Hetero-ConvLSTM for Crime Prediction

Ruben Ahrens*
Lucas de Wolff*
r.t.ahrens@umail.leidenuniv.nl
l.c.wolff@umail.leidenuniv.nl
Universiteit Leiden
Leiden, The Netherlands

Figure 1. Summer protests 2020. Photo by Madison Muller. [7]

Abstract

This study proposes a Heterogenous Convolutional Long-Short Term Memory (Hetero-ConvLSTM) neural network model for predicting crimes in Chicago, IL. A ConvLSTM is able to capture spatial-temporal relations while an ensemble of ConvLSTMs can also capture heterogeneity in the data. We segmented the city into grid cells and predicted the number of crimes in each cell. Through a sliding window technique, we trained an ensemble of ConvLSTMs on subsets of the data. We also experimented with various combinations of extra features. These models were compared to a historical average baseline. We found that Hetero-ConvLSTM outperforms baseline methods.

*Both authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CCS Concepts: • **Computing methodologies** → *Supervised learning by regression*; **Neural networks**; **Image representations**.

Keywords: Recurrent Neural Networks, LSTM, ConvLSTM, Crime prediction, Heterogeneous data, Spatial-temporal data

ACM Reference Format:
Ruben Ahrens and Lucas de Wolff. 2023. Hetero-ConvLSTM for Crime Prediction. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In the city of Chicago, Illinois area, (violent) crime is a huge issue, its crime rate is higher than the United States of America average [5]. The police force works hard to fight against criminal activity. This study aims to aid in the prevention of (violent) crime by providing a machine learning model that is able to predict the number of crimes committed in certain areas. These areas are created by dividing the city into a grid of cells. Through these predictions, more efficient police patrolling routes could be generated. The new routes could allocate more resources to areas predicted to have a higher number of crimes.

Naturally, data is expected to contain some degree of heterogeneity. For example, in the case of crime prediction in

Chicago, we expect that the relevance of certain crime indicators may be different between urban and suburban regions. By using a Heterogenous Convolutional Long-Short Term Memory (Hetero-ConvLSTM) network [13], that is able to capture both spatial and temporal relations, we will expect to provide an accurate estimate of the future number of crimes. This method aims to handle the heterogeneity by using a sliding window technique to select subsets of data and training a separate ConvLSTM for each. Finally, we assemble the outputs of all models to create a final prediction.

There have been numerous other studies [8][2] that developed machine learning models capable of predicting crimes. Crime prediction can help police forces anticipate criminal activity and achieve efficient police patrolling routes. However, to our best knowledge, there are no models that tackle the heterogeneous data in a similar (and effective) way as Yuan et. al. did in the predicting traffic accidents context. Because of its crime rate and open data sources, previous research, on criminal activity is often times done in the city of Chicago. We will also apply the Hetero-ConvLSTM to the criminal data of Chicago.

This study set out to answer the following question: *Can the Hetero-ConvLSTM be used to predict the number of crimes committed per 500x500 meter cell in the city of Chicago, Illinois?*

2 Related Works

In this section, we will discuss previous research that is related to the problem we are addressing in this paper. We will begin by looking at studies that have first proposed the techniques we have used, along with research on similar techniques. Secondly, we will mention studies that used similar methods to solve this problem. Finally, we will move on to discuss work that has focused on related but slightly different aspects of the problem. Our goal in this section is to provide a comprehensive overview of the existing research in this field and to highlight how our work builds upon and extends this previous work.

The Hetero-ConvLSTM architecture was first introduced in the paper "Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data" by Yuan et al. (2018) [13]. The proposed architecture is specifically designed to handle heterogeneous spatiotemporal data and perform accident prediction tasks. The standard ConvLSTM architecture, which was first introduced in the paper "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting" by Shi et al. (2015), combines CNNs and LSTMs to model spatiotemporal data. This architecture has been used in various applications, such as video prediction, action recognition, and anomaly detection.

