Forecasting of the Urban Area State Using Convolutional Neural Networks

Ksenia D. Mukhina, Alexander A. Visheratin, Gali-Ketema Mbogo, Denis Nasonov
ITMO University
Saint Petersburg, Russia
{mukhinaks, alexvish91, ketema.galy, denis.nasonov}@gmail.com

***Abstract*—Active development of modern cities requires not only efficient monitoring systems but furthermore forecasting systems that can predict future state of the urban area with high accuracy. In this work we present a method for urban area prediction based on geospatial activity of users in social network. One of the most popular social networks, Instagram, was taken as a source for spatial data and two large cities with different peculiarities of online activity – New York City, USA, and Saint Petersburg, Russia – were taken as target cities. We propose three different deep learning architectures that are able to solve a target problem and show that convolutional neural network based on three-dimensional convolution layers provides the best results with accuracy of 99%.**

I. Introduction

Active progression and widespread acceptance of a Smart City concept leads to the need for development of systems capable of accurately forecasting a future state of the urban environment [1]. However, such decision support systems are usually limited by the domain and aimed at analyzing a particular type of events, for example, floods [2], evacuation [3], or transportation of hazardous substances [4]. Such systems in the most cases are incapable of observing the whole picture or sharing information with other systems. Thus there is the need for development of a general purpose forecasting system. Complex forecasting of the urban area state requires usage of various data sources, which may be incomplete [5]. This problem can be solved by using a wide range of new data sources, such as mobile phone records, social network data, or streaming cameras [6], or by adding more sensors of various kinds to the system [7] that in turn would result in a cost growth. Significant part of research based on a data from social networks and news is devoted to detection of critical situations, such as fires, hurricanes, and terrorist attacks [8]. Moreover, usage of social networks also makes it possible to identify and predict cultural events [9], such as a football match [10] or a concert [11]. By this means, there is no surprise that data from social networks and news media became one of the valuable additions to input data of modern decision support systems [12].

An active user of a social network can act as an anomaly sensor by increasing his or her activity in case of an unusual situation [13]. Having data about residents activity in an urban area, researchers would be able to extract information about current situation and detect potential events as anomalies in users behavior. The most convenient way to determine an event is to observe for the set of predefined keywords or hashtags [14], [15]. Within a scope of social networks, a sharp rise in interest for a particular topic would be a sign of an event of some sort. Nevertheless, this approach limits monitoring and forecasting systems to expected events only. That is why it is essential to develop methods, which can determine the next state of the city independently from a specified context.

One way to exclude the context of messages is to use other available data, like geographic coordinates, text sentiment and photos. Knowledge of a general state of activity in the city at any moment can help in various tasks from city planning to effective emergency monitoring. In this work, we aim to forecast an urban area state using historical data of users activity in social networks. To achieve that, we studied performance of three different convolutional neural network architectures for two large cities – Saint Petersburg, Russia, and New York City, USA.

The main contributions of the article are as follows:

1) We propose three deep neural network architectures designed to perform prediction of the urban area state using a set of previous states.
2) We perform an extensive experimental comparison of the described convolutional neural networks using two large datasets of geo-located posts from Instagram.
3) We tune parameters of the best performed model, which is based on three-dimensional convolutional layers, to achieve an accuracy of 99% compared to the ground truth data.

II. Related works

Spatial analysis methods have been successfully used for city analysis in decision support systems that utilized data from mobile phones [16] and social networks [17], [18]. Yao et al. [19] proposed the DeepSense framework for mobile sensing, which combines convolutional and recurrent neural networks for a tracking system. In this paper convolutional layers were successively used to extract patterns of frequency and interactions among sensors. Zhou et al. [20] used check-in data from social network WeChat in a framework for cultural planning in Beijing. The solution was based on a temporal latent Dirichlet allocation used for identification of cultural patterns and OPTICS - ordering points algorithm to select essential clusters.

Short-term predictions can be successfully implemented using an autoregressive model [21], however, for a more global and long-term predictions of the city state, neural networks are

actively used nowadays [22], [23]. In [24] it was demonstrated that logistic models and neural network show comparable results, but the neural network achieves better accuracy (up to 85%) compared to the logistic regression (up to 74.9%) for forest fires forecasting in Portugal. As of today, machine learning techniques are widely used for various predictive tasks. For example, gradient boosting decision trees were used for water break predictions in a three years perspective [25]. In [26] Adams et al. proposed a framework based on artificial neural network models for air pollution risk forecasting. Authors constructed three-layer perceptron network and achieved root mean squared error - 3.5 $\frac{\mu g}{m^3}$ for particulate matter and 18.8 $\frac{\mu g}{m^3}$ for nitrogen dioxide.

