Using RNNs to Predict Crime Activity

Evgeniia Tcoi, Regina Gilyan, Maite Küster, Miranda Barros Everett

September 6, 2024

1 Data Presentation and Preliminary Analysis

The city of Chicago was selected for this project because of its large size and high crime rate: in 2019, city population reached 2.7 million, making it third largest nationwide; it has a crime rate of 4,382 crimes per 100,000 people, making it a prime subject for analyzing crime trends in urban environment[1]. The dataset, which was retrieved from the Chicago Data Portal, covers crime incidents from 2001 through June 2024[2]. The initial dataset included 8.09 million reports, detailing the time and location of incidents, crime types, and corresponding codes based on the Illinois Uniform Crime Reporting (IUCR) system, a pre-2021 standard used by the Chicago Police Department.

Three geographic metrics were used to classify crime locations: police beats, districts, and community areas, the latter corresponding to U.S. Census boundaries. In addition, precise longitude and latitude coordinates of these regions were included. Given the well-documented relationship between crime and socio-economic conditions, census data was integrated to enhance analysis, as previous research indicated that its exclusion decreased crime prediction accuracy[3]. Census data was sourced from Community Data Snapshots (CDS) project, which relies on the U.S. Census Bureau's 5-year American Community Survey as its primary source[4]. The 77 community areas in Chicago were selected as the geographic basis for CDS data, because these matched the geographical areas used in the crime dataset. Key socio-economic variables —unemployment rate, total population, and the number of households within the lowest income bracket (less than 25,000 USD per year) per community area — were collected for the period 2018–2024, the range for which CDS data was available.

To account for environmental factors, daily and hourly weather data was incorporated using the Historical Weather API[5]. A total of 15 weather variables, including temperature, precipitation, and daylight duration were extracted to cover the full time period of the crime dataset.

San Diego was initially selected to test the model's performance, given its large population in 2019 (1.4 million) and lower crime rate compared to Chicago (2,221 crimes per 100,000 people)[1]. This made it ideal for evaluating the model's sensitivity to urban areas with lower crime rates. The dataset, retrieved from the SANDAG Open Data Portal [6], included crime data, dated from 2021 to present, that was encoded according to the new standard (NIBRS). Census data, analogous with the one chosen for Chicago, was retrieved from the United Census Bureau directly, using tables S1701 (household income below poverty level), B23001 (unemployment), and DP05 (population)[7]. Historical weather data was also integrated using the same weather API.

However, the San Diego dataset lacked precise geographic coordinates, rendering it unsuitable for the model. As a result, model evaluation was restricted to Chicago data exclusively.

Data preprocessing was conducted using pandas and geopandas packages. Incident dates were converted to datetime format to extract temporal features such as year, season, month, day of the week, and hour. To retain location coordinates, approximately 90,000 rows (1.1% of the dataset) with missing latitude or longitude were removed. Missing community area and district numbers were inputted using geospatial data from the Chicago Data Portal, reducing missing values from 20,748 to 494, which were subsequently dropped. Community area names from both the CDS and Chicago data were standardized before merging the datasets.

The primary challenge we have encountered with the data was the encoding of crime categories. Chicago's crime data adhered to the older UCR (Uniform Crime Reporting) standard, which was officially replaced by the FBI's National Incident-Based Reporting System (NIBRS) in 2021. NIBRS introduced new crime classifications and removed the UCR's hierarchy rule, which only recorded the most serious offense per incident, unlike NIBRS, where multiple offenses within a single incident are captured, providing more detailed information. To align the UCR-encoded data with NIBRS for comparison, multiple approaches were attempted. A fuzzy matching method yielded low accuracy (41% being the best outcome), prompting a manual mapping of crime codes based on guides from NIBRS and Chicago PD's IUCR. However, due to the complexity in legal classifications, expert knowledge was required for the mapping. Additionally, given that San Diego data was deemed unsuitable for our model, the analysis proceeded using the original UCR-encoded data to avoid ambiguity, striving for more precise interpretations of the results but with their relevance reduced to newer data which follows NIBRS standard.

Preliminary analysis of Chicago crime data revealed that the crime rates have been consistently declining throughout the years, with a recent sharp drop in 2020, followed by an increase post-2021, actually surpassing pre-pandemic levels (Figure 1). It should be noted that the dip in 2024 is due to the fact that the year and the data corresponding to it is still incomplete.

The abrupt downturn trend during the pandemic is visible on the heatmap of crime activity across districts from 2018 to 2024 period (Figure 2) as well. This visualization also highlights that there are stable trends in certain districts such as Gresham, Chicago Lawn or South Chicago exhibiting higher than average crime rates, or Lincoln and House of Representatives with significantly lower crime levels.