There has been plenty of study in the field of crime prediction. Kang et. al. [8] proposed a feature-level data fusion method with environmental context based on a deep neural network (DNN). Features used in this study range from statistics, demographic and meteorological data, to street view images. This inclined us to experiment with meteorological features. Rotaru et. al. [9] implemented a model that deals with the bias crime data can have for socioeconomically disadvantaged areas. LSTMs have also been used for predicting crime [6] in the city of Chicago. The interest in this city stems from multiple factors, for one Chicago has a notably high (violent) crime rate. Also, the city of Chicago collects and publishes a lot of data through the *Chicago Data Portal* [4].

In this paper, we propose a novel approach to crime prediction using the Hetero-ConvLSTM architecture, which is designed to handle both spatial and temporal data. Our approach takes into account the potential heterogeneity in the data, which, to our knowledge, hasn't been done before in crime prediction. We also hope that by introducing various new features the Hetero-ConvLSTM might perform even better. By combining these elements, our model aims to improve upon existing crime prediction methods and provide more accurate and actionable results for law enforcement agencies and community organizations. Overall, our work builds upon the existing literature on crime prediction and the use of Hetero-ConvLSTM.

3 Methods

In this section, it will be illustrated how the Hetero-ConvLSTM was implemented and altered to work on the task of crime prediction in Chicago. First, the data will be described. Second, the model will be explained. Finally, it will be made clear how the model was trained and what experiments we ran.

3.1 The data

Main data source

The main data was drawn from the Chicago open data portal. In this dataset, starting in 2001, every crime concerning the city of Chicago police department is registered. Every crime instance contains a few features, such as time, type, description, location, etc. The most important features in this research are the time and location features. Every other feature is considered extra. The features from the *Chicago Data Portal* dataset that we used are the time, location (given by the latitude and longitude), and the type of crime to determine the severity of each crime, which was manually engineered according to an online source [12]. Due to computational constraints, in this study, we will use the years 2011 to 2020 as the training set and the year 2021 as the test set to evaluate the model predictions. Which is equal to a 90:10 train test split ratio, respectively.

Other data sources

Because we hypothesised the weather to be a good predictor of the number of crimes, we added weather as a variable to train the LSTM on. The weather data was collected by a single weather station in the city of Chicago, which is why this feature is spatial invariant. However, we think the weather is potentially temporally correlated with the number of crimes. The data was collected through the weather database *Visual Crossing* [3]. As we had access to just one weather station in the city, we treated the weather data as a global value w.r.t. space. In other words, for each day that exists in the original crime data we stored the same weather value in all grid cells. There were no missing days, so techniques like interpolation were not needed. We have used two features from the weather data, namely: Visibility (V) and Cloud Cover (CC).

We hypothesized that visibility **(V)** has a large influence on the crime rate. As less visibility would mean a suspect would be less visible to potential witnesses or police. Visibility is one of the features extracted from *Visual Crossing*. There is a value for every day and we will populate the grid with these values and add a channel to the data.

Similarly, we expected Cloud cover **(CC)** to act as a good extra feature. Cloud cover is the visibility in the vertical direction. A day with a lot of cloud cover would mean that it is darker outside, both during the day and at night.

Finally, we added a seasonality feature **(Sea)**, which is simply the day of the year. The 1st of January would be 1 and the 31st of December would be 365, while also taking leap years into account.

3.2 Data preparation

Because we are predicting crimes per day, our first data processing involved converting the time and date column to a date column. It might have been possible to predict crimes on minute accuracy, but for computational reasons, we will aggregate locations into grid cells. As it is not feasible to predict crime in a specific location, the city was divided into a grid of 55x50 cells. This ratio created the most square-like grid cells. Each cell is roughly 500x500 meters. Using the location (latitude & longitude) of each crime we were able to map each crime to its corresponding cell. This is illustrated in figure 2, but for visual purposes, we do not show the grid cells located outside Chicago. We could then use these cells to store cell-specific data and feed it to the ConvLSTM. Not all cells were contained inside the bounds of Chicago. Using a shapefile of the city of Chicago supplied by *Chicago Data Portal* we determined what cells were located inside Chicago.