Usually, CNNs are used for image classifying or feature prediction such as object attributes (color, type of clothes), gender [27], age [28], and even more complex task like human actions [29] or depth maps [30]. In [31], Kang et al. used CNN to extract features from images to predict crime occurrence. The method achieved 84.25% accuracy with precision of 74.35% and recall of 80.55%. In [32] Zhang et al. used convolutional layers to extract spatiotemporal features as part of the deep neural network for citizens flow prediction. Despite the fact the deep neural network DeepST proposed by authors outperformed state-of-the-art models, it was also shown that CNN itself provides sufficient results on a bike rent data. Authors continued their work in [33] and presented a new deep learning based approach ST-ResNet. This approach considers additional factors in the model and performs at least 6% better than the closest competitors. In [34], authors combined CNN and Long Short-Term Memory (LSTM) networks to forecast two types of crowd flow: outcome and income. Outcome and income were interpreted as two channels of an image, and a two-dimensional convolution was used to extract spatial features. Authors compared their model with several baseline approaches such as ARIMA, LSTM and one-dimensional convolution Conv1DNet. The proposed model showed the lowest root mean squared error (31.57) and outperformed follow-up approaches Conv1DNet (33.76), ARIMA (35.47), and LSTM (43.65). In [35], a pure CNN was used to predict anomalies in a crowd flow. It was shown that larger dataset provides better and more stable results during the training and in prediction quality.

However, it was shown that CNN itself can predict sequences as efficient as a more popular approach - recurrent neural network (RNN) [36]. The proposed solution genCNN beats the closest competitor LSTM with over 25 points margin. In [37] deep convolutional neural networks were used to evaluate next move in Go game. The system provides an accurate prediction of the expert move for 55% of positions. In [38], RNN and CNN architectures were compared for Atari game next state prediction. Results showed that both models provide comparable results where CNN predicted states of the Ms Pacman game with less mean squared error but RNN was more accurate for Space Invaders. Another comparison between LSTM and CNN were performed for protein sequence prediction [39]. It was shown that CNN Q8 accuracy was higher than others: 0.684 compared to 0.674 for LSTM.

Xu et al. [40] used CNN for event detection in video. This approach allowed to increase mean average precision by 10 % to 36.8%. This idea was later expanded to the real-time event detection and prediction in [41]. Dependence of accuracy on the percentage of video observed was studied; the proposed approach shows satisfactory results on two datasets and usage of optical flow resulted in 5% quality improvement. In [42], authors presented a neural network architecture for Twitter users location prediction using texts, network topology and time zone information. The developed method achieved up to 69% accuracy for labelling the state and approximately 62% for 100-mile zone detection. However, this approach requires a lot of various incoming data and resources for preprocessing. Moreover, @161 metrics (100-mile zone $\sim$ 161 km) used for method estimation indicates the general user location (e.g. city) and cannot be used on a smaller scale.

Farajidavar et al. [43] proposed CNN for event detection in the city using Twitter stream. The authors used dataset labeled by experts into seven categories related to crime, cultural, food, social, sport, environment, and transport events. The averaged accuracy of event extraction is 81%. Despite the authors claim that almost 50% of the traffic comments appears approximately five hours before the official reports, this approach is limited to monitoring systems, and it is not able to forecast a state of the city.

As it can be seen, CNNs is a widely used approach for dealing with social networks data and various types of predictions. Since aggregated data from social networks can be interpreted as a frequency map of user activity, we decided to use CNN architecture as the most natural approach for social network state prediction. Moreover, we use three-dimensional convolution layers to take into account temporal trends.

III. Proposed approach

An urban area state in our work is represented by users posts in the social network Instagram within a specified period of time. Instagram is a fast-growing social network with more than 1 billion users all over the world according to the Verge (https://www.theverge.com/2018/6/20/17484420/instagram-users-one-billion-count). Due to its high popularity and data of various kinds (photos, texts, likes, geolocation), Instagram recently has drawn attention of researchers from different areas of urban studies [44].

A core schema of the proposed solution for the urban area state prediction is presented in Figure 1. On the first step, we collect a considerable amount of data in a target geographical area. On the second step we generate a set of aggregated city states from the data gathered. And after that we train a deep neural network model to predict a state of the city using a number of previous states (in this work we used 5 previous states) on a subset of the aggregated data, while validating the training process using another subset of the aggregated data.

Dataset description. Users of Instagram can share their location by selecting a place from the predefined list with names and addresses. Thus, geolocated data from the Instagram represents a sequence of posts related to a discrete set of places. The data for our research was obtained using a web crawler built upon Instagram GraphQL API. The dataset contains posts with location marks within two areas: Saint Petersburg, Russia, and New York City, USA. The area of Saint Petersburg dataset covers city center and consists of geotagged posts in the area [30.20, 59.87, 30.40, 59.98] for the period

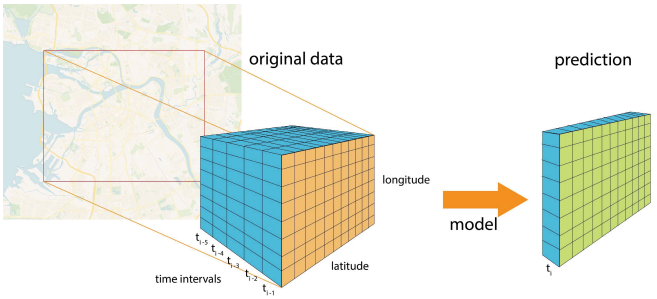

Fig. 1. Schema of urban area state prediction

from 1 January 2016 to 21 July 2017. The second dataset contains posts from New York City within coordinates [-74.06, 40.63, -73.857, 40.83] starting from 1 January 2017 till 13 May 2018. We kept only timestamp and location of each post. The data for every hour of collection periods was aggregated and placed on a geospatial grid with spacing 0.001'. The value of each cell corresponds to the number of posts in a given geographic area during a particular hour. Thus, we obtained 24 grids for every day of each dataset. We then converted each grid into a sparse matrix since the vast majority of cells contains zero posts.

