Machine-Learning-Driven Optimization For NSI Forecasting

Junyi Sheng, Jiaqi Huang, Shawn Jiang, Yuze Fu

December 2025

Abstract

This project forecasts monthly neighbourhood safety in the Greater Toronto Area using the Major Crime Indicators dataset. After cleaning temporal inconsistencies, we aggregate incident-level data into neighbourhood-month records and construct a new target Neighbourhood Safety Index (NSI). We engineer spatial features and temporal signals to further capture short-term dynamics. A range of models are evaluated: Ridge Regression, Polynomial Regression, KNN, Fully-Connected Neural Networks, and LSTM networks. The study compared these models' predictive power based on RMSE and R Squared. The prediction provides a data-driven foundation for safety-oriented resource planning across Toronto neighbourhoods.

1. Introduction

1.1 Problem Definition

In this project, we focus on the *Regression Project Stream – Predicting Safety Index*. Using the **Major Crime Indicators (MCI)** dataset from the **City of Toronto's Open Data Portal**, we aim to calculate a **Neighbourhood Safety Index** and develop regression models that can predict neighbourhood safety one month into the future. This work is motivated by the need for data-driven insights to help city planners and residents better understand patterns of safety and potential risks across Toronto's neighbourhoods.

1.2 Main Research Questions

The main research questions we investigate in this project are as follows:

1. How can we transform the raw Major Crime Indicators (MCI) dataset into a usable feature matrix suitable for regression modeling?
2. Which regression models and hyperparameter configurations yield the best predictive performance for forecasting the Safety Index one month ahead?
3. What types of features and feature engineering techniques most strongly influence model performance?

2. Data Preparation and Feature Representation

2.1 Data Cleaning

Before performing any feature extraction or modeling, we conducted a comprehensive data cleaning process to ensure the integrity and consistency of the Major Crime Indicators (MCI) dataset. The following key procedures were applied:

- **Missing Value Handling:** We detect missing values in `OCC_YEAR`, `OCC_MONTH` for some data points (crime events). We filled them by extracting information from the fully populated `OCC_DATE` column. There's no other missing value.
- **Consistency Checks:** To ensure data reliability, we verified data types, removed duplicate entries, and checked for temporal inconsistencies (e.g., mismatched reporting and occurrence years). No additional imputation required beyond reconstructing temporal items.

2.2 Data exploration with NSI

We noticed that each crime (data point) includes temporal, spatial, and crime category information. To predict the futural safety level for specific neighborhoods, we treat the temporal and spatial info as inputs, and incorporated crime category information by introducing the new target variable: NSI.

Severity Mapping and the Neighbourhood Safety Index (NSI):

We applied **severity mapping** to crime categories, and constructed the Neighbourhood Safety Index (NSI). Instead of using raw crime counts (treating all offences equally), we assigned a relative severity weight to crime type to reflect its social and physical impact. The mapping is summarized as follows:

Crime Type	Severity Weight
Robbery	5
Assault	4
Break and Enter	3
Auto Theft	2
Theft Over	1

These weights were chosen to reflect the relative seriousness of offences based on the Canadian Crime Severity Index (CSI), which assigns weights according to the severity of crimes as determined by actual court sentencing data. We analyzed the general trends in CSI weightings across multiple years to establish our relative weighting scheme. Violent crimes such as `Robbery` and `Assault` were assigned higher weights, while property-related crimes such as `Theft Over` received lower weights. This design ensures that the resulting NSI better represents the perceived safety level of each neighbourhood, as it is grounded in the empirically-derived severity framework used in official Canadian crime statistics.

For each neighbourhood n in a given month, a weighted crime score was computed as:

$$\text{TotalCrimeScore}_n = \sum_i (\text{count}_{i,n} \times \text{weight}_i),$$

where i indexes the five crime categories. The resulting score was then normalized and inverted to obtain a standardized **Neighbourhood Safety Index** (NSI) in the range $[0, 1]$:

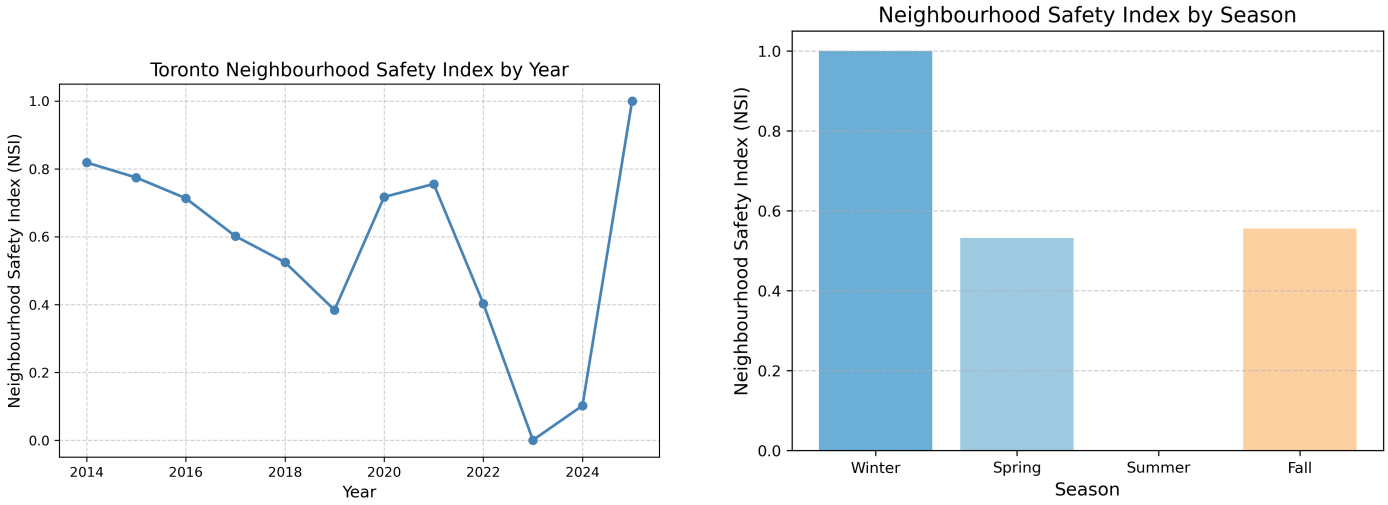
$$\text{NSI}_n = 1 - \frac{\text{TotalCrimeScore}_n - \text{MinScore}}{\text{MaxScore} - \text{MinScore}}.$$

