LSTM-based generation of cellular network traffic

Anne Josiane Kouam, Aline Carneiro Viana, Alain Tchana

LSTM-based generation of cellular network traffic

Anne Josiane Kouam*, Aline Carneiro Viana*, Alain Tchana†,
* *Inria*, France † *ENSIMAG*, France

***Abstract*—Domain-wide recognized by their high value in human activity and network monitoring studies, cellular network traffic (i.e., Charging Data Records, named CDRs), however, present accessibility and usability issues, restricting their exploitation and research reproducibility. This paper tackles such challenges by modeling CDRs that fulfill real-world data attributes. Our designed framework, named *Zen* leverages LSTM to realistically model network users' traffic behavior through a 4-stage generative pipeline. Results show that *Zen*'s models accurately capture individual and global distributions of a fully anonymized real-world traffic CDRs dataset. Finally, we validate *Zen* CDRs ability of reproducing daily cellular behaviors of the urban population and its usefulness in practical networking applications such as Radio Access Network's power savings, and anomaly detection as compared to real-world CDRs.**

***Index Terms*—Cellular traffic modeling, CDRs, LSTM**

I. Introduction

Cellular network datasets are a standard tool for studying users' traffic behavior on a large scale. Such datasets have been leveraged in various literature domains, such as sociology [1] and networking [2]. Most complete traffic datasets (including data usage, voice calls and SMS) provide an understanding of users' mobile traffic demands, which is of fundamental importance to improve the quality of communication service and better foreseen timely planned network resource allocation. We refer to such datasets as Charging Data Records (CDRs).

The exploitation of real-world CDRs for research faces two main limitations. First, *accessibility*: CDRs datasets are not publicly available, imposing strict mobile operators' agreements. Second, *usability*: CDRs are usually available in an aggregated form (i.e., users flows and coarse temporal information), limiting related analyses' preciseness. This paper *addresses such limitations by enabling the autonomous generation of realistic traffic CDRs by scientific community, thus providing new avenues for research advances.*

Our generative framework aims at producing *complete* network traffic behaviors, *described per user*. Indeed current literature has modeled cellular traffic based on a unique event type: voice calls [3], [4] or data [5]. Such models, unfortunately, lack the richness provided by both social interactions (in voice calls and SMS) and data usage included in complete network traces. Moreover, we differ from the literature contributions modeling network usage aggregated by geographical area [6], [7], periods, or user profiles [5], [3]. Instead, we reproduce individuals' traffic behaviors required for generated traces' realisticness; which implies coping with the notable heterogeneity in user traffic habits.

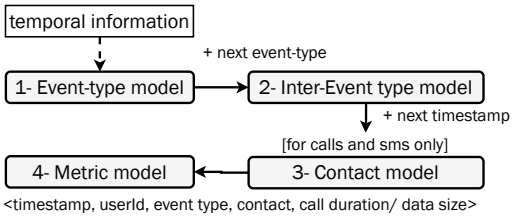


Fig. 1: *Zen* generative pipeline.

To the best of our knowledge, *this is the first work in literature producing realistic Charging Data Records (CDRs) that fulfill the above-mentioned attributes*. Our designed framework, named *Zen* employs the following methodology:

(1) Leveraging on a real-world fully anonymized CDRs describing individual traffic behaviors (event type – data, call, and SMS –, duration, and pairwise information, etc), *we propose the first modeling that captures long-range and inter-CDRs feature correlations, while addressing the traffic behavior heterogeneity.* We use three separate *Long-Short-Term Memory neural networks* (LSTM) to model event types generation (i.e., *what*), inter-event duration (i.e., *when*), social interactions (i.e., *whom*), and leverage statistical analysis to model CDRs metrics such as calls duration (*i.e., how*). We show *Zen* traffic modeling presents significant high performance values, with for 80% of users (i) more than 95% (for event-type) and 75% (for inter-event) of modeling accuracy, and (ii) less than 6.68% (for inter-event) and 12.5% (for social) of Mean Absolute Error's maximum values.

(2) *Zen* implements a real cellular structure for the generation of realistic CDRs traffic while providing configuration flexibility (e.g., number of users, duration). Upon publication, we will make *Zen* publicly available, i..e, code and generated dataset.