We used the first 12 months for model training and the rest of the data (four and a half months for Saint Petersburg and approximately seven months for New York) were used for validation. Usage of full-year interval allows correcting seasonality. Because validation performs on the following year, we can make assumptions about model behavior with constant increasing of the number of Instagram users.

IV. Convolution Neural Networks

As it was previously discussed, convolutional neural networks are successfully used in various tasks from pattern recognition and image classification to event detection in video and social networks. Since our task is directly connected with the recognition of spatiotemporal patterns, we selected CNN as a target approach for a urban area state prediction. To obtain the best architecture for our task, we decided to compare three different methods and evaluate their performance. We started from the most common approach – usage of two-dimensional convolutional layers for frequency map analysis. The second approach takes into consideration temporal features of the data – we used three-dimensional convolutional layers that convolve input data not only by width and height, but also by depth that in our case represents time. The third approach utilizes 3D-convolutional and transposed 3D-convolutional layers in order to perform data compression and decompression with regard to all three dimensions.

We refer to each network based on the distinguished layer in its architecture. Thus we have three CNN named Conv2D-Net, Conv3D-Net, and TransposedConv3D-Net, respectively. Each neural network contains six convolutional layers with ELU as activation function, architectures of these networks is presented in Figure 2.

The first CNN architecture (Conv2D-Net, Figure 2a) contains only two-dimensional convolutional layers with kernel

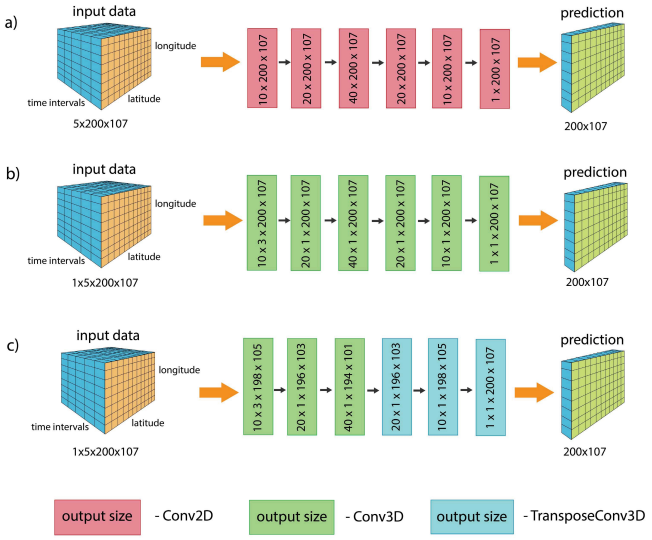

Fig. 2. Architecture of proposed CNNs: a - architecture based on two-dimensional convolutional layers, b - architecture based on three-dimensional convolutional layers, and c - architecture based on combination of three-dimensional convolutional and transposed convolutional layers.

size equal to 3. The idea behind this approach lays on the fact that spatial grids could be interpreted as images where longitude and latitude are equivalent of width and height and hours performs as channels. Stride and padding for convolutional layers were selected to preserve original size of the grid. The number of filters was defined as $5 \cdot K$ where 5 is a number of previous hours and $K = 2$ is an integer. The last layer reduces the number of filters to 1 as the size of the target map.

In the Conv3D-Net, we replaced two-dimensional layers with three-dimensional convolutional layers to reveal the activity patterns between hours (Figure 2b). To meet the final grid resolution, after two convolutions we changed the kernel size to $k = (1, 3, 3)$, the stride was equal to 1 and the value of padding was set as $p = (0, 1, 1)$ to prevent a size change along the x and y axes. The use of three-dimensional convolutional layers is a common approach for the analysis of temporal data. Since a three-dimensional convolution operates all dimensions, we added the fourth dimension to our data to define the filters. In this case number of filters will be multiplied to 1 that is why we decided to set the number of output filters for the first convolutional layer to 10. Thus, the number of filters for the first and second approach is the same. On the last step, we squeeze all empty dimensions to obtain a two-dimensional map.

In the third architecture, TransposeConv3D-Net, we used transposed convolutional layers instead of last three convolutional layers (Figure 2c). In this approach, a pair of convolutional and transposed convolutional layers ensures that the output size of prediction will be the same as the input. That is why we kept the default value of padding, which is equal to 0. Since transposed convolutional layers used for map size restoration, kernel sizes were equal to $k = (1, 3, 3)$ for every transposed layer. The stride was kept to its default values.