As not every grid cell will belong to the city of Chicago and we do not want to compute the model loss over grid cells outside Chicago, we have created a mask. If a cell exists in the city, the value of this cell in the mask matrix is 1, otherwise, it is 0. This mask will be used in our model to create a custom loss function that ignores predictions for crimes outside of the city.

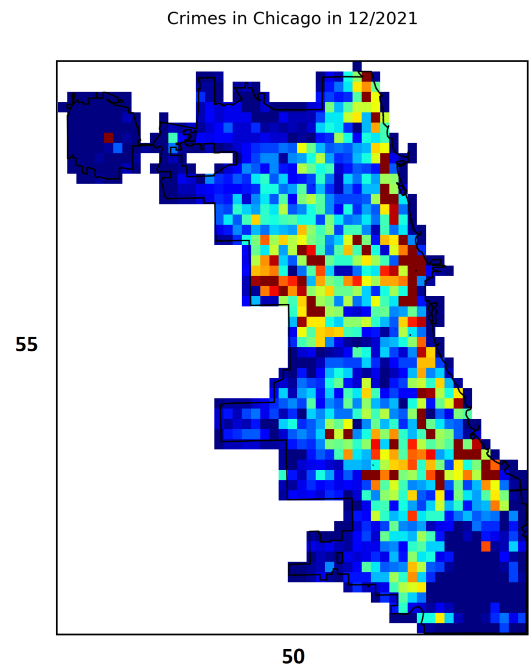

Figure 2. A map of crimes in Chicago after being aggregated into grid cells. This image is for illustration purposes.

Finally, we applied feature engineering to create an extra 'severity' **(Sev)** feature for our dataset. To compute the severity of a crime we used an online source to first classify the types of crimes into three categories in order of severity: soft, middle and worst. Table 2 shows how we classified the types of crimes. We assigned the least severe crimes a score of 0, the worst crimes a score of 1 and the rest a score of 0.5. We then computed the average crime score (or severity) for each grid cell per day.

After preprocessing, the final shape of the data was (*days*,55,50,*n_features*), with *days* being the number of days we used for our training or testing data and *n_features* being the number of features we chose to use.

3.3 The Model

Long short-term memory (LSTM) is a type of recurrent neural network. A recurrent neural network is a neural network that is able to remember activations that it had with a previous data instance when feeding data during training. For this, it uses a memory cell. It also has a forget gate. The forget gate has the effect of giving priority to recently seen data. This memory and forgetting capability make an LSTM well-suited for temporal data. It is also often used for natural language processing as the order of words is often important in language. A convolutional LSTM is an extension of the

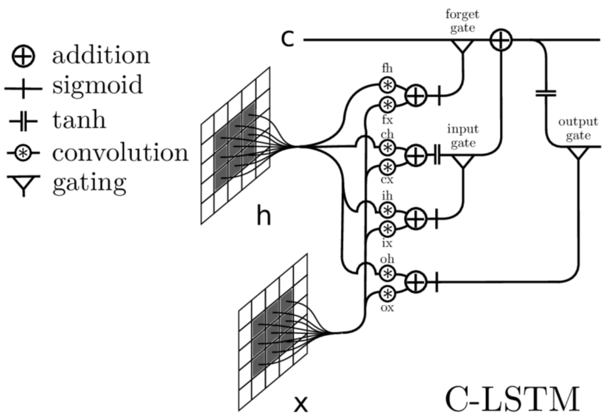

Figure 3. Convolutional LSTM structure [11]

LSTM model that enables image data to be fed into an LSTM. In this study, we essentially represent the data as images with each grid cell being an image pixel. The convolutional layers are able to able to capture the spatial relations in the data while the LSTM is capable of capturing the temporal features.

