Crime Analysis and Prediction Using Deep Learning Algorithms

Yinuo Tang, Xiran Li, Dheeksha Mageswaran
University of California, Davis

June 2024

1 Motivation

The motivation behind this project is to improve public safety by predicting future crime patterns. Using historical crime data and deep learning algorithms, our goal is to identify areas where or in which crime is likely to happen. This information can help law enforcement agencies allocate resources more efficiently and prevent crime before it occurs. By making accurate predictions, this project aims to support better decision-making in policing strategies and ensure safer communities through data-driven insights.

In addition to enhancing public safety, another motivation is to explore how deep learning can be used effectively in crime prediction. Crime analysis presents unique challenges due to the complex patterns and multiple factors influencing criminal behavior. Using advanced neural network models, we aim to uncover hidden relationships and non-linear patterns in the data that traditional methods might miss. With this project, we want to show how deep learning can be applied practically to real-world problems, making valuable contributions to the growing research on artificial intelligence in crime prediction.

2 Related Work

2.1 K-Means Clustering

In the paper "Crime Analysis using K-Means Clustering," the authors utilize the K-Means clustering algorithm to analyze crime data from England and Wales between 1990 and 2011-12. The process begins with data collection and preprocessing, where the dataset is cleaned, and filtered, and missing values are handled using RapidMiner's "Replace Missing Value" operator. The data is then normalized to ensure each feature contributes equally to the clustering process. The K-Means algorithm is applied by setting a predefined number of clusters (K) and iteratively assigning data points to the nearest centroids, recalculating centroids until convergence is achieved. The clustered data is visualized using RapidMiner's plotting tools, enabling the identification of distinct crime patterns and trends over different periods. This method allows for the effective analysis of crime data, revealing insights such as high-crime periods and geographical hotspots, which can aid in developing strategies for crime prevention and resource allocation.[1].

2.2 Deep Learning and Crime Analysis

Deep learning has been extensively utilized for crime prediction and classification, employing various architectures and methodologies to enhance accuracy and effectiveness. One study leverages a hybrid model combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, with the Bi-directional LSTM (BiLSTM) model achieving superior performance in weekly crime predictions compared to other methods, particularly for spatiotemporal crime data from cities like Chicago, New York, and Lahore [2]. Another comprehensive review explores the application of deep learning models, including CNNs, Recurrent Neural Networks (RNNs), and hybrid models that integrate CNNs with RNNs or Gated Recurrent Units (GRUs). This review highlights high accuracy rates in tasks such as weapon detection and behavioral tracking, achieved through the use of multimodal data encompassing text, audio, and images [3]. Furthermore, a comparative study examines three deep learning configurations for encoding spatial and temporal patterns in crime data: encoding spatial patterns first, then temporal patterns; encoding temporal patterns first, then spatial patterns; and encoding both patterns in parallel. This study finds that deep learning models consistently outperform traditional methods and offer design recommendations for configuring and training these systems to optimize performance [4]. Collectively, these studies demonstrate the efficacy of deep learning in crime prediction and classification, emphasizing the importance of model selection, data integration, and methodological configuration to achieve optimal results.

3 Experiment and Results

3.1 Basic Crime Analysis using Heatmap

The heatmap reveals distinct patterns in crime distribution by day of the week. "LOITERING" and "PORNOGRAPHY/OBSCENE MAT" peak on Tuesdays, with the latter at 28.57%. "GAMBLING" incidents are higher on Wednesdays and Fridays (20.00%). "DRIVING UNDER THE INFLUENCE" is most common on Saturdays (20.97%) and Sundays (19.22%), indicating more offenses on weekends. Fridays show elevated rates for "EMBEZZLEMENT," "KIDNAPPING," and "VEHICLE THEFT," highlighting it as a day with higher criminal activity. These trends suggest specific days have higher occurrences of certain crimes, with weekends particularly prone to substance-related offenses.

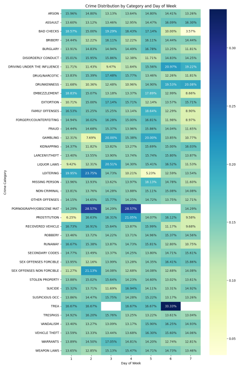


Figure 1: Crime Distribution by Category and Day of Week

3.2 K-Means Clustering

The map points are color-coded by clusters identified by the KMeans algorithm. A color bar on the right shows cluster indices from 0 to 4, indicating 5 clusters. Each dot represents a crime incident, with tooltips providing details such as coordinates, cluster number, and crime category (e.g., "VANDALISM").