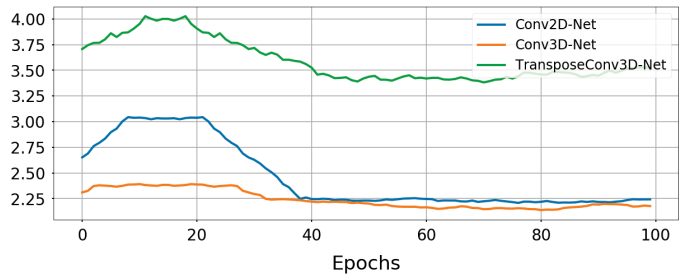

Fig. 3. Comparison of different architectures performance for New York

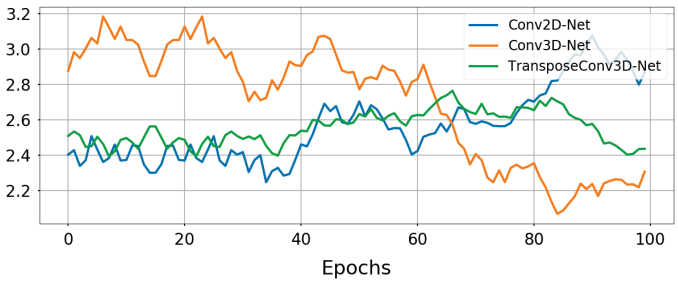

Fig. 4. Comparison of different architectures performance for Saint Petersburg

To evaluate performance of different models, we propose the following function:

$$ev = \frac{1}{N}\sum_{i,j}^{N}\frac{\mid y_{ij} - \hat{y}_{ij} \mid}{\hat{y}_{ij}} \cdot 100\%, \quad (1)$$

where y_{ij} - element of the prediction matrix and $\hat{y}_{ij}$ represents the actual data. A result of this function can be interpreted as an average deviation percentage of the predicted map, where the lesser value represents better accuracy of prediction. This function was designed to ensure that the neural network will learn to correctly predict areas with a high activity as well as areas with a low activity.

V. Experiments

All experiments were implemented with using PyTorch framework of version 0.4.1 [45]. Workstation was equipped with NVIDIA Tesla P100.

A. Architecture comparison

For this experiment we trained models based on aforementioned architectures twice separately for each city (Saint Petersburg and New York City). For a clear comparison of architectures we used the same optimizer - Adadelta algorithm [46] with a starting learning rate $lr = 1$. All models were trained for 100 epochs with the batch size equals to 1. And L1 criterion was used during the training.

In Figure 3, results of the evaluation function are presented for networks trained on the New York dataset. As it can be seen from the plot, TransposeConv3D-Net demonstrates the worst results from the very beginning with the highest starting deviation. Despite a clear declining trend from the 18th epoch until the 50th epoch, the network based on this architecture showed the worst results during the whole training. Conv2D-Net demonstrated a sharp drop in error after 20th epoch, but the minimum achieved at 47th epoch was still slightly higher than results of Conv3D-Net. In spite of the lesser changing during the training period, the best results were achieved by Conv3D-Net.

Figure 4 illustrates results obtained for training on the Saint Petersburg dataset. In the beginning, the 3D-convolutional network had a higher value of deviation and this tendency continues until the 65th epoch. Despite the mediocre beginning, Conv3D-Net showed a sharp decline in average deviation from the actual state and after 85th epoch scored the better results. It is clear from the chart that Conv2D-Net starts with the best results, however, the average deviation just slightly changes during the first half of training. After that, a deviation starts to increase, and this tendency is lasting until the end of training. TransposeConv3D-Net demonstrates the similar tendency: the first deviation is close to the value of Conv2D-Net case and deviation continuously grown until the 82nd epoch.

It is important to note that all models demonstrated worse results in the Saint Petersburg dataset. This effect happened due to fact Instagram users are less active in Russia than in the USA, and raw data for Saint Petersburg contains more empty cells with zeros, which complicates the modeling process. Interesting fact is that architecture that uses transposed convolutional layers performed worse than architecture based on convolutional layers in both cases. Because we have only five previous hours ($depth = 5$), first two convolutions reduce input data depth to one and next process is an equivalent to two-dimensional convolutions. Since in both cases Conv3D-Net have demonstrated the best results, we selected Conv3D-Net as our primary approach for further tuning and prediction.

B. Model tuning

During the first part of model tuning we examined several popular loss functions L1 loss, Smooth L1 loss, and root mean squared error (RMSE) [47], [26]. Results presented in Figure 5 show that L1 provides better results for the New York dataset. Despite the lesser decline in the average deviation, model with Smooth L1 loss function performs better for the Saint Petersburg dataset. It should be noted that change of loss function allows to improve accuracy for Saint Petersburg to almost 1% of an average deviation from ground truth data, which is twice better than the best result for New York. Thus, the further experiments were conducted with L1 loss and Smooth L1 loss functions for New York and Saint Petersburg, respectively.