As we are dealing with heterogeneous data we will use a sliding window over the grid cells to deal with the heterogeneity in the data. For the sliding window, we divide the grid into 9 subgrids for we will use a 2 by 2 window on this subgrid. The window will move through the subgrid with a stride of 1 and without padding. For each window location, we will train an LSTM on the cells that are covered by the window. There are 4 window positions, so an ensemble of 4 LSTMs will be used to produce the final prediction. Like in the study by Yuan et al [13], we will give each LSTM the same weight. Note that it would be possible to train a linear regression model to find the best combination of ensemble weights.

For each ConvLSTM we use two stacks of two ConvLSTM layers, where every layer has 128 filters with a kernel size of 1x1, padding to retain image size and a Tanh activation function. The stack of layers is illustrated in figure 5. In between the stacks of convolutional LSTM layers, we added a batch normalization layer as was described by Yuan et. al. After concatenating the two stacks a 2d convolutional layer with a kernel size of 1x1 is used to compress the concatenated outputs into one final prediction. Because it is a regression task, we used a linear output activation function. We used an LSTM lookback size of 7 days and a stride of 1. Meaning, for every time series that the network will be trained on, the network will predict the number of crimes based on the 7 previous days. Also, the network moves with a stride of 1 through the data, meaning that we feed #days - 7 time series to the network in one epoch. The general training structure

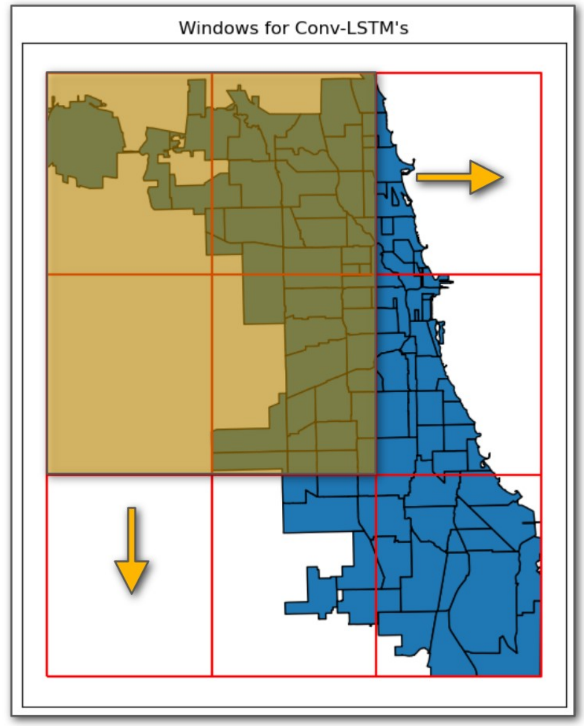

Figure 4. An illustration of the sliding window technique that was used to train the ensemble of ConvLSTMs.

of the ConvLSTM model can be seen in figure 6. In the testing stage, our model no longer trains and a 7-day sliding window is moved through the data to produce our final test predictions. We have used mean squared error (MSE) and root mean squared error (RMSE) to measure the performance of our models.

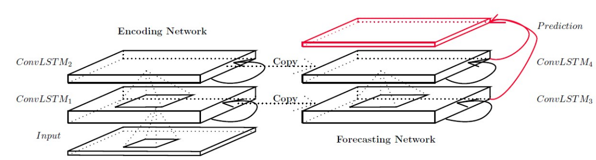

Figure 5. ConvLSTM network stack proposed by Shi et. al. [10]. In our implementation we added batch normalization layers in between the ConvLSTM layers as was described by Yuan et. al.

Training the network

All networks used a batch size of 4 to avoid OOM errors. Due to the large dataset, which is necessary for the Hetero-ConvLSTM, and the limited computational resources we have only been able to train the networks for 1 single epoch