(3) We validate *Zen* capability of reproducing daily urban cellular behaviors and its usefulness in two practical networking application: Radio Access Network's power savings and anomaly detection.

II. Dataset description

Zen is built from a real-world reference dataset, which we name *RefCDRs*. *RefCDRs* refers to a fully-anonymized CDRs dataset collected by a major mobile network operator. It describes 1-month (*from 2018-06-01 to 2018-06-30*) per-user traffic resulting in about 3 million timestamped events generated by 186,738 distinct phone numbers, where about 17,000 are from the *RefCDRs*' operator. *RefCDRs* lacks incoming-SMS traffic type (i.e., only have outgoing SMS) and provides

no information on data events' session sizes. First, we only consider events of *users subscribed* to the *RefCDRs*'s operator. Then, we filter out users having less than 3 generated events in the whole period of 4 weeks and those with more than one event at the same timestamp. Those manipulations result in the selection of nearly 6000 users totalizing 1,782,829 events or CDRs entries, i.e., 77.8% of the *RefCDRs*' initial size.

III. Design

This Section describes the generative model used to reproduce CDRs traffic behavior, as depicted in Fig. 1. Our generative model has enough expressive power to capture inter-CDRs feature correlations while considering individual users' behavior. In particular, we leverage an enhanced *recurrent neural network* (RNN), named *Long-Short-Term Memory* (LSTM), known for its ability to generate complex, realistic long-range sequences.

Our models are trained from *RefCDRs* which report a set of timestamped events generated by several users. Each CDRs' event (or a line) includes the following information: start time, user id (i.e., phone number), event-type (i.e., data, SMS, call), corresponding user id (for calls and SMS), call duration (for calls only), and data volume (for data only).

We organize *RefCDRs* by user: the set of events chronologically generated by the user u throughout the trace forms a sequence of events $(e_1^u, e_2^u, e_3^u, ...e_{N_u}^u)$ of size N_u, which is the model basis. Hence, data reproduction is done in a sequential order, i.e., from time step 1 to N_u; and the generation of each event of the sequence is a four-stage process, where each stage relies on the previous output.

Stage 1: at step t, we predict the next event-type e_{t+1}^u a user will perform, using the *event-type model* (cf. §III-A).

Stage 2: given the event-type, the *inter-event time (IET) model* generates the IET value used to deduce the starting time for the predicted event-type e_{t+1}^u (cf. §III-B).

Stage 3: the *contact model* predicts which of its contact a user will interact with for the next event e_{t+1}^u (§III-C). This model is executed only if e_{t+1}^u is a call or SMS, i.e., the only events requiring contact interactions.

Stage 4: Finally, the *metric model* refers to how the events are generated: For call events, it generates its duration, while for for data events, it produces the data volume (§III-D). Note that the temporal information is not constant throughout the pipeline. From stages 1 to 2, we use the temporal information of the event-type at step t to predict the one of the event-type at step $t+1$, then used in stage 3.

A. Event-type modeling

The *event-type model* predicts the next event-type a user will generate from four types of events: data, local call (uniquely outgoing), international call (outgoing or incoming), and local SMS (uniquely outgoing). Local incoming calls and SMS are modeled here as they are induced from outgoing calls and SMS during the generation. Modeling international calls separately from local calls, rather than having a unique "call" event-type and determining probalistically if it is local or international, allows distinguishing different user behaviors towards international calls. Indeed, some users may not make international calls while others make them frequently. Finally, we did not model international SMS event-type because it is rare and not present in *RefCDRs*.

The event-type model. We model sequences of event-types using an LSTM. At step t, the LSTM takes as input a vector of features x_t and generates a vector of four scores, $y_t = (y_t^1, y_t^2, y_t^3, y_t^4)$. These scores parameterize a multinomial distribution $Pr(\widehat{e}_t^u|y_t)$ for the next event-type $\widehat{e}_{t+1}^u$, through a softmax function: $Pr(\widehat{e}_t^u|y_t) = \frac{exp(y_t^k)}{\sum_{k'=1}^{4} exp(y_t^{k'})}$.

When training, the true previous event-types at step t are encoded as input for the next step. Network parameters' training is done according to the standard approach of minimizing the negative-log-likelihood of the training data. We compute the gradient of this loss with respect to our network parameters through backpropagation.

