Fig. 2. Location category inference pipeline

3.1 Location Clustering

A location clustering model should be able to answer a simple question: which of the stationary events in the user timeline correspond to visiting the same location. We hypothesise that by clustering separate locations and considering their centroids instead of raw GPS coordinates, we can average the noise out and get closer to the true location. Figure 3 shows that there is indeed a positive correlation between the number of visits to the same location and the accuracy of the model.

Existing research on the topic of location clustering based on GPS coordinates shows that density-based clustering techniques work well for the task [12,

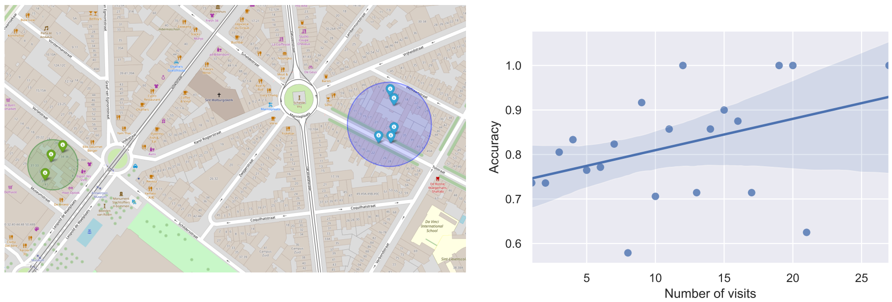


Fig. 3. Visualization of the location clustering action and its impact on the LCI model. We can see that as the cluster grows with more repeated visits, the model has bigger chance to classify it correctly

13]. Following the ideas from these papers, we implement a version of DBSCAN compatible with tensorflow-lite and use the scikit-learn implementation of DBSCAN as a benchmark to make sure our model produces the same clusters on both synthetic and real-world data.

3.2 Temporal Models

Numerous research papers such as [4, 11] have shown how temporal data can improve location category inference models compared to those that only rely on spatial data. Following the ideas of McKenzie et al. [4] we construct temporal models similar to temporal semantic signatures and use the likelihoods produced by these models as input features for the neural network. Instead of having histograms with wide bands of 1 day or 1 hour, we aim for more granularity and fit a kernel density estimation (KDE) model per category. Each KDE model represents the likelihood of being at location of the corresponding category at the given time of day for each day and each time bin of 12 minutes. In figure 5 we can see Monday and Sunday time of day models for 3 different categories *drink-evening*, *education-parents* and *travel-hotel*. We can clearly see that those models successfully captured the visiting patterns for each of the categories. For

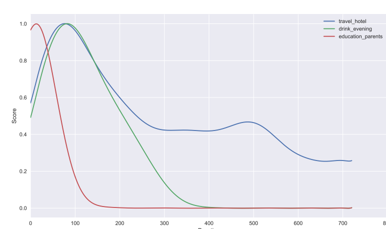


Fig. 4. Example of duration models for *drink-evening*, *education-parents* and *travel-hotel*. Duration is measured in minutes.

example, the *education-parents* on Monday has two spikes: one early in the morning and around 16:00. Those are typical hours of bringing children to school and picking them up once their classes finish. The corresponding curve on Sunday is almost flat which is expected as the schools are closed on Sunday.

Unlike with other check-in data sets, we have access to both start time and the duration of stay for each stationary event in the data set. Thus, we apply the same technique to build temporal signatures based on duration as well. Fig. 4 shows an example of the duration models for the same three categories. A vivid example of its potential impact is the model for *travel-hotel*: it has a second spike around 8 hours and it will yield much larger likelihood values for longer durations than the other two models.