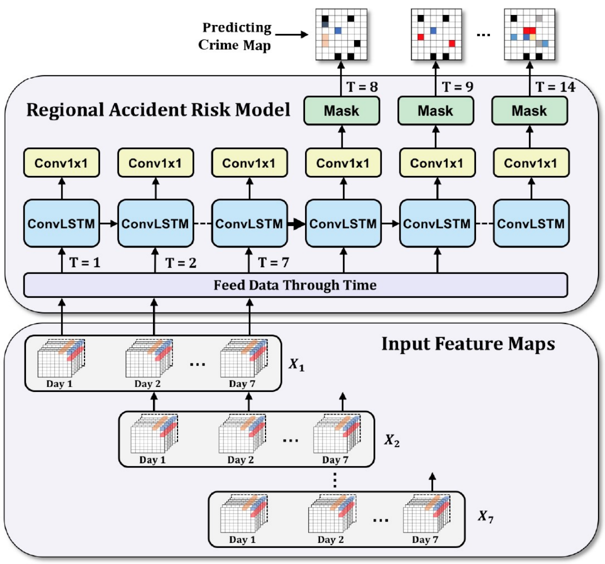

Figure 6. ConvLSTM structure [13]. An example of feeding data to the LSTM model. In this example, the LSTM is being trained on data of 14 days. In actuality, there are more predictions and time steps.

each. We used an RTX 2060 GPU using TensorFlow 2 with GPU acceleration.

Since we want the model to focus only on cells inside Chicago (especially during training), we created a custom loss function using our precomputed mask of Chicago, where we compute the masked MSE and update our model on.

Evaluation

For the final performance, we use the MSE and RMSE. Like previously, we ignore the cells that are outside of Chicago. As we use a linear activation in the output layer, predictions could become negative. To prevent this from making the predictions worse than they should be, we clipped the predictions with a minimum of 0. A negative number of crimes is impossible.

To make our results comparable, we implemented a historical average (HA) model where for each day in 2021, we predict the average of all crimes per cell from 2020. Although this approach is simple, it is usually surprisingly hard to beat. We also computed the MSE and the RMSE for the HA model.

4 Experiments

In this section, we will illustrate the experiments conducted to answer the research question: *Can the Hetero-ConvLSTM be used to predict the number of crimes committed per 500x500 meter cell in the city of Chicago, Illinois?* We will compare the Hetero-ConvLSTM performance with the HA baseline model.

Model	MSE	RMSE
HA global average (last year)	0.5399	0.7348
HA weekday average (last year)	0.5554	0.7453
ConvLSTM	0.5405	0.7352
Hetero-ConvLSTM (Cr)	**0.5369**	**0.7327**
Hetero-ConvLSTM (Cr + Sev)	0.5404	0.7351
Hetero-ConvLSTM (Cr + Sea)	0.6037	0.7770
Hetero-ConvLSTM (Cr + CC)	0.5689	0.7543
Hetero-ConvLSTM (Cr + V)	0.5448	0.7381

Table 1. In this table the result of crime prediction can be compared

We experimented with adding extra features to the data used to train the model. All features used are described in the Methods section. We experimented with adding weather data, the day of the year and a feature-engineered crime severity value. For the weather data, the focus was on visibility in the vertical and horizontal directions (visibility & cloud cover) as this was expected to be the most influential feature among the weather data. Instead of stacking multiple features, we experimented with only adding one feature at a time.

To evaluate the research question it was important to compare a Hetero-ConvLSTM with a regular ConvLSTM. Enabling the evaluation of the effect of the sliding window. We, therefore, implemented a regular ConvLSTM alongside the Hetero-ConvLSTM.

5 Results

In this section, we will present the results that were achieved during the experiments described in the previous section. Our performance results on the test set can be seen in table 1. In figure 7, the ground truth is visually compared with our best models prediction for a single day in 2021.

The best performance was achieved by the Hetero-ConvLSTM, using only our Crimes as a feature. In the conclusion section, it will be illustrated in more detail as to why this might be the case. We found that the Hetero-ConvLSTM slightly outperformed the regular ConvLSTM. In all cases, the Hetero-ConvLSTMs performed slightly worse with more features.

In figure 7, we see that the ground truth has more contrast in comparison to the best model's prediction. The model prediction produces a more spread-out prediction. It is also noticeable that the values are not integers, meaning the model could, for example, predict a crime rate of 1.4 for a single cell.