In the next experiment, we varied the batch size in the range [1, 4, 8, 16, 32]. As can be seen from Figure 6 with an increment of batch size there is more clear declining trend in deviation during the training. However, during training process results achieved with different batch sizes become even. The minimal batch size allows to obtain the best accuracy, but in case of Saint Petersburg bigger batch ensures more stable behavior during the training. The controversial tendency could be observed for New York dataset where the largest batch size leads to a significant overfitting. Besides all that, since the

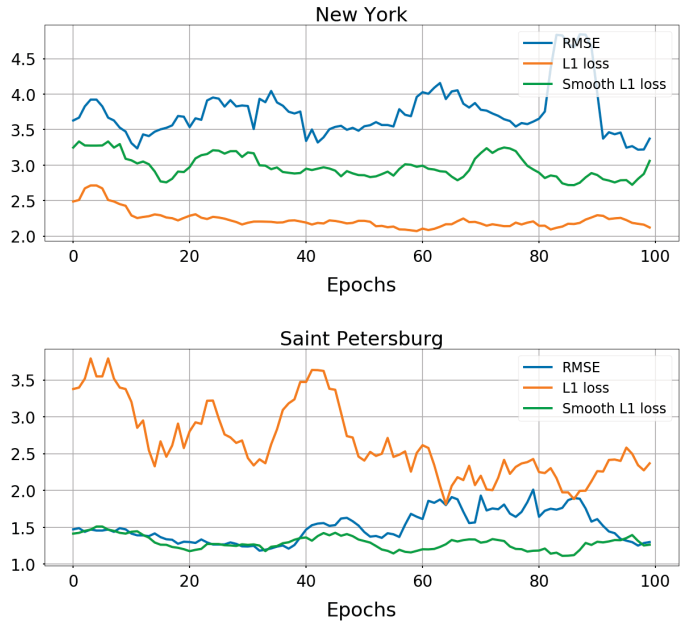

Fig. 5. Effect of different loss functions on CNN performance

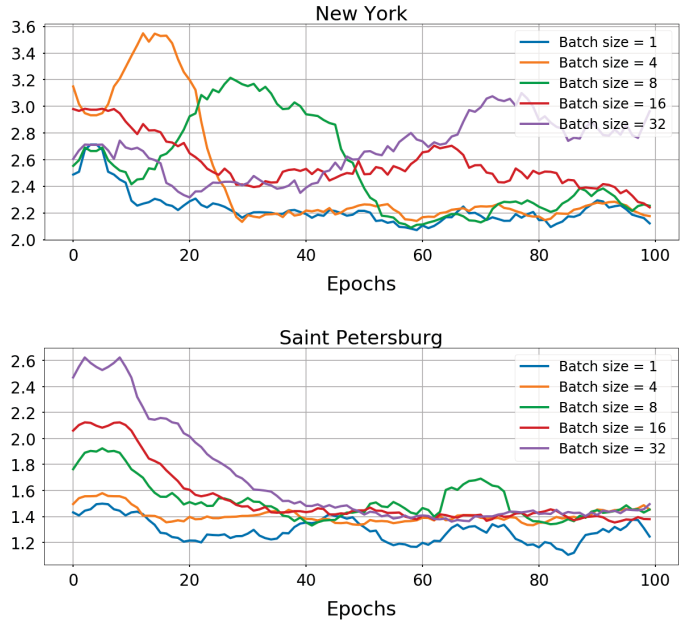

Fig. 6. Effect of different batch size on CNN performance

minimal batch size provided the best results for both cases, we decided to keep batch size equal to 1 for the next experiment.

In the last experiment we trained our model using different initial learning rates. Since the optimization strategy used in our research – Adadelta – dynamically changes learning rate, our task was to select an appropriate initial value. Figure 7 illustrates results obtained with initial learning rates $lr = 1$, $lr = 0.7$, $lr = 0.5$. For both cities, smaller value of learning rate produced better results with higher accuracy.

Thus, after the tuning process, we identified the following parameters for the best results for both cities.

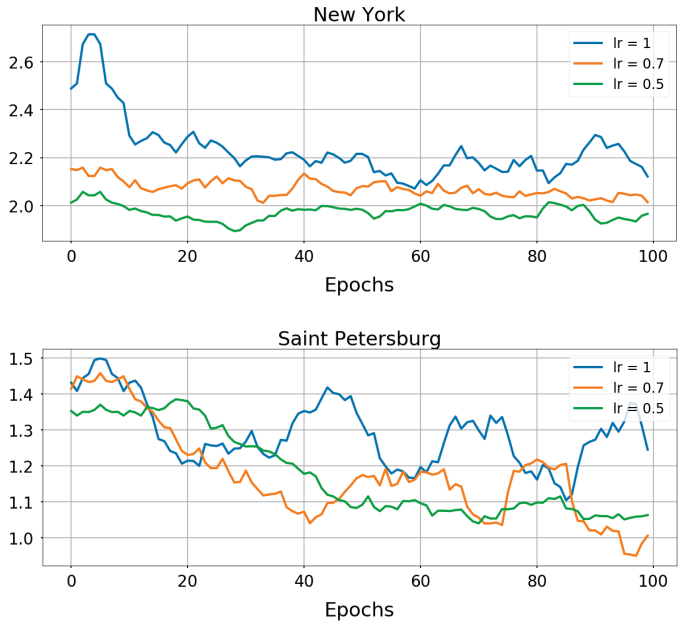

Fig. 7. Effect of different initial learning rate on CNN performance

1) New York (the best average deviation – 1.825%): L1 loss function, $batch = 1$, $lr = 0.5$;
2) Saint Petersburg (0.946%): Smooth L1 loss function, $batch = 1$, $lr = 0.7$.