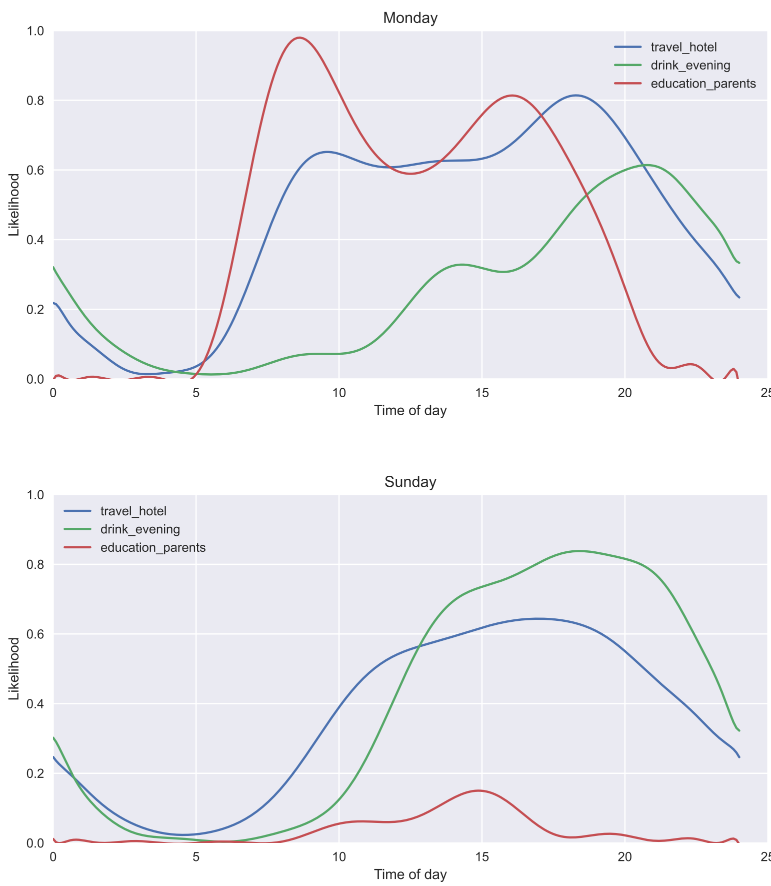
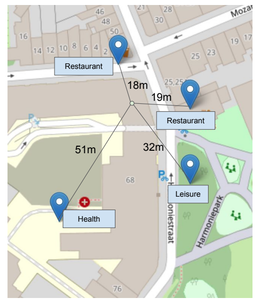


Fig. 5. Example of ToD models for Monday and Sunday for *drink-evening*, *education-parents* and *travel-hotel*

3.3 Spatial Histogram

Once we computed clusters and obtained the centroid of the cluster related to the latest stationary, we need to query the venue provider. Discussing implementation of such queries is out of scope of this paper, but it's important to mention that avoiding the exchange of GPS coordinates with a server requires an on-device implementation of the venue provider. For our application we populate a local on-device database with OSM data in a large radius around the user, so their exact location is never exchanged.



	Leisure	Shop	Resto	Health
<10m	0	0	0	0
<40m	1	0	1	0
<100m	0	0	0	1

Fig. 6. Example of encoding environment fingerprint of arbitrary size into a fixed-size spatial histogram

A query to the venue provider returns an arbitrary number of venue candidates. For example, a query for venues in a city center could return hundreds of candidates whereas a query for venues around a gas station in the outskirts would only return a few. Tensorflow-lite is a restricted environment and the input to the tensorflow-lite models must be of a fixed size. However, taking only the N nearest venues is not optimal. Setting N to be too small would lead to missing true venues in densely built-up areas and making it too large would result in unnecessary computations. We solve this technical issue by encoding the venue candidates into a fixed-size table called spatial histogram. Every column in this table corresponds to a venue category and each row is a band of distances. Since the true venue is more likely to be among the candidates located closer to the centroid than those further away, a logarithmic scale is used instead of a fixed step. Each cell c_{ij} contains 1 if a venue category j is present in the band i and 0 otherwise. Fig. 6 shows an example of the spatial histogram construction.

3.4 Location Category Inference Model

The output of the spatial histogram encoding procedure described above is passed to the neural network as it is. In order to complete the discussion of data preparation for the multi-modal NN model, we need to explain how the temporal input is constructed. We can think of each time of day model as a function that maps a triplet of $(category, day, hour)$ to the corresponding likelihood: $tod : (category, day, hour) \rightarrow \mathbb{R}$. The duration model can be defined similarly: $dur : (category, duration) \rightarrow \mathbb{R}$. We precomputed seven tables for time of day models per day of week and one table for the duration models. For