Features x_t**.** At step t, we distinguish four features for predicting e_{t+1}^u: the event-type at step t (one-hot encoded) and its temporal features, i.e., Day-of-Week (DOW, one-hot encoded), Hour-of-Day (HOD, one-hot encoded), and Second-of-Day (SOD, cyclical encoded). A one-hot encoding represents the *i*th of N features using a N-sized vector of all zeros, except for the *i*th element, which is set to 1. A cyclical encoding maps a continuous inherently-cyclical feature into two dimensions using a sine and cosine transformation. The *HOD* and *DOW* features capture the seasonality and regularity of mobile traffic (less activity at night and during weekends). The fine-grained encoding of time as *SOD* is used to capture the very short temporal difference between consecutive events (e.g., tens of seconds for data events).

B. Inter-event time modeling

The *IET model* returns the possible time values between a sequence's events with a confidence interval. It works in two steps: first, we use an LSTM to parameterize a multinomial distribution over a discrete set of time bins. Then, we use statistical methods to sample a continuous value inside a predicted time bin. In the following, we present our considerations for discrete IET estimation, then the detail of our LSTM network, and finally, our methodology for sampling an IET value given an IET bin.

Discrete IET estimation. We divide IET into discrete bins, $b_1, .., b_J$, representing J consecutive time intervals. We found that setting the bin boundaries at evenly-spaced quantiles of time in training data was not ideal in our case. Indeed, it results in tiny intervals for the smallest values of IET due to the IET's heavy-tailed distribution. For instance, considering the 4-quantiles, there are as many elements in $[1s - 20s)$ as in $[20s - 72s)$. A division at the 20s could distort the model's accuracy while being acceptable for realistic CDRs. Thus, we chose the IET bins empirically to make the model less complex and easier to train without increasing the error in mapping

TABLE I: IET distribution and parameters per bin

IET bin	Distribution	Parameters
$[0s - 30min]$	Lognormal	$\sigma = 1.798, \mu = 4.04, x_0 = 0.99$
$(30min - 24h]$	Lognormal	$\sigma = 1.731, \mu = 8.59, x_0 = 1749.08$
$> 24h$	Exponential	$\lambda = 6.21e-06, x_0 = 86401$

back to continuous values. We, therefore, divide IET into three intervals: $[0s - 30min]$ $(30min - 24h]$, and $> 24h$.

The IET LSTM model. The LSTM network takes at each step, t, as input a feature vector, x_t and generates as output a vector of scores y_t, with one score for each possible IET bin. As with the *event-type model*, these scores are used as logits in a softmax to get a multinomial distribution over the time bins. To train the network parameters, we minimize the negative-log-likelihood of the training data.

Features x_t**.** At each step t, we consider as features, the temporal information of e_t^u (§III-A) as well as the predicted event-type e_{t+1}^u, one-hot encoded.

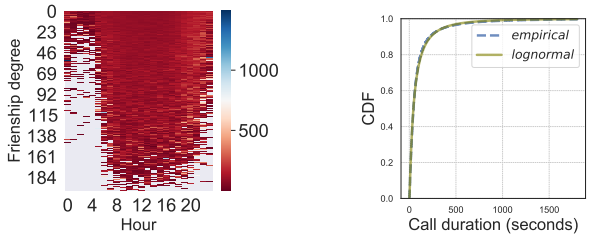
Continuous estimation. Generating CDRs traffic requires knowing the precise starting time of the next event of the sequence, which is used for further predictions. Therefore, we convert the predicted discretized IET bins to real-values. We apply to each IET bin the KS statistic test to estimate the distribution and related parameters best fitting the corresponding empirical distribution in *RefCDRs*. Table I shows the fitted distributions to sample an IET value per bin. The model returns the median value and the confidence interval of the values obtained after n sampling (by default $n = 1$).

C. Contact modeling

The *contact model* applies only for event-types requiring interaction with a contact (i.e., SMS and local or international calls). We first define the notion of *friendship degree* (fd), intuitively capturing the friendship strength of a user with each of its contacts. Let u be a user, with $\#c_u$ contacts over the considered period, we then call $\#e_c^u$ the number of events the user u had with his contact c. We increasingly order the contacts of u according to their corresponding number of events such that $\#e_1^u \leq \#e_2^u \leq .. \leq \#e_j^u \leq .. \leq \#e_{\#c_u}^u$. The *friendship degree* of the contact c of u is the rank j of c in this order. Hence, at step t, the *contact model* returns a predicted *friendship degree* $\widehat{fd}_t^u$ for the contact with whom the event e_t^u is done.