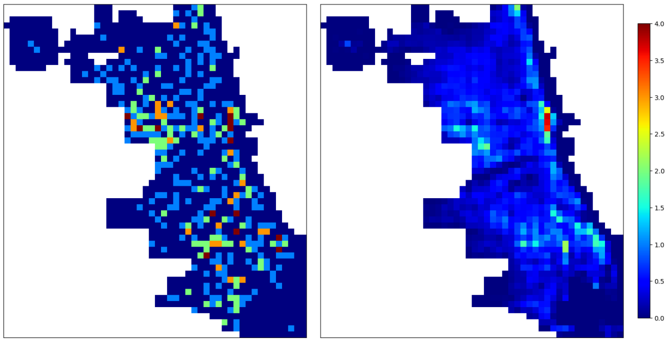

Figure 7. A comparison between the ground truth (left) and our best model (right).

6 Conclusion

The main argument of this study is that a machine learning model based on a Heterogeneous Convolutional Long-short term memory (Hetero-ConvLSTM) network can aid in the prevention of crime in Chicago by providing accurate predictions of the number of crimes committed in certain areas of the city. we argue that this approach can handle the heterogeneity of crime data by using a sliding window technique to select subsets of data and training a separate ConvLSTM model for each, then assembling the outputs of all the models to create a final prediction. This is an effective way of handling the heterogeneous data that no other models tackle similarly. The research question we set out to answer is *Can the Hetero-ConvLSTM be used to predict the number of crimes committed per 500x500 meter cell in the city of Chicago, Illinois?*

When training for one epoch, all Hetero-ConvLSTM models failed to marginally outperform the historical average baseline. We highly expect the model to significantly outperform the HA baseline when increasing the number of epochs. Similarly, currently the Hetero-ConvLSTMs with extra features, especially space invariant features performed worse than Hetero-ConvLSTM (Cr). Again, we expect the Hetero-ConvLSTMs with extra features to perform better when increasing the number of epochs. The reasoning behind this is that adding more features makes a model harder to train, but also allows more advanced patterns to be discovered. Therefore, the Hetero-ConvLSTM with added features suffers when the number of training epochs is low.

To conclude, the Hetero-ConvLSTM is a viable model to predict the number of crimes per 500x500 meter cell in the city of Chicago, Illinois. Although various versions of the model failed to outperform the other methods by a large margin, we expect that a larger number of training epochs will most likely result in an increased performance difference.

This study contributes to the existing knowledge in the field of Hetero-ConvLSTMs by showing that using an ensemble of ConvLSTM models has the potential to outperform a plain ConvLSTM. It might encourage other researchers to explore different ways to handle the heterogeneity in data.

Admittedly, training the Hetero-ConvLSTM takes longer than the regular ConvLSTM. Perhaps, given the same time budget, the ConvLSTM would perform better. Another suggestion for future research is training the models for more epochs. We expect with more training epochs the Hetero-ConvLSTM to marginally outperform all other models in table 1. To get a better understanding of the model performances, implementing more baselines (like linear regression) could be an improvement to future research.

Adding more features and testing combinations of said features are potential improvements for future research. It is unfortunate that there was only one weather station publicly available for historical weather data. This makes the weather a space-invariant feature and makes it less suitable for a ConvLSTM. In general, the space-invariant features do not improve model performance after training for one epoch, we would like to do more research in this area.

One could also experiment with increasing the number of windows or changing the shape of a window in its entirety. You could, for example, train different ConvLSTMs on subsets of neighbourhoods of Chicago.

In conclusion, this study has shown that the Hetero-ConvLSTM machine-learning model is a promising technique. It has a lot of potential for time series analysis for time-series forecasting and video prediction.