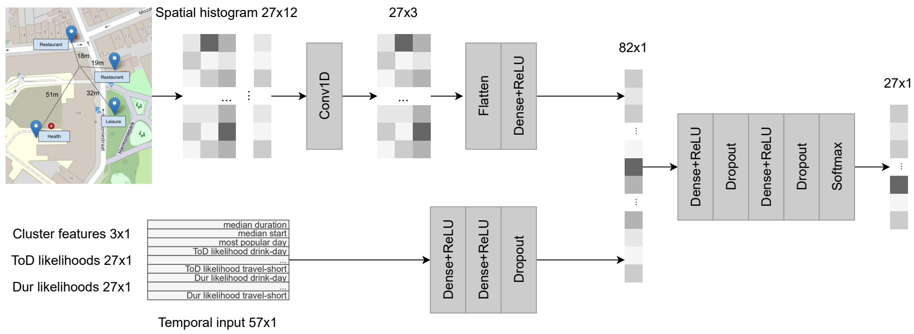


Fig. 7. Architecture of the Location Category Inference NN model

each training sample we simply query the corresponding time of day table and the duration table to obtain 27 time of day and 27 duration likelihoods and concatenate them into a single vector together with cluster features. The resulting vector is visualized on the figure 7 as the 'Temporal Input'

Since temporal features are encoded as a one-dimensional vector and the spatial histogram is two-dimensional, we build a multi-modal neural network which has two separate inputs. Temporal features are processed with a small fully-connected neural network, whereas the spatial histogram is first processed with a 1D convolutional layer with 3 filters and then flattened. Vector representations of both temporal and spatial features are then concatenated into a single vector which is further processed by few more dense layers. The last layer is a softmax layer which represents probabilities assigned by the neural network to each of the supported categories. Figure 7 visualizes the architecture of the NN.

4 Experiments

The model is trained, fine-tuned and evaluated on a data set of 25709 clusters labelled via the feedback functionality of the Journeys app discussed above. The data set spans 64 different countries but is imbalanced and the majority of records come from the US and Europe. For the experiments we set a fixed 10% subset of our data aside for evaluation. A stratified split was applied to ensure that the class imbalance in both training and evaluation are similar. We train and validate each model as a multi-class classifier that should predict one of the 27 predefined location categories. We use 4 metrics to judge the models: *accuracy* and macro-averaged *precision*, *recall* and *f1 score*. In the following subsections we will describe how each model was trained. The results are analysed in the next section.

4.1 Nearest Neighbor

The nearest neighbor is a trivial classifier which selects the venue nearest to the cluster centroid and uses it's category as a prediction for the location category.

4.2 Random Forest

Random forests (RF) as well as ensemble models based on gradient boosting, are known to perform well in classification tasks on tabular data. Alsudais et al. apply RF to infer categories of locations where users twitted from [2]. In a more recent study Kim et al. apply RF to infer location categories based on a variety of personal data such as *age*, *gender*, *hobby*, etc [9]. To train our version of RF LCI model, we transform the data set into a tabular form where each row represents a cluster. For that, we transform the spatial histogram into a simplified version where we only keep the distance to the nearest venue of each category.

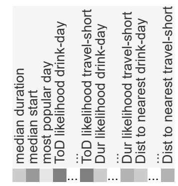


Fig. 8. Vector of features for the Random Forest model

4.3 Simple Fully-Connected Neural Network

In order to verify that our multi-modal architecture justifies additional complexity, we also train a simple fully-connected network. We reuse the tabular dataset produced for the RF model, but we additionally apply standard feature normalization for *median-duration*, *median-start-hour*, *most-popular-day* and the distance-based features by dividing each feature by the corresponding maximal value. To address the problem of overfitting, aggressive regularization is applied. We use l_2 regularization with the l_2-weight $w = 0.01$ on each dense layer and also add dropout layers after each dense layer with the dropout-rate $\alpha = 0.4$ which is similar to the regularization used for the multi-modal NN. Early-stopping is used based on the value of the loss function on the dev set which is 15% of the training data set. Class weights are computed similar to the multi-modal NN model.