C. Results

Figure 8 illustrates examples of results for both cities. As it can be seen from the maps, convolution neural networks reproduce areas with zero and high activity correctly. It can be noticed that for Saint Petersburg high activity areas concentrate in the city center and there are only few such clusters. Active areas are placed near very popular places like the main street or subway stations, which are highlighted even during the late hours. Despite some divergence in activity level for areas with medium activity level, the final accuracy equals to 99% for Saint Petersburg.

New York dataset contains one cell with very high value of activity, which required more time for training and resulted in a lower value of accuracy. One of such cell represents the default point of New York City and references the whole city itself. The light green areas that can be seen on the ground truth data contains less than 5 posts in each cell. Thus, due to the high variance in activity between the cells, more accurate prediction in New York requires methods, which would be able to ensure both high active areas and areas with zero activity. Nevertheless, accuracy achieved by usage of three-dimensional convolutional layers equals to 98%.

VI. Conclusion and Future Works

In this work, we proposed a new approach to forecast urban area state by using CNN. The urban area state was defined as activity level of users from the social network Instagram. We took two cities with different Instagram activity – New York and Saint Petersburg – for demonstration of applicability of

Fig. 8. Examples of prediction for one hour

our method. Presented model takes into consideration users activity for 5 previous hours and forecasts the next state. To achieve the best results, we tested three different architectures based on different types of convolutional layers – two-dimensional convolution, three-dimensional convolution, and three-dimensional transposed convolution. It was demonstrated that three-dimensional convolution provides the best result for both cities. After model tuning by variance of loss functions, batch size and learning rate, the best model predicts the next activity state with up to 1% of average deviation from the ground truth data.

It should be noted that during this work we assumed that five previous hours are enough for the next state prediction. But the size of a retrospective window should be studied more since using five-hour window may lead to noise caused by daily patterns. On the other hand the shorter window size will lead to errors triggered by event which took place in previous hours. Five-hour windows was chosen due to the strong need in balance between daily activity cycle and mass events. Thus, the chosen window seems optimal to decrease influence of noise caused by different reasons.

However, this approach has some limitations. First, the high value of accuracy (99% for Saint Petersburg and 98% for New York) is ensured by large number of cells with zero activity in both cities. This effect occurs due to the way of location representation in Instagram. Thus, the prediction of activity state for datasets with precise coordinate, for example, Twitter or Vkontakte, requires further studies. Nevertheless, the sufficient results achieved for the New York dataset with higher and more solid activity areas allow us to expect the good performance in such cases.

Another peculiarity of these results is a low variance between beginning and end of training with a lower bound in 1%. This could be explained by limitation of convolutional methods prediction ability. As it was discussed in the Related Works section, CNN provides better results in some prediction task when in another problems LSTM performs better. Since the comparison of different approaches for prediction was out of the scope of this study, comparison of different recurrent neural networks architectures is one of possible directions of future works.

Another way to improve this work is to use combination of data from different sources. In this paper, we aimed to predict activity state for one social network, but usage of various datasets allow to study and forecast real behavior of citizens in the city. Since our approach does not rely on specific features of used data and based on a spatial grid, it can be easily expanded to different social networks or even news media. The open web cameras can be used to enhanced the input data and to adjust actual activity rate in different city areas.

Despite all further improvements, the model proposed

in this work showed solid performance for the city state predictions in two cities with accuracy of 99%. Thus, it can be concluded that three-dimensional convolutional networks can be successfully used for spatial predictions based on retrospective data.

ACKNOWLEDGMENTS

This research is financially supported by The Russian Science Foundation, Agreement #18-71-00149, and by RFBR according to the research project #18-37-00076/18.

REFERENCES

[1] L. Filipponi, A. Vitaletti, G. Landi, V. Memeo, G. Laura, and P. Pucci, "Smart City: An Event Driven Architecture for Monitoring Public Spaces with Heterogeneous Sensors," in *Proceedings - 4th International Conference on Sensor Technologies and Applications, SENSORCOMM 2010*. IEEE, 7 2010, pp. 281–286.

[2] H. Mirfenderesk, "Flood emergency management decision support system on the Gold Coast," *The Australian Journal of Emergency Management*, vol. 24, no. 2, 2009.

[3] R. E. Mohamed, E. Kosba, and K. Mahar, "A Framework for Emergency-Evacuation Planning Using GIS and DSS," *Lecture Notes in Geoinformation and Cartography*, pp. 213–226, 2018.

[4] K. G. Zografos, G. M. Vasilakis, and I. M. Giannouli, "Methodological framework for developing decision support systems (DSS) for hazardous materials emergency response operations," *Journal of Hazardous Materials*, vol. 71, no. 1-3, pp. 503–521, 1 2000.