Contact LSTM model. The *contact model* is also a LSTM network that takes as input at step t, a feature vector x_t per user. It generates as output the predicted *friendship degree* $\widehat{fd}_t^u$. The network parameters training minimizes the Mean Absolute Error (MAE) of the training data.

Features x_t**.** At step t, the features are: the temporal information of e_t^u (cf. §III-A) except the *SOD*, the one-hot encoded event-type e_t^u, and the number of contact of u, $\#c_u$. This later is constant throughout a user sequence and is essential to help the model captures that $\widehat{fd}_t^u \leq \#c_u$. Accordingly, it is not encoded and is left to its actual value.



(a) (b)

Fig. 2: (a) Avg call duration (s) per hour and friendship degree (b) Call duration CDF for *RefCDRs*.

D. Metric modeling

This section presents the models used to generate the metrics (i.e., a model per metric) associated with events generation, namely the call duration and the data volume.

Call duration. We use a statistical method to model the call duration. In fact, contrary to the previously modeled parameters, there is no explicit features dependency or variability (and therefore, no complexity) regarding call durations, which implies that a used RNN could hardly train. This is confirmed in Fig. 2a, which shows the variation of the average call duration per hour and per friendship degree over the entire dataset. We can see that overall, call duration does not vary much, and thus, there is no particular correlation between these parameters. Moreover, the per-user behavior regarding call duration (easily assessed through the average call duration per user) closely depends on the number of calls each user makes over the CDRs duration, which is opportunely already captured by the *IET model*. Accordingly, the *call duration model* corresponds to the estimation of the parameters of the continuous distribution that best fits the empirical distribution of call duration, as shown in Fig. 2b. From a statistical test, we found this distribution to be Lognormal of parameters $\sigma = 1.29, \mu = 3.78, x_0 = -0.47$.

Data volume. The *data volume model* returns a data volume value for each data event. According to 3GPP standards, each data-typed CDRs line corresponds to the generation of a data session by a user. Unfortunately, as *RefCDRs* lack this information, we rely on the study done in [5] to design the *data volume model*. To the best of our knowledge, [5] is the only work that conducts a thorough characterization of data volume usage per session and per user over time extracted from real-world CDRs, as well as designs a generator of realistic CDRs that conforms to these characterizations.

[5] profiled users' data usage over time according to their generated amount of data (*volume profile*, i.e., Light, Medium, or Heavy) and to how often they generate data sessions (*frequency profile*, i.e., Occasional or Frequent). Besides, it extracted from real-world CDRs the distributions of data session volume according to a user's profile and the day period (peak or off-peak hours) and the percentage of users per profile. We use such percentages to first assign a *volume profile* to each user in *Zen*. As the *frequency profile* could be

inconsistent with the frequency of data event-type as predicted by the *event-type model*, we attribute to each user, in *Zen* the *Occasional frequency profile*. In fact, the distribution of the number of data sessions per day and user from *RefCDRs* shows the majority of the population to be of this latter profile. Finally, we sample from the distributions found in [5] to get a data volume.

IV. Implementation Details

Generating CDRs from *Zen*'s trained models first requires defining a network structure by setting users phone numbers and defining contacts. Then, through model inference, each well-defined user produces a sequence of timestamped events over the total duration.

Network structure. First, we generate a phone number in the format `<MCC><MNC> <5 random digits>` for each user. MCC and MNC respectively describe the mobile country code and the mobile network code, taken as parameter.