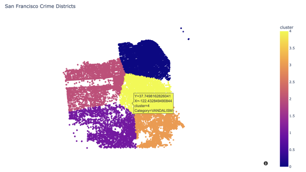


Figure 2: Plot shows different clusters of crime in San Francisco.

Two clustering configurations are evaluated: one with 5 clusters and the other with 3 clusters. The performance of these configurations is assessed using the Silhouette Score, which measures how similar an object is to its own cluster compared to other clusters.

- Clustering with 5 Clusters, Silhouette Score: 0.42657348350986973
- Clustering with 3 Clusters, Silhouette Score: 0.5082089205558781

Silhouette Score Analysis: Figure 3 illustrates the Silhouette Scores for different numbers of clusters (k). This graph helps to determine the optimal number of clusters by identifying the k value that maximizes the Silhouette Score.

Key Observations: The highest Silhouette Score is observed for 3 clusters, indicating that this configuration provides the most cohesive and well-separated clustering of the data.
As the number of clusters increases beyond 3, the Silhouette Score generally decreases, suggesting that the additional clusters may not provide meaningful distinctions and may introduce noise.

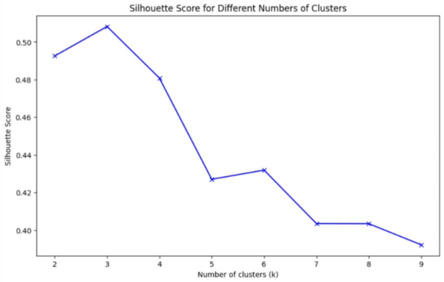
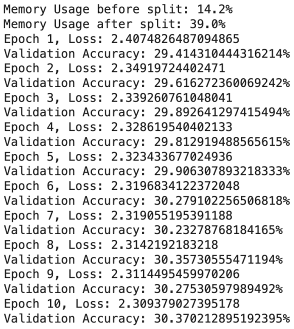


Figure 3: Graph shows Silhoutte Score Analysis.

3.3 Convolutional Neural Network

Our program sets up a CNN model for classifying crime data based on both textual and numerical features. The architecture leverages embedding layers for text, convolutional layers for feature extraction, and fully connected layers for classification. The CNNClassifier class provides methods for training and validating the model.

The model, named CrimeCNN, includes an embedding layer to convert text data into dense vectors, followed by a 1D convolutional layer (nn.Conv1d) with ReLU activation and max pooling. The output from the convolutional layer is flattened and concatenated with numerical features before passing through two fully connected layers (nn.Linear).



```
Memory Usage before split: 14.2%
Memory Usage after split: 39.0%
Epoch 1, Loss: 2.4074826487094865
Validation Accuracy: 29.414310444316214%
Epoch 2, Loss: 2.34919724402471
Validation Accuracy: 29.616272360069242%
Epoch 3, Loss: 2.339260761048041
Validation Accuracy: 29.892641297415494%
Epoch 4, Loss: 2.328619540402133
Validation Accuracy: 29.812919488565615%
Epoch 5, Loss: 2.323433677024936
Validation Accuracy: 29.906307893218333%
Epoch 6, Loss: 2.3196834122372048
Validation Accuracy: 30.279102256506818%
Epoch 7, Loss: 2.319055195391188
Validation Accuracy: 30.23278768184165%
Epoch 8, Loss: 2.3142192183218
Validation Accuracy: 30.35730555471194%
Epoch 9, Loss: 2.3114495459970206
Validation Accuracy: 30.27530597989492%
Epoch 10, Loss: 2.309379027395178
Validation Accuracy: 30.370212895192395%
```

Figure 4: Result after 10 epchoes of CNNclassifier

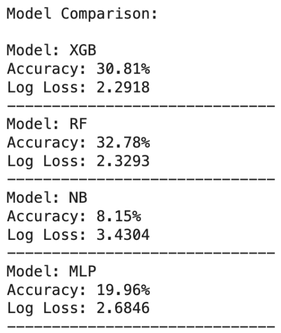
The training loss steadily decreases over the epochs, indicating that the model is learning and improving its fit to the training data. The validation accuracy shows a slight improvement, from 29.41% in the first epoch to 30.37% in the tenth epoch. However, the improvement is modest, suggesting that the model may not be generalizing well to the validation data.