[5] C. De Maio, G. Fenza, M. Gaeta, V. Loia, and F. Orciuoli, "A knowledge-based framework for emergency DSS," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1372–1379, 12 2011.

[6] J. Borges, P. Bozsoky, S. Sudrich, and M. Beigl, "Advances in event detection," *Proceedings - 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017*, vol. 2018-Janua, pp. 1017–1024, 2018.

[7] F. Corno, T. Montanaro, C. Migliore, and P. Castrogiovanni, "SmartBike: An IoT crowd sensing platform for monitoring city air pollution," *International Journal of Electrical and Computer Engineering*, vol. 7, no. 6, pp. 3602–3612, 2017.

[8] D. Pohl, A. Bouchachia, and H. Hellwagner, "Automatic sub-event detection in emergency management using social media," *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*, p. 683, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2187980.2188180

[9] H. Gu, X. Xie, Q. Lv, Y. Ruan, and L. Shang, "ETree: Effective and efficient event modeling for real-time online social media networks," *Proceedings - 2011 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2011*, vol. 1, pp. 300–307, 2011.

[10] H. Abdelhaq, C. Sengstock, and M. Gertz, "EvenTweet: Online Localized Event Detection from Twitter," *Proceedings of the VLDB Endowment*, vol. 6, no. 12, pp. 1326–1329, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2536274.2536307

[11] J. Capdevila, J. Cerquides, J. Nin, and J. Torres, "Tweet-SCAN: An event discovery technique for geo-located tweets," *Pattern Recognition Letters*, vol. 93, pp. 58–68, 2017.

[12] F. Atefeh and W. Khreich, "A survey of techniques for event detection in Twitter," *Computational Intelligence*, vol. 31, no. 1, pp. 133–164, 2015.

[13] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: Real-time Event Detection by Social Sensors," *Proceedings of the 19th international conference on World wide web - WWW '10*, p. 851, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1772690.1772777

[14] A. Ritter, Mausam, O. Etzioni, and S. Clark, "Open domain event extraction from twitter," *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, p. 1104, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2339530.2339704

[15] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou, "LeadLine: Interactive visual analysis of text data through event identification and exploration," *IEEE Conference on Visual Analytics Science and Technology 2012, VAST 2012 - Proceedings*, pp. 93–102, 2012.

[16] T. Horanont and R. Shibasaki, "An Implementation of Mobile Sensing for Large-Scale Urban Monitoring," *2008 UrbanSense08 - Nov. 4, 2008, Raleigh, NC, USA*, pp. 51–55, 2008.

[17] C. Zhang, G. Zhou, Q. Yuan, H. Zhuang, Y. Zheng, L. Kaplan, S. Wang, and J. Han, "GeoBurst: Real-Time Local Event Detection in Geo-Tagged Tweet Streams,," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16*, 2016, pp. 513–522. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2911451.2911519

[18] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl, "Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition," *IEEE Conference on Visual Analytics Science and Technology 2012, VAST 2012 - Proceedings*, no. July, pp. 143–152, 2012.

[19] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing," Tech. Rep., 2016. [Online]. Available: http://arxiv.org/abs/1611.01942

[20] X. Zhou, A. Noulas, C. Mascolo, and Z. Zhao, "Discovering Latent Patterns of Urban Cultural Interactions in WeChat for Modern City Planning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18*, vol. 18, 2018, pp. 1069–1078. [Online]. Available: https://doi.org/10.1145/3219819.3219929http://dl.acm.org/citation.cfm?doid=3219819.3219929

[21] J. Massana, C. Pous, L. Burgas, J. Melendez, and J. Colomer, "Identifying services for short-term load forecasting using data driven models in a Smart City platform," *Sustainable Cities and Society*, vol. 28, pp. 108–117, 2017.

[22] I. Kok, M. U. Simsek, and S. Ozdemir, "A deep learning model for air quality prediction in smart cities," *2017 IEEE International Conference on Big Data (Big Data)*, pp. 1983–1990, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8258144/

[23] P. Ta-Shma, A. Akbar, G. Gerson-Golan, G. Hadash, F. Carrez, and K. Moessner, "An Ingestion and Analytics Architecture for IoT Applied to Smart City Use Cases," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 765–774, 2018.