Correlational analysis of numerical features demonstrated (Figure 3) that the outlined blocks of data (weather, location, census) have statistically significant relationships between the variables within them and are mostly independent from each other. The exception to be noted is a relationship between the census and location data, with mostly neutral and moderately negative correlation. Upon close inspection, strong positive correlations between each of the chosen census variables and the number of crimes committed were revealed (Figure 4). Household income had the highest correlation coefficient (0.81), indicating that areas with more low-income households tend to experience higher crime rates. This aligns with expectations, as economic instability often correlates with increased crime.

Given the noted shifts in crime trends during the pandemic, a decision was made to exclude the pre- and post 2020 data for a more accurate analysis of current trends, free from pandemic-related distortions. This approach better aligned the model with the non-pandemic landscape, enhancing its suitability for predictive analysis. The final Chicago dataset included over 700,000 crime reports from 2021 to 2024, offering a relevant and up-to-date foundation for further study.

2 Deep Learning Model Implementation

In this chapter, we present our proposed CNN+LSTM model. The model is designed to learn from spatial, temporal and socio-economic data. The first component is a Convolutional Neural Network (CNN), a type of deep neural network that processes images by applying filters to detect edges and shapes, enabling the recognition of patterns. In our case, these images are heatmaps, showing the crime hotspots for each district at a given week. The second component is a Long Short-Term Memory (LSTM) network, which is a specialized type of Recurrent Neural Network (RNN). An LSTM uses gating mechanisms to manage what information to retain or forget from previous memory cells. RNNs process sequences of inputs $x = (x_1, \ldots, x_n)$ and generate outputs y by applying operations on these sequences. Designed specifically for time series data, RNNs feature a feedback loop that recycles processed information as input for subsequent time steps in the sequence. The LSTM processes the heat map images through its cells, determining which aspects of the image to retain or discard in order to make accurate predictions about crime counts, while another LSTM channel handles socio-economic and weather data to capture additional influential factors.

As mentioned, the dataset included data from the years 2021, 2022, 2023 and 2024. This includes geographic data, temporal information, weather attributes, and socio-economic indicators. The first step in preprocessing involved inspecting the data for missing values. The dataset is found to be complete, ensuring the integrity of the data moving forward. We proceeded by cleaning and renaming various columns for ease of use. Given the extremely low frequency of district 31 (only 48 occurrences compared to tens of thousands for all other districts), it was excluded from the model, knowing it is unlikely to learn meaningful patterns to build a prediction. Following this, the data types of specific columns are optimized, with categorical variables such as `season` and `primary_type` being converted into the category type. This allows for efficient memory usage and easier manipulation during training. Additionally, a date column is created by extracting the date portion from `time_stamp` , and a `week_in_year` column is added to label each record by its corresponding week number. The data was aggregated by week and district, creating a summarized dataset that serves as the model's input. The aggregation includes total weekly crime counts, weather data (e.g., temperature, precipitation), and socio-economic indicators like population and unemployment rates. Here the median was chosen as the suitable option for both weather and socio-economic data aggregation. The organized structure of the data ensures that it is ready for modeling, with each row representing all districts during a specific week.

A key part of the data preprocessing involved creating sequences of input data that the CNN-LSTM model can learn from. Three types of sequences are generated:

- `X_images_sequences`, which holds heatmaps of crime activity for each district over a specified number of weeks.
- `X_socio_sequences`, containing socio-economic data like population, unemployment and weather features for each district over the same period.
- `y_labels_sequences`, which holds the crime counts for each district for the week following the time window defined by the previous data.

For example, with a window size of 10 weeks, each element in `X_images_sequences` holds heatmap data for all 22 districts over the past 10 weeks. Similarly, `X_socio_sequences` stores socio-economic data for the same period. Finally, `y_labels_sequences` contains the crime count for each district for the week following the 10-week window. The relevance of the `window` variable lies in the sliding window method we have incorporated. This involves using a fixed-size "window" of past data to make predictions about future data. In the case of our model it means we are using data from the past 10 weeks as input to predict the crime count for the next week. Overall this structure allows the model to learn patterns in both the image (spatial) and numerical (temporal) data, making it suitable for forecasting district-level crime.

The final data preparation starts with data normalization. The socio-economic data stored in `X_socio_sequences` is normalized using the `StandardScaler`, which transforms the data to have a mean of 0 and a standard deviation of 1. This ensures that the socio-economic features are on a comparable scale, which, among other things, can improve the convergence of the model during training. The crime counts in `y_labels_sequences` are also normalized, but the normalization is applied separately for each district. The `StandardScaler` is used to flatten `y_labels_sequences` into a 2D array, perform normalization, and then reshape the data back into its original 3D shape (samples $\times$ window size $\times$ districts). This separate scaling ensures that variations in crime counts between districts do not affect the model's ability to learn patterns for each district individually. The heatmap sequences (`X_images_sequences`) are reshaped into 5D arrays to match the input shape expected by the model, which takes time-distributed sequences of 2D image data. Specifically, the data is reshaped to have dimensions of (samples, window size, 256, 256, 1)—representing the number of samples, time steps (weeks), image height, image width, and a single channel for grayscale images. The dataset is then split into training and test sets using the `train_test_split` function. A 90-10 split is applied, meaning that 90% of the data is used for training, and 10% is held out for testing.