4.4 Multi-Modal Neural Network

A series of experiments have been conducted to fine-tune hyperparameters and avoid overfitting. First, a model without any regularization has been trained and proved to overfit quickly. We searched for optimal values for the l_2-weight w in range $[10^{-5}, 10^{-1}]$ and obtained best results with $w = 0.01$. For the dropout-rate hyperparameter we searched in range $[0.1, 0.5]$. Larger dropout-rates allow training larger models without overfitting and for our model the optimal value was $\alpha = 0.4$. Increasing it leads to overfitting whereas increasing the model size does not yield any improvements in terms of validation metrics. Finally, early stopping based on the value of the change of the loss on the validation dataset is applied with the patience factor of 100.

drink-day	367	leisure-museum	213	shop-short	2481
drink-evening	830	leisure-nature	52	sport	946
education-independent	175	leisure-park	901	sport-attend	65
education-parents	618	office	2579	travel-bus	675
health	459	religion	247	travel-conference	6
industrial	32	residential	290	travel-fill	1395
leisure-beach	87	resto-mid	2446	travel-hotel	682
leisure-day	68	resto-short	363	travel-long	1123
leisure-evening	272	shop-long	3741	travel-short	1971

Table 1. Classes are highly imbalanced

Location-based data with true labels for visited venues are imbalanced by nature: people visit venues of certain categories more often than others. Willingness to share personal visits is also biased towards more popular venues such as restaurants and bars. Table 1 shows the class imbalance in our data set. In order to overcome this issue, we compute the standard class weights computation procedure as defined in scikit-learn and specify these weights during the training.

5 Results and Discussion

Table 2 shows the results for each model. Due to the class imbalance problem, the accuracy score alone is not enough to compare the models. A model that trades off recall on the less presented classes to increase its accuracy and precision, for example, would lead to poor user experience on each of the classes underrepresented in the training data set. For that reason, the f1 score is considered more important for the current study.

- 1-NearestNeighbor achieves 68.9% accuracy and 64.4% f1 score. To the best of our knowledge, this is significantly larger than in the previous research. For example, Shaw et al. report only 20%, although it is important to notice that in their research the exact venue is predicted and not only its category.
- As expected, Random Forest performs significantly better than the benchmark in terms of accuracy (+7.2%). It is important to notice, however, that the f1 score gain is much lower (+2.2%). Due to the imbalanced classes, Random Forest achieves the best precision of all the models, but the recall is significantly low.
- The naive NN fails to achieve the performance of Random Forest and shows performance similar to the baseline model.
- The multi-modal architecture we propose performs significantly better than the competitors. It is on par with the Random Forest model in terms of accuracy, while achieving the best recall score, which gives the f1 score of 73.2% which is 6.6% better than that of the Random Forest.
- The multi-modal NN model only weighs 1.9Mb whereas a serialized version of the random forest weighs 364Mb.

Model	Accuracy	Precision	Recall	F Score
1-NearestNeighbor	68.9	62.5	71.2	64.4
Naive NN	67.6	62.0	71.7	64.5
RandomForest	75.7	81.8	61.5	66.6
Multi-modal NN	76.4	75.2	72.2	73.2

Table 2. Results

6 Related Work

The problem addressed in this paper has many variations and is also known under different names. Yi et al. propose an LCI model that shows promising results [11]. The main difference of our model is that it is lightweight and can be used on-device. The authors also mention extremely high location estimation uncertainty in their data set, which is not the case for the data we collected. McKenzie et al. address the problem of mapping the user location to a specific venue under the name of 'reverse geocoding' [4]. We reused their ideas on constructing temporal signatures with some changes geared towards more granularity in time resolution and used KDE models instead of raw histograms to avoid having underrepresented time bands. Shaw et al. pose a similar problem as learning to rank [1]. The main difference is in the application domain: we are interested in fully autonomous prediction of a single category whereas [1] aim to select top-N candidates and let the user pick the correct one.

He et al. are solving a very similar problem but from a recommendation systems point of view [6]. PoI recommendation seems to be the most actively studied setting with respect to the LCI problem and Islam et al. [15] provide an overview of the most recent advancements in this field with deep learning techniques. Bao et al. also produced a survey on recommendations in location-based services without a specific focus on deep learning [3]. Duan et al. apply recurrent neural networks to build embeddings of user locations and predict the next visited PoI [8]. We also learn representations of user locations, but the main difference is that we only use geospatial data for that, whereas Duan et al. mix in some data on the user and textual data on the location. Another difference is that for us it is less relevant to apply the RNN since we do not try to predict the next visited PoI.