Then, we create the social graph of users' interactions. Here, we rely on the distribution of contacts per user from *RefCDRs*. Let $u \in U$ be a user with $\#c_u$ contacts; we consider the non-parametric distribution $P_{\#c} = P(\#c_u = \#c) \quad \forall\ \#c \in [1, MAX]$. For each generated user u', its number of contacts $\#c_{u'}$ is obtained with the multinomial distribution of parameters $P_{\#c}$. We further define four disjoint categories of contacts: international contacts (c_{inter}), outgoing local contacts (c_{out}), incoming local contacts (c_{in}), and both outgoing and incoming local contacts (c_{both}). Thus, $\forall\ u \in U,\ \#c_u = \#c_{inter,u} + \#c_{out,u} + \#c_{in,u} + \#c_{both,u} = (x_{inter,u} + x_{out,u} + x_{in,u} + x_{both,u}) \times \#c_u$. We export the average values $\overline{x_{cat,u}} \quad \forall\ cat \in \{inter,\ out,\ in,\ both\}$. and use the multinomial distribution of $P = \overline{x_{cat,u}}$ to induce the number of contacts, in each category, for each user. Such categories allow us to distinguish users generating only mobile data events (i.e., with no contacts), or users making no international calls (i.e., no international contacts), etc. By implementing a variant of the configuration model algorithm, we obtain a graph from given users degrees, describing users likely to interact through voice calls or SMS.

Model inference. Using *event-type* and *IET models* traffic models, we first generate timestamped sequences of events over the total duration. Then, each sequence is associated with a user based on its number of contacts per category, indicating which event-types the user can generate. Next, the *contact model* inference gives a contact friendship degree per user event that is later associated with the corresponding contact's phone number from users' phonebooks. At last, we add complementary metrics to users' events. For all calls events, the call duration metric relates only to available contacts of users. We do not consider unavailable users' contacts (i.e., already in an ongoing communication) at the caller-callee association. Hence, for available contacts, a call duration value is sampled from the *call duration* model distribution. This value is upper-bounded by the time to the closest scheduled call. As well, for data events, the data volume metric is assigned according to the *data volume* model.

V. Evaluation Results

This section confirms *Zen*'s validity by evaluating traffic models performance and generated CDRs applicability.

A. Traffic Models

Hereafter, we evaluate the accuracy and performance of *Zen*'s traffic models. As there is no similar contribution in the literature, we make comparisons with designed baseline predictors. Table II summarizes all comparison metrics and provides their distributions on the right of each result.

1) ***Experimental datasets****:* We consider as *training set* the first two weeks of the *RefCDRs* dataset, the 3rd week as *validation set*, and the 4th week as the *test set*.

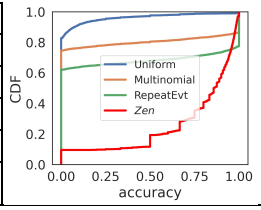
2) ***Models training and Hyper-parameters****:* We used a 2-layer LSTM with 50 hidden units per layer for the *event-type model* and 100 hidden units per layer for the two other models. To avoid over-fitting the training dataset, we used a dropout regularization with $p = 0.2$. The LSTM losses are iteratively minimized using mini-batch gradient descent with the Adam optimizer. Each mini-batch contains 64 sequences of events (i.e., users). We chose event sequences' lengths of 302 for training, 157 for validation, 159 for test, sampled from the distribution of the number of events generated by users in each experimental set. Therefore, we pad all sequences to the sequence length in each experimental set to homogenize datasets and ease the training. We use a masking layer to tag added values in each sequence to ignore them in the loss calculation. Besides, we fixed a gradient clip value of 0.01 to avoid "exploding gradients" prone to affect RNN.

3) ***Event-type model****:* We compare our *event-type model*'s predictions (cf. §III-A) to the ones of the following baselines: *Uniform* – each event-type is equally likely to occur at each time step; *Multinomial* – each event-type probability is given by its empirical count in training data; *RepeatEvt* – the next event-type is always predicted to be the same as the previous one. We use the following evaluation metrics: (*NLL*) Negative-log-likelihood of next-step probabilities, and (*Accuracy*) next-step 1-best correct classification rate (for this metric, the traditional Multinomial approach always output the most frequent event-type). Results are presented in Table II. Selecting event-type according to the *Multinomial* is significantly more predictive than the *Uniform*, but worse than *RepeatEvt*. Our *Zen*'s *event-type model* works the best. For both NLL and Accuracy, *Zen* is significantly better than *RepeatEvt*, i.e., the most probable event-type is not always the previous one.