The current architecture might not be complex enough to capture the nuances in the data. Experimenting with deeper networks or different architectures could help, such as adding more layers or using different activation functions. Also, experimenting with different batch size and learning rate can lead to better optimization and improve training efficiency.

3.4 Other Models

We preprocess the data by scaling numerical features and encoding categorical features. We extract date-related features and vectorize text data. We then train or fine-tune the model. If n_iter is greater than 0, we use RandomizedSearchCV for hyperparameter tuning. The data is split into training and validation sets. If n_iter is specified, we tune the hyperparameters; otherwise, we train the model with default settings. Finally, we evaluate the model on the validation set and record the results.

- **XGBoost (XGBClassifier):** Enables effective handling of complex data patterns.
    - max_depth: 3-10, learning_rate: 0.01-0.3, n_estimators: 50-200



```
Model Comparison:

Model: XGB
Accuracy: 30.81%
Log Loss: 2.2918
------------------------------
Model: RF
Accuracy: 32.78%
Log Loss: 2.3293
------------------------------
Model: NB
Accuracy: 8.15%
Log Loss: 3.4304
------------------------------
Model: MLP
Accuracy: 19.96%
Log Loss: 2.6846
------------------------------
```

Figure 5: Comparison between four models

- **Random Forest (RandomForestClassifier):** Utilizes ensemble learning for improved accuracy and robustness.
  - RandomForestClassifier(random_state=42)
  - n_estimators: 50-200, max_features: auto, sqrt, log2, max_depth: None, 10, 20, 30
- **Naive Bayes (GaussianNB):** Fast and effective baseline classifier for large datasets.
  - GaussianNB() with default parameters
- **Multi-Layer Perceptron (MLPClassifier):** Allows effective learning of complex relationships through deep learning.
  - MLPClassifier(random_state=42)
  - hidden_layer_sizes: (50,), (100,), (50, 50), activation: tanh, relu, alpha: 0.0001-0.05, learning_rate: constant, adaptive

The Random Forest model currently outperforms the others in terms of accuracy, but XGBoost has a lower log loss, indicating better-calibrated predictions. By focusing on further tuning and considering ensemble methods, the overall performance of the crime classification system can be improved. To improve the accuracy, the Random Forest model can increase the number of trees, adjust max depth, and experiment with feature selection strategies.

4 Conclusion

This project explored deep learning algorithms for crime analysis and prediction to enhance public safety by identifying high-risk areas and times. Using historical crime data, we applied K-Means clustering, CNNs, Random Forest, XGBoost, and Naive Bayes, each providing unique insights and accuracy levels.

We found specific crime patterns, like increased substance-related offenses on weekends, and identified distinct crime clusters in San Francisco. While CNN showed modest accuracy improvements, deeper networks and optimized hyperparameters could enhance performance.

Future work should integrate additional data sources, such as socioeconomic factors and real-time reports, and explore ensemble methods for improved robustness. This project shows the potential of deep learning in crime prediction, aiding more effective law enforcement strategies and contributing to safer communities.

Appendix

Our group's distribution of tasks is as follows:
Xiran Li: codes for CNN, RF, XGboost, MLP and NP models
Yinuo Tang: responsible for collecting information and writing the report
Dheeksha Mageswaran: Responsible for K-means Clustering and Silhoutte Score analysis

References

[1] Jyoti Agarwal, Renuka Nagpal, and Rajni Sehgal. “Crime Analysis using K-Means Clustering”. In: *International Journal of Computer Applications* 83.4 (2013). DOI: 10.5120/14496-2561. URL: https://www.researchgate.net/publication/269667894.

[2] Umair Muneer Butt et al. “Leveraging Transfer Learning with Deep Learning for Crime Prediction”. In: *PLOS ONE* 19.4 (2024). DOI: 10.1371/journal.pone.0296486. URL: https://doi.org/10.1371/journal.pone.0296486.

[3] Varun Mandalapu et al. “Crime Prediction Using Machine Learning and Deep Learning: A Systematic Review and Future Directions”. In: *IEEE Access* 4 (2023). DOI: 10.1109/ACCESS.2023.3286344. URL: https://ieeexplore.ieee.org/document/9676889.

[4] Panagiotis Stalidis, Theodoros Semertzidis, and Petros Daras. “Examining Deep Learning Architectures for Crime Classification and Prediction”. In: *Forecasting* 3.4 (2021). DOI: 10.3390/forecast3040046. URL: https://www.mdpi.com/2571-9394/3/4/46.