Acknowledgments

Research done on behalf of the *"Urban Computing"* course at *Universiteit Leiden* taught by Dr. M. Baratchi. The code of this study can be accessed through GitHub [1]

References

[1] Ruben Ahrens and Lucas de Wolff. 2023. Hetero-ConvLSTM for Crime Prediction GitHub repository. https://github.com/boomerr1/CrimeChicagoUC

[2] Xinyu Chen, Youngwoon Cho, and Suk Young Jang. 2015. Crime prediction using Twitter sentiment and weather. In *2015 Systems and Information Engineering Design Symposium*. 63–68. https://doi.org/10.1109/SIEDS.2015.7117012

[3] Visual Crossing Corporation. 2023. Weather Data Services: Visual Crossing. https://www.visualcrossing.com/weather/weather-data-services

[4] City of Chicago Government. 2023. City of chicago: Data Portal. https://data.cityofchicago.org/

[5] Ames Grawert, James Cullen, and Vienna Thompkins. 2015. *Crime in 2015: A final analysis*. Brennan Center for Justice at New York University of Law.

[6] Xinge Han, Xiaofeng Hu, Huanggang Wu, Bing Shen, and Jiansong Wu. 2020. Risk Prediction of Theft Crimes in Urban Communities: An Integrated Model of LSTM and ST-GCN. *IEEE Access* 8 (2020), 217222–217230. https://doi.org/10.1109/ACCESS.2020.3041924

[7] Kiran Misra Jim Daley. 2021. Chicago police overtime broke Department Records in 2020. https://southsideweekly.com/chicago-police-overtime-broke-department-records-in-2020/

[8] Hyeon-Woo Kang and Hang-Bong Kang. 2017. Prediction of crime occurrence from multi-modal data using deep learning. *PloS one* 12, 4 (2017), e0176244. https://doi.org/10.1371/journal.pone.0176244

[9] Victor Rotaru, Yi Huang, Timmy Li, James Evans, and Ishanu Chattopadhyay. 2022. Event-level prediction of urban crime reveals a signature of enforcement bias in US cities. *Nature human behaviour* 6, 8 (2022), 1056–1068.

[10] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28 (2015).

[11] Marijn Stollenga. 2016. *Advances in Humanoid Control and Perception*. Ph. D. Dissertation.

[12] Nick Stripe. 2022. Crime Severity Score (Experimental Statistics). https://www.ons.gov.uk/peoplepopulationandcommunity/crimeandjustice/datasets/crimeseverityscoreexperimentalstatistics

[13] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. 2018. Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 984–992. https://doi.org/Yuan:Traffic

A Data sources and links

- Criminal records data of Chicago: https://data.cityofchicago.org/Public-Safety/Crimes-One-year-prior-to-present/x2n5-8w5q/data
- Meteorological data: https://www.visualcrossing.com/weather/weather-data-services
- Shapefile of Chicago with neighbourhoods: https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Neighborhoods/bbvz-uum9
- GitHub repository of this study: https://github.com/boomerr1/CrimeChicagoUC

B Crime type severity

We split the categorical feature 'Primary Type' into three severity levels. The three levels are soft, middle and worst.

Soft = 0	**Middle = 0.5**	**Worst = 1**
DECEPTIVE PRACTICE	INTIMIDATION	HOMICIDE
INTERFERENCE WITH PUBLIC OFFICER	PROSTITUTION	HUMAN TRAFFICKING
GAMBLING	OTHER OFFENSE	KIDNAPPING
LIQUOR LAW VIOLATION	CRIMINAL DAMAGE	SEX OFFENSE
NON - CRIMINAL	STALKING	CRIM SEXUAL ASSAULT
NON-CRIMINAL	RITUALISM	CRIMINAL SEXUAL ASSAULT
NON-CRIMINAL (SUBJECT SPECIFIED)	CONCEALED CARRY LICENSE VIOLATION	THEFT
PUBLIC PEACE VIOLATION	PUBLIC INDECENCY	ASSAULT
	NARCOTICS	BURGLARY
	OTHER NARCOTIC VIOLATION	ROBBERY
	CRIMINAL TRESPASS	WEAPONS VIOLATION
		MOTOR VEHICLE THEFT
		OFFENSE INVOLVING CHILDREN
		BATTERY
		OBSCENITY
		ARSON
		DOMESTIC VIOLENCE

Table 2. The categorisation was done with the help of a source published by the *Office for national statistics* in the UK [12]