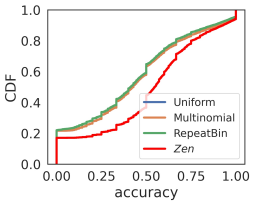
4) ***IET model****:* As before, we compare the acuteness of our model in predicting the next IET Bin (cf. §III-B) with the corresponding above-defined baselines. Table II shows that for both metrics, NLL and Accuracy, the performance of *Zen*'s *IET model* is much higher than *RepeatBin* (that simply repeats the previous IET Bin), followed by the Uniform and the *Multinomial* baselines. Disregarding the prediction approach, we compute the discretized probabilities of IET Bins and map them to IET values in a continuous domain: named *Bin sampling* mapping. To evaluate how efficient *Zen*'s and

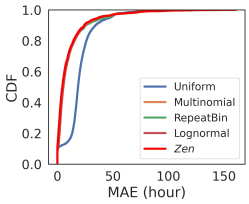
TABLE II: *Traffic LSTM models evaluation results.*

Event-type model		
Predictor type	NLL	Accur.
Uniform	0.27	2.91%
Multinomial	0.21	38.97
RepeatEvt	N/A	43.27
Zen	**0.037**	**91.82**

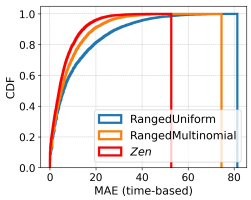


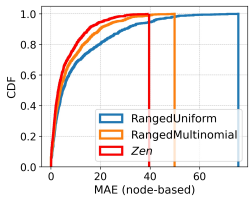
IET model						
Predictor type	NLL	Accur.	MAE	MAE]0, 30 min] (82.8%)	MAE]30min, 24h] (15.45%)	MAE >24h (1.75%)
Uniform	0.215	64.56	1097	1033	1120	2319
Multinomial	0.165	64.56	231	68	334	2877
RepeatBin	N/A	58.05	239	73	347	2871
Lognormal	N/A	N/A	249	78	361	2973
Zen	**0.118**	**69.25**	**185**	**16**	**295**	**2904**





Contact model								
Predictor type	MAE (time-based)				MAE (user-based)			
	All	[1,6]	]6,21]	>21	All	[1,6] (50%)	]6,21] (30%)	>21 (20%)
Ranged-Uniform	25.12	1.17	5.04	29.57	26.38	1.08	4.53	31.17
Ranged-Multinomial	15.78	0.91	3.87	18.42	17.28	0.81	3.41	20.38
Zen	11.81	0.65	3.02	13.77	13.23	0.63	2.57	15.68





baselines' *Bin sampling* are, we compare them to the *Overall sampling* mapping, both described next.

- *Bin sampling*: At each Bin, the IET value is obtained after averaging $n = 500$ samplings of the corresponding continuous IET distribution (see §III-B). We apply this approach to all the previously Bin-based models, i.e., *Zen*'s *IET model*, Uniform, Multinomial, and RepeatBin predictors.
- *Overall sampling*: We perform a fitting of the empirical IET distribution (i.e., with no bins) and obtain a Lognormal distribution with $\sigma = 2.67, \mu = 4.97, x_0 = 1$. Then, we straightly predict continuous values by sampling the resulted fitted IET distribution. We name this prediction *Lognormal*.

The Mean Absolute Error (MAE) of the IETs in minutes is used as the comparison metric. It estimates the average distance between actual and predicted IET.

From Table II, we can notice that the *Bin-sampling* of *Multinomial* and *RepeatBin* have comparable MAE performances, followed by the *Overall-sampling Lognormal* predictor. This behavior is also verified per Bin (three last columns). Overall, *Zen* works the best. In the first bin $]0, 30min]$, which is the most sensitive, we note that except for the *Zen*, all models on average predict an IET value outside the initial interval.

5) ***Contact model****:* We evaluate the *contact model* (cf. §III-C) by comparing its predictions to the following baselines:

- *RangedUniform*: Per user u, contacts c_i, $\forall i = 1, 2, ..., \#c_u$ are equally likely to be predicted at each sequence step.
- *RangedMultinomial*: Per user u, each contact c_i is chosen with a probability $(p_i^u, 1 \leq i \leq \#c_u)$ extracted from the procedure as follows:

Let U be the set of users and u a user in U. We recall that $\#e_{c_i}^u$ refers to the number of events u made with his contact c_i. From this definition, we derive $P_{c_i}^u$ the proportion of events made by u with its contact c_i : $P_{c_i}^u = \#e_{c_i}^u / \sum_i \#e_{c_i}^u$. For all $i = 1, 2, ..., MAX(\#c_u)$ we extract the mean values $\overline{P_{c_i}} = \overline{P_{c_i}^u}$ $\forall u \in U$. Hence, for a user u, the probabilities $(p_i^u, i = 1, 2, ..\#c_u)$ is obtained by normalizing the first $\#c_u$ mean values $(\overline{P_{c_i}}, i = 1, 2, ..\#c_u)$ such that $\sum_i p_i^u = 1$.