As mentioned, the model combines a CNN for processing heatmaps (spatial data) and an LSTM for handling both image features and socio-economic data (temporal data). The image input (`input_images`) has a shape of (`window_size`, 256, 256, 1), where `window_size` represents the 10 weeks of data, and each image is 256x256 pixels in size. A series of time-distributed convolutional layers (`TimeDistributed(Conv2D)`) is applied to extract spatial features from the heatmaps. These layers include:

- Three convolutional layers with `relu` activations and L2 regularization, followed by batch normalization to stabilize training.
- Max pooling layers reduce the spatial dimensions after each convolution, downsampling the feature maps.

- Finally, a `Flatten` layer transforms the 2D feature maps into 1D vectors for the LSTM to process.

The output of the CNN is fed into an LSTM layer, which captures temporal dependencies in the extracted image features. Two LSTM layers are applied, followed by a dropout layer to prevent overfitting. Similarly, the socio-economic input (`input_socio`) is passed through its own LSTM layers. The LSTM layers for both the image features and socio-economic data are configured with 50 units each and use `tanh` activation functions. Dropout layers are also applied to reduce overfitting. The outputs from the LSTM layers for both the image and socio-economic data are concatenated to form a combined feature vector. This combined feature vector is passed through a fully connected (`dense`) layer with 50 neurons and L2 regularization, followed by a dropout layer. The final output layer uses a `relu` activation function and is designed to predict crime counts for each district. The number of output neurons corresponds to the number of districts (`num_districts`), ensuring that the model provides separate crime predictions for each district. For more details, please see Appendix A.

The model was trained for 36 epochs using a batch size of 32 and the Adam optimizer with an initial learning rate of 0.0001. The loss function used was mean squared error (MSE), targeting the prediction of crime counts across 22 districts. During training, the loss steadily decreased from 6.38 in the first epoch to 0.88 in the final epoch, indicating a consistent improvement in fitting the training data. The validation loss showed a similar trend, reducing from 4.61 to 0.74 by the end of training. These results suggest the model's generalization capabilities were improving, though diminishing improvements were observed beyond the 30th epoch, where the validation loss plateaued. On the test set, the final MSE was 0.68, with an R^2 score of 0.20, implying that the model explains 20% of the variance in crime counts across districts. This relatively low R^2 score indicates that while the model captures some patterns in the data, a significant portion of the variance remains unexplained.

In evaluating the performance of our model, we compare the predicted and true values for selected districts. For instance, the true value for District 1 is 0.5155 while the predicted value is 0.6361. Similarly, for District 5, the true value is 1.0630 compared to a predicted value of 0.5652. For District 10, the true value is 1.2897 while the predicted value is 0.6861. These examples illustrate that there is a noticeable discrepancy between the true and predicted values, particularly for districts with higher true values. Further analysis of the variance of the true and predicted crime counts reveals additional insights. The variance of the true crime counts across all districts was 0.85, while the variance in predicted counts was significantly lower at 0.10. This suggests that the model tends to predict values within a narrower range, possibly underestimating the variability in crime rates across districts. The mean of the predicted crime counts was slightly higher than the mean of the true counts, with values of 0.2633 and 0.0209, respectively. The lower variance and slight overestimation could be indicative of a tendency for the model to regress toward the mean, particularly in districts with lower crime rates.

In summary, the model demonstrates moderate predictive capability, as seen by the steady reduction in training and validation loss, but further refinement is necessary to improve its ability to capture the full range of crime count variability across districts. Techniques such as regularization adjustments, hyperparameter tuning, or incorporating additional features could potentially enhance the model's performance.

3 Research Analysis

Machine learning and deep learning algorithms, such as the one we have implemented, have become important tools in modern law enforcement. In recent years, an increasing number of police departments have adopted predictive policing, an approach that relies on historical crime data to help guide law enforcement's decision-making and improve efficiency [8]. Location-based predictive policing is grounded in crime pattern theories, which are based on the observation that crimes are not evenly distributed across all areas, rather they tend to be concentrated in specific hotspots. Predictive policing is therefore based on the idea that by analyzing data that specifies locations of past crimes, law enforcement can identify geographic areas with an increased probability of criminal activities and make more informed decisions based on that information [9]. As such, forecasting crime hotspots enables law enforcement to deliver resources to areas that have been predicted to be high-risk for crime. In contrast to traditional, reactive policing, officers can engage in proactive patrols and surveillance in order to reduce the opportunity for criminal behavior and thereby increase public safety [10].