Angmo et al. study the impact of clustering on identifying significant locations from spatio-temporal data and propose improvements for the classical DBSCAN algorithm [13]. Due to the specific features of the data set, authors use rather large values for the minimum number of points in a cluster - 50, 80, 100. For our model, we set this parameter to 1, since even a single visit to a location matters and we do not have any a-priori assumptions about this feature of the data set. Andrade et al. also study DBSCAN for location clustering and introduce a new variation of DBSCAN for spatio-temporal data that derives significant locations [12]. They also apply Gaussian Mixture Models on the obtained clusters in order to derive habits of the users. Given the success of density-based clustering techniques on this task, we also applied DBSCAN for location clustering.

Some researchers address the LCI problem but with the data sets enriched with data inaccessible to us. Kim and Song successfully apply random forest models to predict categories of visited venues based on fusion of location data and personal data such as *age, job, salary*, etc [9]. Alsudais et al. also train random forest models for the same problem, but they mix in textual data from the tweets of the users. We take inspiration from these results and train our own version of a random forest model to challenge the multi-modal neural network.

Another interesting study related to building embeddings is DeepCity [5]. The authors propose a framework that can learn embeddings for both locations and users by utilizing task specific random walks on a bipartite graph. As one of the use cases, they consider location category prediction. The embeddings learned by the DeepCity model are passed to a task-specific classifier model. Unfortunately, the size of the embeddings table is prohibitively large for our use case.

7 Conclusion

In this paper we present an on-device multi-modal neural network model for the location category inference problem which outperforms large ensemble based models while being lightweight and embeddable into mobile applications. We present a simple yet effective method to incorporate both spatial and temporal features - temporal models and the spatial histogram. Due to this method, our model performs significantly better than a naive neural network which utilizes simpler representation for its input data.

8 Appendix

Category	Description	Example OSM tag
drink-day	Coffee bars and tea rooms	amenity:cafe
drink-evening	Bars and pubs	amenity:bar
education-independent	Educational centers for adults, universities	building:university
education-parents	Primary schools and kindergartens	amenity:kindergarten
health-long	Hospitals	amenity:hospital
health-short	Dentists and GP's	amenity:dentist
industrial	Factories and warehouses	building:industrial
leisure-beach	Beaches and resorts	leisure:beach
leisure-day	Day-time entertainments	sport:paintball
leisure-evening	Evening entertainments	amenity:cinema

leisure-museum	Museums	tourism:museum
leisure-nature	Nature reserves	leisure:fishing
leisure-park	City parks and gardens	leisure:park
office	Office buildings	building:office
religion	Monasteries, abbeys	building:cathedral
residential	Residential buildings, houses	building:apartments
resto-mid	Restaurants	amenity:food_court
resto-short	Fast food, sandwich bars	amenity:ice_cream
shop-long	Malls, shopping centers	shop:mall
shop-short	Small local shops, bakeries	shop:bakery
sport	Gyms, sport centers	leisure:fitness_center
sport-attend	Stadiums	building:stadium
travel-bus	Bus stops	highway:bus_stop
travel-conference	Conference halls	amenity:conference_centre
travel-fill	Gas stations	amenity:fuel
travel-hotel	Hotels and B&B's	building:hotel
travel-long	Airports	aeroway:airport
travel-short	Public transport stations	station:subway

Table 3: Full list of location categories.

References

[1] B. Shaw et al. "Learning to rank for spatiotemporal search". In: *WSDM '13*. 2013. DOI: 10.1145/2433396.2433485.

[2] Abdulkareem Alsudais, Gondy Leroy, and Anthony Corso. "We Know Where You Are Tweeting From: Assigning a Type of Place to Tweets Using Natural Language Processing and Random Forests". In: *2014 IEEE International Congress on Big Data*. ISSN: 2379-7703. June 2014, pp. 594–600. DOI: 10.1109/BigData.Congress.2014.91.

[3] Jie Bao et al. "Recommendations in location-based social networks: A survey". In: *GeoInformatica* 19 (July 2015). DOI: 10.1007/s10707-014-0220-8.

[4] Grant McKenzie and Krzysztof Janowicz. "Where is also about time: A location-distortion model to improve reverse geocoding using behavior-driven temporal semantic signatures". In: *Computers, Environment and Urban Systems* 54 (Nov. 2015). DOI: 10.1016/j.compenvurbsys.2015.05.003.