The evaluation metric is the MAE of the predictions $\widehat{fd_t}$ in the test dataset. We found that as we train the *contact model* with chronologically-separated experimental windows (defined in §V-A1), the MAE loss value continually increases in the validation dataset. This is due to the fact that in the training period (i.e., first two weeks), users only interact with some of their contacts, making it difficult for the model to generalize. To fix this issue, we instead split training, validation, and test datasets by selecting users traffic over the whole dataset period (4 weeks). The training dataset includes 60% of the users, while the validation and test datasets each represent 20%. Results in Table II show the *RangedMultinomial* predictor has significantly better results compared to *RangedUniform*.

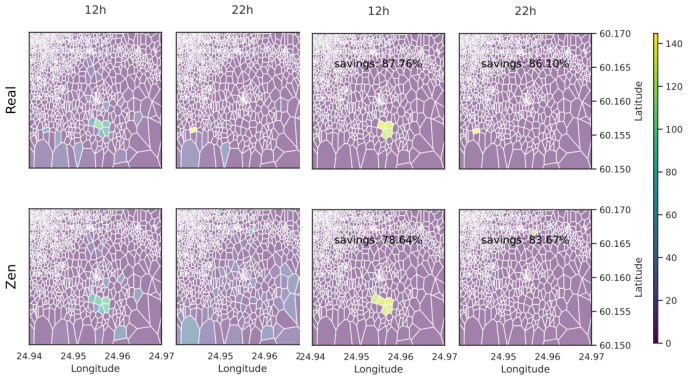
Overall, *Zen* is the modeling that best performs, showing its ability to capture users interaction with their contacts. In particular, the detailed distribution plots show *Zen* presents for 80% of users (i) more than 95% and 75% of accuracy for respectively, the event-type and IET models, and (ii) less than 6.68% and 12.5% of MAE maximum values for respectively, the IET and contact models.

B. Zen CDRs use cases

We evaluate CDRs resulting from *Zen* framework as compared to *RefCDRs* when applied to two use cases. We generate for a week period *Zen* CDRs with 6000 users, corresponding to the same number of users in *RefCDRs* (see §V-A1).

Data-Driven Micro BS Sleeping. Numerous works studied power savings in Radio Access Networks (RAN). We investigate how a traffic-aware Base Station (BS) on/off-switching strategy [8] performs when informed with *Zen* CDRs compared to *RefCDRs*. To this end, we enrich both CDRs datasets with emulated user trajectories in the Helsinki UE city, using the realistic Working Day Mobility model [9]. Such process supplies each CDRs dataset with users' cell Id position at each event generation. We assume a heterogeneous RAN deployment where each cell is served by a separate micro BS, whereas macro BSs provide umbrella coverage to a larger area. Specifically, we consider a grid tessellation of 5X5 macro BSs in the considered zone. The power needed to the operation of a BS at time t is $P(t) = N_{trx}(P_0 + \Delta_p P_{max}\rho(t))$, $0 \leq \rho(t) \leq 1$, where $\rho(t)$ is the relative traffic load at time t with P_0, N_{trx}, P_{max} and Δ_p being constants defined for micro and macro BSs in [7]. Then, if $\rho(t) \leq \rho_{min} = 0.37$ the micro BS offloads its local traffic to the macro BS and goes into sleep mode, where it consumes negligible power. Accordingly, Fig. 3 shows the power consumption ($P(t)$ values in the color bar) of each cell's micro BS at two hours in Helsinki (a zoomed-in area of 2.2km $\times$ 1.6km) with and without such a strategy implemented. We can see that comparable cells are kept on, while the strategy brings similar energy savings.