However, along with these seemingly obvious benefits of predictive policing, there are some significant drawbacks and ethical concerns. Besides privacy concerns and legal implications, such as to what extent excessive surveillance can be justified, the potential for racial bias in predictive policing algorithms has been a topic of debate among scholars. Proponents argue that policing supported by advanced statistical methods enhances law enforcement's accountability and negates discriminatory practices by letting decisions be guided by "objective" and scientific tools [10]. On the other hand, opponents claim that crime prediction models may perpetuate or even exacerbate racial bias because predictions are based on a skewed depiction of society. Research from the past decades has shown that policing – especially in the US – has historically been a racially biased practice, with minorities overrepresenting every part of the criminal justice system relative to their representation in the US population [10]. Because predictive policing algorithms rely on existing crime data, critics argue that they are inherently biased against minorities, as the data itself is biased. There are mainly two kinds of data that are used to train predictive models: data that contains arrests and data that contains crimes reported by the public. Bias is present in both cases. It seems straightforward that arrest data must be biased if policing is characterized by discriminatory actions. In the case of reported crimes, bias is present because minorities often over-report crimes compared to their white counterparts [11]. Concerning location-based predictive policing, the real-life risks highlighted by scholars are twofold: first, patrols are encouraged by the algorithm to unduly target minority communities; and second, officers' awareness of being in a high-risk area for crime may amplify biases [10, 12]. Overall, the concern is that the promise of more objective decision-making does not hold, instead, predictive policing amplifies patterns of social inequality.

Empirical evidence on the effectiveness of predictive policing and its repercussions for minorities remains scarce and inconclusive. Evaluating the effectiveness of predictive policing is usually based on multiple criteria, such as the number of correct crime predictions, crime rates before and after implementation, and the cost-effectiveness of utilizing an algorithm as compared to crime analysts [13]. In a 24-week study, Levine et al. (2017) evaluated predictive policing by the NYPD based on a crime-monitoring system and compared it to traditional hotspot policing methods. The results showed that predictive policing led to more accurate crime forecasts and increased efficiency of law enforcement through faster response times, while overall crime decreased by 6% [14]. Similarly, a randomized controlled trial conducted by Mohler et al. (2015) in Los Angeles and Kent compared a predictive policing model with traditional crime analyst methods and found that patrols based on model predictions led to an overall 7.4% reduction in crime [15]. In contrast, other publications report on inconclusive results or draw attention to the possibility of increasing crime rates in areas around police interventions. Ratcliffe et al. (2020) found only non-significant reductions in property crime, while highlighting the limited nature of this effect due to low baseline crime volumes which make changes expressed in percentages appear exaggerated. Similarly, low incident volume and small sample sizes lead to insignificant results concerning violent crimes [16]. Ariel and Partridge (2017) observed a 37% reduction in crime around bus stops through predictive patrols, however, the authors emphasize a significant displacement effect, as they report a 25% increase in victim-reported crimes in the immediate area around the targeted stops [17]. Overall, as Hardyns and Rummens (2017) conclude, evidence on the efficiency of predictive policing is limited and remains preliminary, as research is still insufficient to gain comprehensive insights [13]. This also holds true for the effect on racial disparities when measuring arrest rates of minority groups. An analysis conducted by Brantingham et al. (2018) investigated how arrest rates in three LAPD divisions impact race and ethnicity through the use of predictive policing. The authors report no significant differences in arrest rates for minorities, nor a difference in overall arrest rates between treatment and control conditions when adjusted for area crime rates [12]. However, it is conceivable that observed differences in arrest rates might be delayed as a result of the so-called feedback loop. The concept of the feedback loop is simple: predictions lead to increased police presence in certain areas, which leads to more crimes being recorded in those areas. This information then feeds back into the model, resulting in an even higher predicted crime rate in those areas, and the cycle begins again with increased policing and crime detection [18, 12]. Therefore, significant rises in arrest rates might take time to become evident in the data. Instead of examining arrest rates, a part of academic research has focused on investigating bias in the data itself as well as in predictive algorithms. Lum and Isaac (2016) aimed to investigate how biased police datasets are by comparing them to a combined dataset of a demographically synthetic population of Oakland with data from the 2011 National Survey on Drug Use and Health (NSDUH). This approach was used to try and capture a complete record of all crimes that occur, including crimes that remain unrecorded by law enforcement. The authors argue that relying on the NSDUH data can be considered a more accurate measure of drug use compared to police data because it provides a representative sample free from biased law enforcement practices [18]. The comparison between drug arrests based on data from the Oakland Police Department and the estimated drug use in Oakland reveals a considerable discrepancy. In 2010, police data showed that drug arrests were concentrated in two predominantly non-white, low-income neighborhoods, where arrests were 200 times higher than in other areas of the city. At the same time, the estimated drug crimes show a much more even distribution across the city with variations in occurrence proportionate to population density. This indicates that even though drug use occurs uniformly, crimes in non-white and low-income communities are highly overrepresented in police data. A report on racial disparities in drug-related arrests, published by the American Civil Liberties Union (ACLU) in 2013 [19] supports this claim. The report asserts that while Black and White Americans use drugs at similar rates, Blacks are nearly four times more likely to be arrested for drug-related crimes. Lum and Isaac's research goes a step even further than this. They fed Oakland's police data into Geolitica (formerly known as PredPol), a software used by the LAPD for crime forecast, and found that the algorithm disproportionately targeted minority communities, thereby reinforcing racial bias in the data [18]. Similar conclusions were drawn in other reviews on the topic [11, 8, 10].