[5] Jun Pang and Yang Zhang. "DeepCity: A Feature Learning Framework for Mining Location Check-Ins". en. In: *Eleventh International AAAI Conference on Web and Social Media*. May 2017. URL: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15555 (visited on 07/28/2022).

[6] Jing He et al. *Personalized next point-of-interest recommendation via latent behavior patterns inference*. Number: arXiv:1805.06316 arXiv:1805.06316 [cs]. May 2018. DOI: 10.48550/arXiv.1805.06316. URL: http://arxiv.org/abs/1805.06316 (visited on 07/25/2022).

[7] Carsten Keßler and Grant McKenzie. "A geoprivacy manifesto". en. In: *Transactions in GIS* 22.1 (2018). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/tgis.12305, pp. 3–19. ISSN: 1467-9671. DOI: 10.1111/tgis.12305. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.12305 (visited on 08/08/2022).

[8] Yingying Duan et al. "PBEM: A Pattern-Based Embedding Model for User Location Category Prediction". In: *2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*. Nov. 2019, pp. 1–6. DOI: 10.23919/ICMU48249.2019.9006662.

[9] Young Myung Kim and Ha Yoon Song. "Analysis of Relationship between Personal Factors and Visiting Places using Random Forest Technique". In: *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*. ISSN: 2300-5963. Sept. 2019, pp. 725–732. DOI: 10.15439/2019F318.

[10] Shannon Wongvibulsin et al. "An Individualized, Data-Driven Digital Approach for Precision Behavior Change". In: *American Journal of Lifestyle Medicine* 14 (Apr. 2019), p. 155982761984348. DOI: 10.1177/1559827619843489.

[11] Jinfeng Yi et al. "Fast Unsupervised Location Category Inference from Highly Inaccurate Mobility Data". In: *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*. Proceedings. Society for Industrial and Applied Mathematics, May 2019, pp. 55–63. DOI: 10.1137/1.9781611975673.7. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611975673.7 (visited on 08/22/2022).

[12] Thiago Andrade, Brais Cancela, and João Gama. "Mining Human Mobility Data to Discover Locations and Habits". en. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Peggy Cellier and Kurt Driessens. Communications in Computer and Information Science. Cham: Springer International Publishing, 2020, pp. 390–401. ISBN: 978-3-030-43887-6. DOI: 10.1007/978-3-030-43887-6_32.

[13] Rigzin Angmo et al. "An Improved Clustering Approach for Identifying Significant Locations from Spatio-temporal Data". en. In: *Wireless Personal Communications* 121.1 (Nov. 2021), pp. 985–1009. ISSN: 1572-834X. DOI: 10.1007/s11277-021-08668-w. URL: https://doi.org/10.1007/s11277-021-08668-w (visited on 08/09/2022).

[14] Carmen De Maio et al. "Who and where: context-aware advertisement recommendation on Twitter". en. In: *Soft Computing* 25.1 (Jan. 2021), pp. 379–387. ISSN: 1433-7479. DOI: 10.1007/s00500-020-05147-5. URL: https://doi.org/10.1007/s00500-020-05147-5 (visited on 08/05/2022).

[15] Md. Ashraful Islam et al. "A survey on deep learning based Point-of-Interest (POI) recommendations". en. In: *Neurocomputing* 472 (Feb. 2022),

pp. 306–325. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2021.05.114. URL: https://www.sciencedirect.com/science/article/pii/S0925231221016106 (visited on 07/28/2022).

[16] Grant McKenzie et al. "PrivyTo: A privacy-preserving location-sharing platform". en. In: *Transactions in GIS* 26.4 (June 2022), pp. 1703–1717. ISSN: 1361-1682, 1467-9671. DOI: 10.1111/tgis.12924. URL: https://onlinelibrary.wiley.com/doi/10.1111/tgis.12924 (visited on 07/28/2022).

[17] Hongyu Zhang and Grant McKenzie. "Rehumanize geoprivacy: from disclosure control to human perception". en. In: *GeoJournal* (Feb. 2022). ISSN: 1572-9893. DOI: 10.1007/s10708-022-10598-4. URL: https://doi.org/10.1007/s10708-022-10598-4 (visited on 08/03/2022).