Anomaly detection. The fine-grained state of *Zen* CDRs allows for the investigation of per-user temporal behavior for cellular anomaly detection. For instance, SIMBox fraud is a prevalent scam in telecommunication networks consisting of "fake" user accounts re-injecting diverted international calls as local calls to a country [10]. We assess the utility of *Zen* CDRs for investigating such fraud by applying a user profiling method where traffic or mobility users' behaviors are leveraged to classify a user as fraudulent or not. To this end, we apply for both *Zen* and real ones, a DBSCAN clustering to a set of per-user traffic-related features specific to detect SIMBox fraudulent behavior as described in Table 1 of [11]. Results show a similarity between *Zen* CDRs and real-world ones: while *RefCDRs*' estimated number of clusters and outliers are 10 and 1241, Zen CDRs' confidence intervals for these metrics are 9.1 ± 1.66 and 1122.3 ± 35.02 for 10 samples of *Zen* CDRs' call duration feature (ref. §III-D).



(a) Always-active micro BS. (b) Cell-sleeping strategy.

Fig. 3: Power consumption per cell (a) for always-active micro BS and (b) with a cell-sleeping strategy.

VI. Conclusion and Discussion

This paper presented *Zen*, the first framework allowing the autonomous generation of complete and realistic traffic CDRs in an individual basis. We relied on a fully anonymized traffic CDRs to provide the first model that captures long-range and inter-CDRs traffic features correlation, individuals heterogeneity and social-ties in communication. Finally, we validated *Zen* realisticness in reproducing daily traffic behaviors of individuals and usefulness in practical networking applications. Next, we provide some extra discussions.

Generalization: Though *Zen* provides realistic traffic behavior models trained from a unique real-world traffic CDRs, the modeling methodology of this paper is general and can be applied to other CDRs with different cultural traffic habits.

Future improvements: *Zen* is still open for improvements, such as adding complementary mobility features to the generated CDRs. Yet, this implies coping with privacy issues related to individual-based mobility modeling of real-world CDRs.

References

[1] D. Rhoads, I. Serrano, J. Borge-Holthoefer, and A. Solé-Ribalta, "Measuring and mitigating behavioural segregation using call detail records," *EPJ Data Science*, vol. 9, 12 2020.

[2] M. Ozturk, A. I. Abubakar, J. P. B. Nadas, R. N. B. Rais, S. Hussain, and M. A. Imran, "Energy optimization in ultra-dense radio access networks via traffic-aware cell switching," *IEEE Trans. on Green Communications and Networking*, vol. 5, pp. 832–845, 2021.

[3] M. Songailaitė and T. Krilavičius, "Synthetic call detail records generator," *CEUR Workshop proceedings*, vol. 2915, 2021.

[4] B. Hughes, S. Bothe, H. Farooq, and A. Imran, "Generative adversarial learning for machine learning empowered self organizing 5g networks," in *ICNC*, 2019, pp. 282–286.

[5] E. Mucelli Rezende Oliveira, A. Carneiro Viana, K. Naveen, and C. Sarraute, "Mobile data traffic modeling: Revealing temporal facets," *Computer Networks*, vol. 112, pp. 176–193, 2017.

[6] Z. Lin, A. Jain, C. Wang, G. Fanti, and V. Sekar, "Using gans for sharing networked time series data: Challenges, initial promise, and open questions," in *IMC*, 2020, p. 464–483.

[7] K. Xu, R. Singh, M. Fiore, M. K. Marina, H. Bilen, M. Usama, H. Benn, and C. Ziemlicki, "Spectragan: Spectrum based generation of city scale spatiotemporal mobile network traffic data," in *CoNEXT*, 2021, p. 243–258.

[8] G. Vallero, D. Renga, M. Meo, and M. A. Marsan, "Greener ran operation through machine learning," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 896–908, 2019.

[9] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working day movement model," in *ACM SIGMOBILE MobilityModels*, 2008, p. 33–40.

[10] A. J. Kouam, A. C. Viana, and A. Tchana, "Simbox bypass frauds in cellular networks: Strategies, evolution, detection, and future directions," *IEEE Communications Surveys Tutorials*, vol. 23, pp. 2295–2323, 2021.

[11] R. Sallehuddin, S. Ibrahim, A. Zain, and H. Elmi, "Detecting sim box fraud by using support vector machine and artificial neural network," *Jurnal Teknologi*, vol. 74, pp. 137–149, 04 2015.