To conclude, it can be said that while predictive policing has the potential to offer benefits for law enforcement efficiency and crime prevention, there is an urgent need for strict regulations that consider legal, as well as ethical implications. To achieve this, more research is needed on the efficiency and fairness of predictive algorithms to understand the full scope of their influence. Additionally, any potential advantages of data-driven decision-making must be carefully weighed against the risk of reinforcing social inequalities and their effects on society at large.

References

[1] Offenses Known to Law Enforcement by State by City, 2019. Crime in the U. S. https://ucr.fbi.gov/crime-in-the-u.s/2019/crime-in-the-u.s.-2019/tables/table-8/table-8.xls/view

[2] Chicago Data Portal. https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2

[3] Stec, A., and Klabjan, D. (2018). Forecasting Crime with Deep Learning. https://doi.org/10.48550/arXiv.1806.01486

[4] Community Data Snapshots (2014-2024) from Chicago Metropolitan Agency for Planning. https://arcg.is/0baz1e0

[5] Historical Weather API from Open-Meteo. https://open-meteo.com/en/docs/historical-weather-api

[6] CIBRS Group A and B Public Crime Data from ARJIS on SANDAG Open Data Portal, Group A and Group B

[7] United States Census Bureau Data Portal, https://data.census.gov/

[8] Meijer, A., & Wessels, M. (2019). Predictive policing: Review of benefits and drawbacks. *International Journal of Public Administration*, *42*(12), 1031–1039. https://doi.org/10.1080/01900692.2019.1575664

[9] Jenga, K., Catal, C., & Kar, G. (2023). Machine learning in crime prediction. *Journal of Ambient Intelligence and Humanized Computing*, *14*(4), 2887–2913. https://doi.org/10.1007/s12652-023-04530-y

[10] Browning, M., & Arrigo, B. (2021). Stop and risk: Policing, data, and the digital age of discrimination. *American Journal of Criminal Justice*, *46*(2), 298–316. https://doi.org/10.1007/s12103-020-09557-x

[11] Gallon, A. (2023). *Racism repeats itself: AI racial bias in predictive policing algorithms*. Retrieved from https://thinkyou.bayhonors.org/wp-content/uploads/BHS_Papers/2023_Symposium/AlexiaGallon_RacismRepeatsItself.pdf

[12] Brantingham, P. J., Valasik, M., & Mohler, G. O. (2018). Does predictive policing lead to biased arrests? Results from a randomized controlled trial. *Statistics and Public Policy*, *5*(1), 1–6. https://doi.org/10.1080/2330443X.2018.1438940

[13] Hardyns, W., & Rummens, A. (2018). Predictive policing as a new tool for law enforcement? Recent developments and challenges. *European Journal of Criminology*, *24*(2), 201–218. https://doi.org/10.1007/s10610-017-9361-2

[14] Levine, E. S., Tisch, J., Tasso, A., & Joy, M. (2017). The New York City Police Department's Domain Awareness System. *Interfaces*, *47*(1), 70–84.

[15] Mohler, G. O., Short, M. B., Malinowski, S., Johnson, M., Tita, G. E., Bertozzi, A. L., & Brantingham, P. J. (2015). Randomized controlled field trials of predictive policing. *Journal of the American Statistical Association*, *110*(512), 1399–1411. https://doi.org/10.1080/01621459.2015.1077710

[16] Ratcliffe, J. H., Taylor, R. B., Askey, A. P., & others. (2021). The Philadelphia predictive policing experiment. *Journal of Experimental Criminology*, *17*(1), 15–41. https://doi.org/10.1007/s11292-019-09400-2

[17] Ariel, B., & Partridge, H. (2017). Predictable policing: Measuring the crime control benefits of hotspots policing at bus stops. *Journal of Quantitative Criminology*, *33*(4), 809–833. https://doi.org/10.1007/s10940-016-9312-y

[18] Lum, K., & Isaac, W. (2016). To predict and serve? *Significance*, *13*(5), 14–19. https://doi.org/10.1111/j.1740-9713.2016.00960.x

[19] American Civil Liberties Union (ACLU). (2013). *The war on marijuana in black and white*. ACLU. https://www.aclu.org/publications/report-war-marijuana-black-and-white