[24] M. J. Perestrello De Vasconcelos, S. Sllva, M. Tome, M. Alvim, and J. M. Pereira, "Spatial Prediction of Fire Ignition Probabilities : Comparing Logistic Regression and Neural Networks," Tech. Rep. 1, 2001. [Online]. Available: https://www.researchgate.net/publication/235004861

[25] A. Kumar, S. Ali, A. Rizvi, B. Brooks, R. A. Vanderveld, K. H. Wilson, C. Kenney, S. Edelstein, A. Finch, A. Maxwell, J. Zuckerbraun, and R. Ghani, "Using Machine Learning to Assess the Risk of and Prevent Water Main Breaks," *CEUR Workshop Proceedings*, vol. 2065, pp. 472–480, 2018. [Online]. Available: https://doi.org/10.1145/3219819.3219835https://arxiv.org/pdf/1805.03597.pdf

[26] M. D. Adams and P. S. Kanaroglou, "Mapping real-time air pollution health risk for environmental management: Combining mobile and stationary air pollution monitoring with neural network models," *Journal of Environmental Management*, vol. 168, pp. 133–141, 3 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030147971530428X

[27] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia, "Multi-Task CNN Model for Attribute Prediction," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1949–1959, 2015. [Online]. Available: https://arxiv.org/pdf/1601.00400.pdf

[28] G. Antipov, S. A. Berrani, and J. L. Dugelay, "Minimalistic CNN-based ensemble model for gender prediction from face images," *Pattern Recognition Letters*, vol. 70, pp. 59–65, 2016. [Online]. Available: http://face.

[29] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013. [Online]. Available: https://pdfs.semanticscholar.org/52df/a20f6fdfcda8c11034e3d819f4bd47e6207d.pdf

[30] R. Garg, G. Carneiro, V. K. Bg, and I. Reid, "Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue," in *ECCV*, no. April, 2016, pp. 1–16. [Online]. Available: https://github.

[31] H.-W. Kang and H.-B. Kang, "Prediction of crime occurrence from multi-modal data using deep learning," *PLOS ONE*, vol. 12, no. 4, p. e0176244, 4 2017. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0176244

[32] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '16*, 2016, pp. 1–4. [Online]. Available: http://dx.doi.org/10.1145/2996913.2997016http://dl.acm.org/citation.cfm?doid=2996913.2997016

[33] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artificial Intelligence*, vol. 259, pp. 147–166, 6 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370218300973

[34] W. Jin and Y. Lin, "Spatio-Temporal Recurrent Convolutional Networks for Citywide Short-term Crowd Flows Prediction," pp. 28–35, 2018. [Online]. Available: https://doi.org/10.1145/3193077.3193082

[35] S. Takano, M. Hori, T. Goto, S. Uchida, R. Kurazume, and R.-I. Taniguchi, "Deep learning-based prediction method for people flows and their anomalies," *ICPRAM 2017 - Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*, vol. 2017-Janua, 2017. [Online]. Available: http://www.scitepress.org/Papers/2017/62488/62488.pdf

[36] M. Wang, Z. Lu, H. Li, W. Jiang, and Q. Liu, "genCNN: A Convolutional Architecture for Word Sequence Prediction," 2015. [Online]. Available: https://arxiv.org/pdf/1503.05034.pdfhttp://arxiv.org/abs/1503.05034

[37] C. J. Maddison, A. Huang, I. Sutskever, D. Silver, G. Deepmind, and G. Brain, "Move Evaluation in Go Using Deep Convolutional Neural Networks," pp. 1–8, 2015. [Online]. Available: https://arxiv.org/pdf/1412.6564.pdf

[38] J. Oh, X. Guo, H. Lee, R. Lewis, and S. Singh, "Action-Conditional Video Prediction using Deep Networks in Atari Games," *Advances in neural information processing systems*, 2015. [Online]. Available: http://papers.nips.cc/paper/5859-action-conditional-video-prediction-using-deep-networks-in-atari-games. pdfhttp://arxiv.org/abs/1507.08750

[39] Z. Lin, J. Lanchantin, and Y. Qi, "MUST-CNN: A Multilayer Shift-and-Stitch Deep Convolutional Architecture for Sequence-based Protein Structure Prediction," 2016. [Online]. Available: www.aaai.orghttp://arxiv.org/abs/1605.03004

[40] Z. Xu, Y. Yang, and A. G. Hauptmann, "A Discriminative CNN Video Representation for Event Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Xu_A_Discriminative_CNN_2015_CVPR_paper.pdf

[41] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin, "Online Real-Time Multiple Spatiotemporal Action Localisation and Prediction," Tech. Rep., 2017. [Online]. Available: https://github.com/

[42] T. H. Do, D. M. Nguyen, E. Tsiligianni, B. Cornelis, and N. Deligiannis, "Multiview Deep Learning for Predicting Twitter Users' Location," Tech. Rep., 2017. [Online]. Available: http://arxiv.org/abs/1712.08091

[43] N. Farajidavar, S. Kolozali, and P. Barnaghi, "A Deep Multi-View Learning Framework for City Event Extraction from Twitter Data Streams," 2017. [Online]. Available: http://facebook.com/http://arxiv.org/abs/1705.09975

[44] L. Laestadius, "Instagram," in *The SAGE Handbook of Social Media Research Methods*, 2017, pp. 573–592.

[45] A. Paszke, G. Chanan, Z. Lin, S. Gross, E. Yang, L. Antiga, and Z. Devito, "Automatic differentiation in PyTorch," *Advances in Neural Information Processing Systems 30*, 2017.

[46] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv*, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[47] I. Kok, M. U. Simsek, and S. Ozdemir, "A deep learning model for air quality prediction in smart cities," *2017 IEEE International Conference on Big Data (Big Data)*, pp. 1983–1990, 2017.