A Model architecture

```
# Model architecture
input_images = Input(shape=(window_size, 256, 256, 1))

# CNN layers with batch normalization
x = TimeDistributed(Conv2D(32, (3, 3), activation='relu', padding
    ='same', kernel_regularizer=l2(0.01)))(input_images)
x = TimeDistributed(BatchNormalization())(x)
x = TimeDistributed(MaxPooling2D(pool_size=(2, 2)))(x)
x = TimeDistributed(Conv2D(64, (3, 3), activation='relu', padding
    ='same', kernel_regularizer=l2(0.01)))(x)
x = TimeDistributed(BatchNormalization())(x)
x = TimeDistributed(MaxPooling2D(pool_size=(2, 2)))(x)
x = TimeDistributed(Conv2D(128, (3, 3), activation='relu',
    padding='same', kernel_regularizer=l2(0.01)))(x)
x = TimeDistributed(BatchNormalization())(x)
x = TimeDistributed(MaxPooling2D(pool_size=(2, 2)))(x)
x = TimeDistributed(Flatten())(x)

# LSTM layers on image features
x = LSTM(50, activation='tanh', return_sequences=True,
    kernel_regularizer=l2(0.01))(x)
x = LSTM(50, activation='tanh', kernel_regularizer=l2(0.01))(x)
x = Dropout(0.5)(x)

input_socio = Input(shape=(window_size, X_socio_train.shape[2]))

# LSTM layer on socio-economic data
y = LSTM(50, activation='tanh', return_sequences=True,
    kernel_regularizer=l2(0.01))(input_socio)
y = LSTM(50, activation='tanh', kernel_regularizer=l2(0.01))(y)
y = Dropout(0.5)(y)

combined = concatenate([x, y])

# Fully connected layers with dropout and L2 regularization
z = Dense(50, activation='relu', kernel_regularizer=l2(0.01))(
    combined)
z = Dropout(0.5)(z)

# Output layer modified to predict per district
output = Dense(num_districts, activation='softplus')(z)

# Compile the model
model = Model(inputs=[input_images, input_socio], outputs=output)
model.compile(optimizer=Adam(learning_rate=0.0001), loss='mse')

# Summarize the model
model.summary()

# Callbacks for dynamic learning rate adjustment and early
    stopping
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1,
    patience=5, min_lr=1e-6)
early_stopping = EarlyStopping(monitor='val_loss', patience=3,
    restore_best_weights=True)

# Training
history = model.fit(
    [X_images_train, X_socio_train], y_train,
```

```
    validation_data=([X_images_test, X_socio_test], y_test),
    epochs=50,
    batch_size=32,
    callbacks=[reduce_lr, early_stopping]
)
```

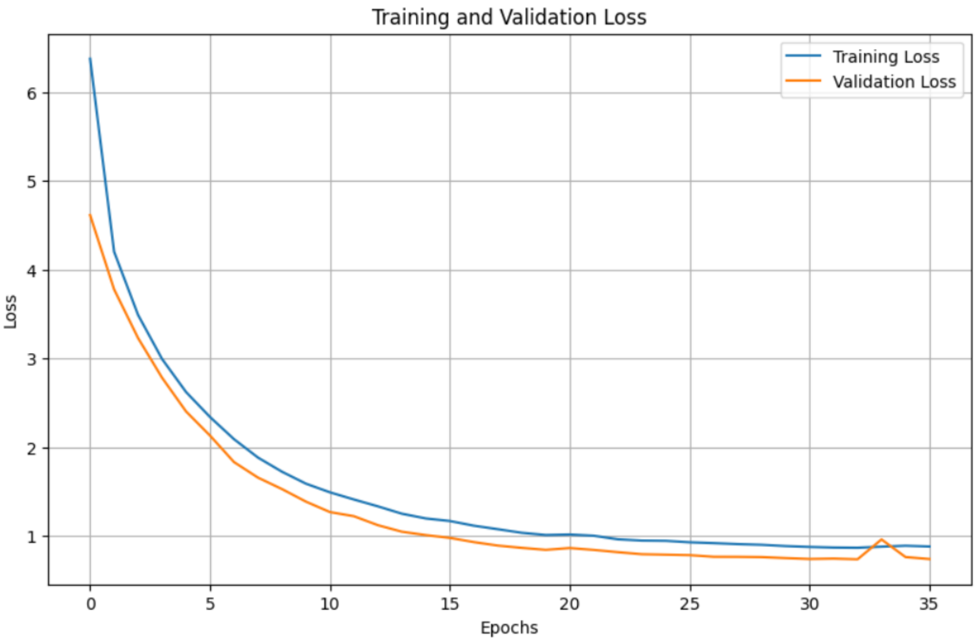

Figure 1: Your image caption here

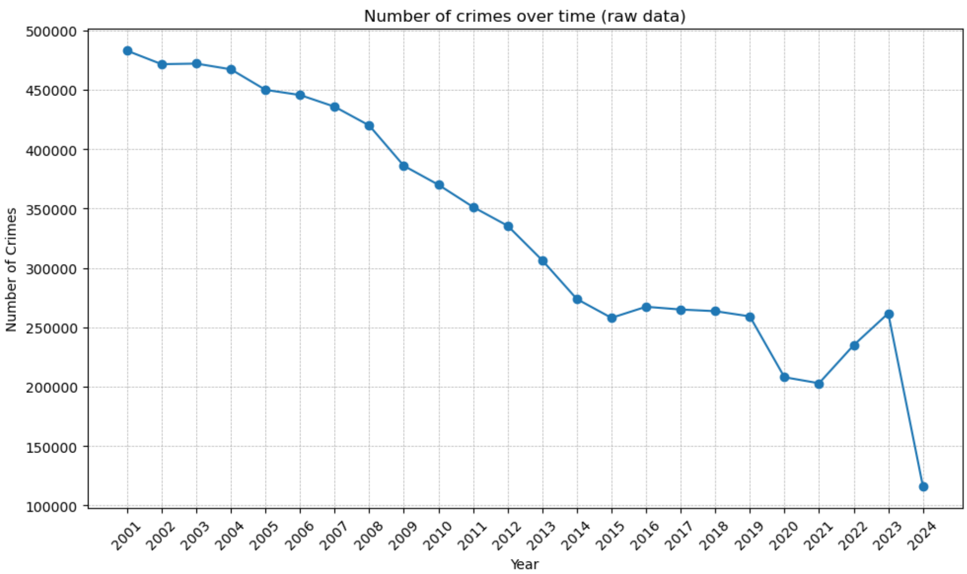

Figure 2: Crime Counts (2001-2024)

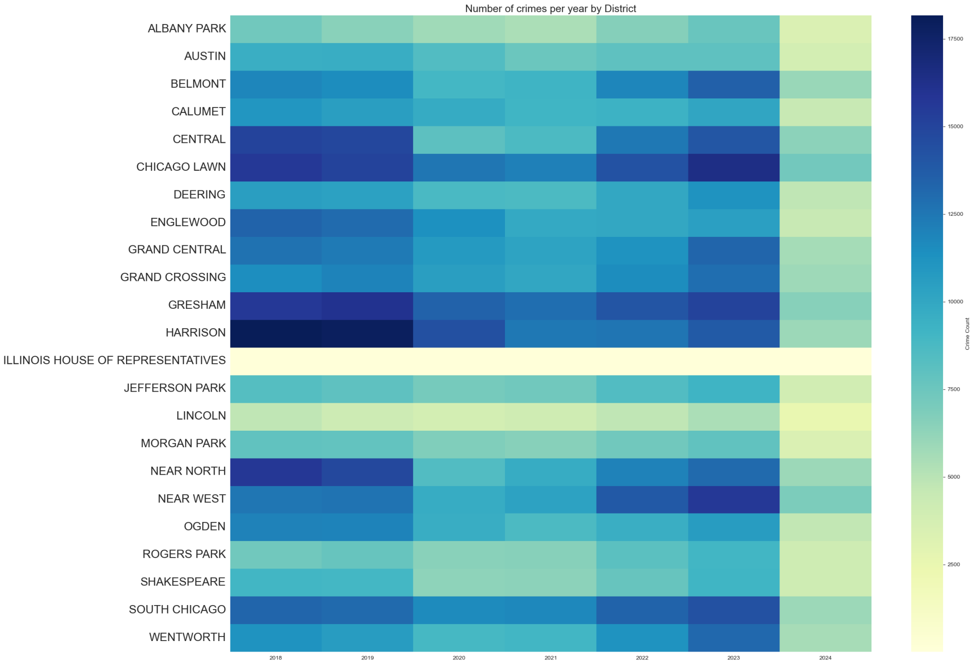

Figure 3: Crime Counts by Districts (2018-2024)

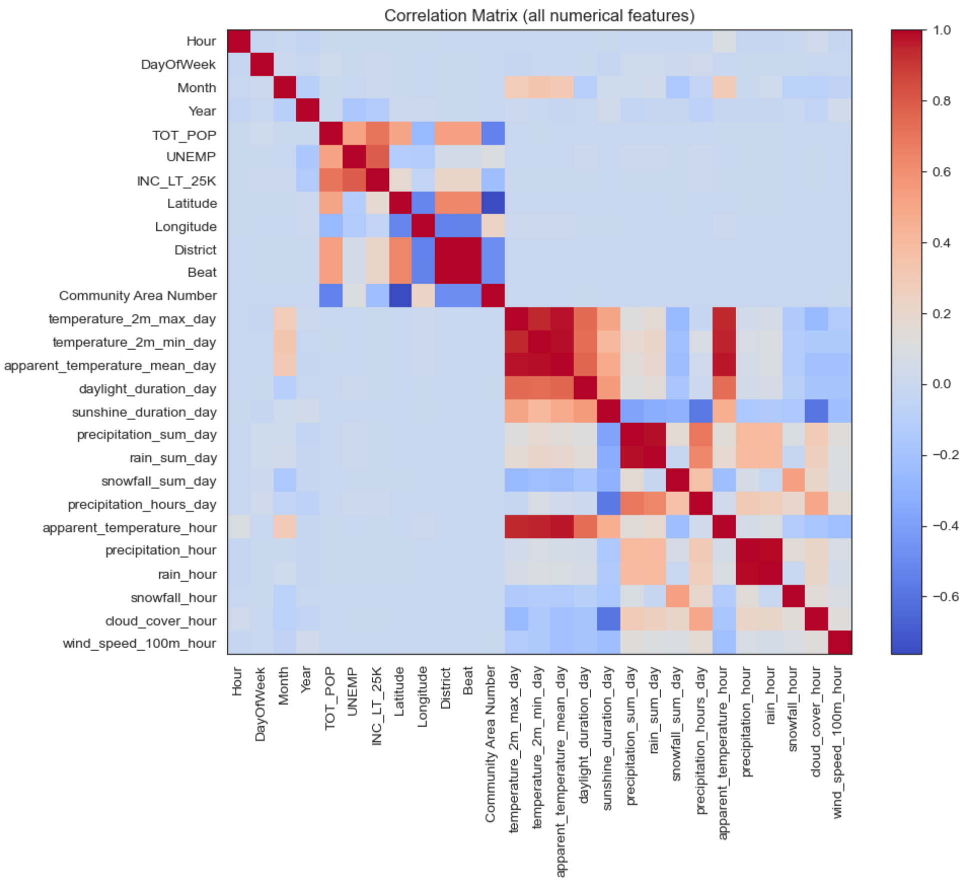

Figure 4: Correlation matrix: all numerical data

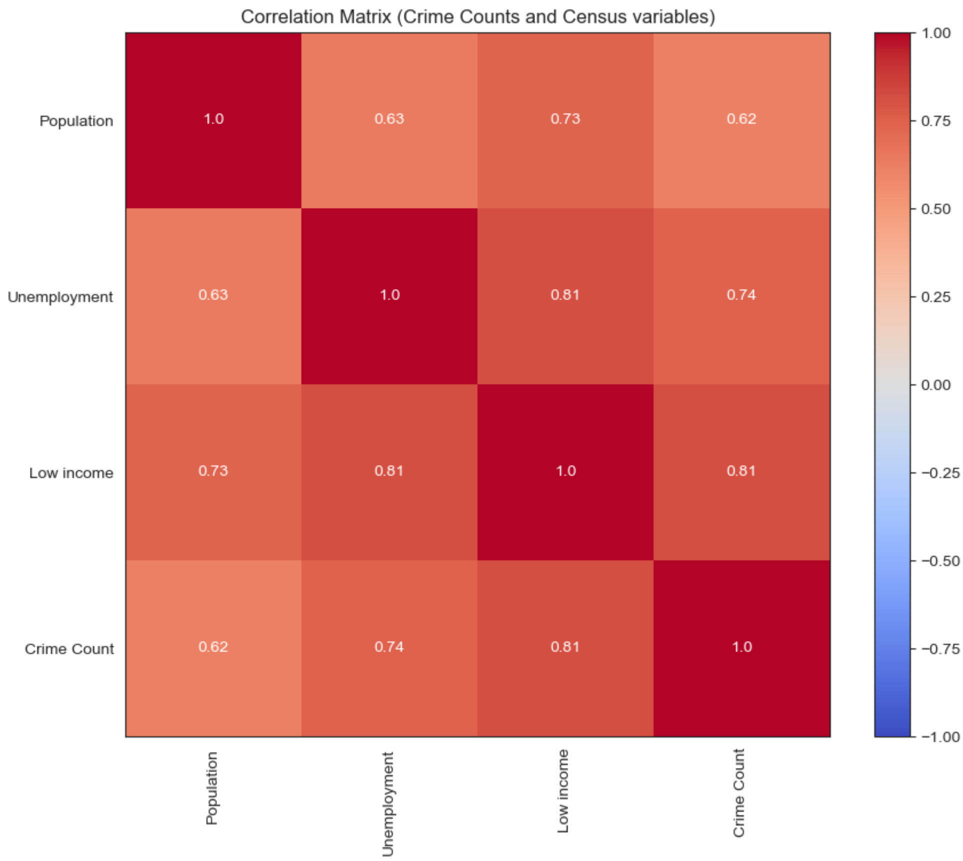

Figure 5: Correlation matrix: census variables and crime counts