A higher NSI value indicates greater neighbourhood safety. This transformation converts raw crime data into a continuous and interpretable target variable suitable for regression modeling. The following is some data exploration based on NSI:

- **Long-term Safety Trend (Fig. 1a):** The city-wide Neighbourhood Safety Index (NSI) exhibits a gradual decline from 2014 to 2020, indicating deteriorating safety conditions during this period. A brief recovery is observed in 2021–2022, followed by a sharp drop in 2023. The spike in 2024 likely reflects incomplete or early-year reporting rather than an actual improvement.
- **Seasonal Variation (Fig. 1b):** Winter demonstrates the highest NSI values, suggesting lower crime severity, while Summer and Fall present noticeably reduced NSI levels. This aligns with criminology literature showing that crime activity often increases during warmer months.

2.3 Feature Construction

- **Feature Aggregation:** Incident-level records were aggregated using a three-key grouping `(NEIGHBOURHOOD_158, REPORT_YEAR, REPORT_MONTH)`, where each unique combination defines one monthly datapoint for a neighbourhood.
- **Lagged Features:** We constructed the Neighbourhood Safety Index (NSI) by normalizing monthly crime severity scores, and created a lagged feature (`Prev_Month_NSI, NSI_3M_Avg`) to capture temporal dependencies in neighbourhood safety trends since classic ML models cannot learn the strong relation between the current month and the recent ones.



(a) NSI by Year

(b) NSI by Season

Figure 1: Exploratory visualizations of neighbourhood safety in Toronto.

Besides the key, these are the columns of our finalized dataset used for training:

1. `NSI`: target, as described above
2. `TotalCrimeScore`: feature, as described above
3. `Crime_Count`: feature, the number of crimes happened in each key
4. `x, y`: feature, the average location of crimes happend in each key
5. `Prev_Month_NSI`: feature, NSI of the same neighbourhood but previous month
6. `NSI_3M_Avg`: feature, average NSI of the same neighbourhood over the previous 3 months

3. Model Comparison Study

3.1 General Metric Selection

We evaluate model performance using two standard regression metrics: RMSE and R^2. **RMSE** serves as our primary metric because it measures prediction error in the original NSI scale and penalizes large deviations more strongly. This is particularly important in a safety context, where substantial mispredictions (e.g., overstating the safety of a high-risk neighbourhood) carry significant practical consequences.

In addition, we report the R^2 **score** to assess how much of the variation in NSI the model explains beyond simple averaging. While RMSE reflects absolute predictive accuracy, R^2 provides a complementary view of the model's ability to capture underlying crime-related patterns. Together, they offer a balanced and interpretable basis for comparing forecasting models in this study.

3.2 Linear Regression

Linear Regression serves as the baseline model in our study. It predicts NSI as a weighted sum of the input features, providing a straightforward interpretation of how neighbourhood and temporal factors contribute to safety. If more complex models fail to significantly outperform this baseline, it would suggest that NSI varies in a predominantly linear manner and does not require nonlinear feature transformations. Check Tab. 2 for the coefficients of our model.

We introduced regularization here because it helps reduce the potential overfit (brought by great great numbers of categorical variables). Multicollinearity caused by lagged features could also be reduced. We performed a grid search for hyperparameter alpha over Ridge Regression (L2 penalty). Not using Lasso because we don't want any feature to be removed. We tested multiple regularization strengths: $\alpha \in \{0.001, 0.01, 0.1, 1.0\}$. Our experiments showed that regularization term improved model performance than simple multi-linear regression. The best-performing alpha is 0.01.

3.3 KNN Regression

K-Nearest Neighbours (KNN) Regression is a non-parametric learning method that predicts target values by averaging the outcomes of the most similar observations in the feature space. We include this model in our comparison study because it was the first non-linear model introduced in the course, is conceptually simple, and generally performs well on regression tasks without requiring strong assumptions about data distribution.

We performed a comprehensive grid search over three hyperparameters: the *number of neighbours* k, the *weighting scheme*, and the *distance metric*. The search spanned $k = \{1, 3, 5, \ldots, 49\}$, both weighting options (`uniform`, `distance`), and two common metrics (`euclidean`, `manhattan`). Each configuration was trained using the same train–test split, and its performance was evaluated and recorded. This exhaustive exploration ensured that every combination was assessed rather than relying on heuristic parameter choices.

The tuning results demonstrate that k has the largest impact on model performance: very small values tend to overfit local noise, while excessively large values oversmooth the NSI signal. The best-performing configuration was obtained at $k = 25$, striking a balance between stability and responsiveness to neighbourhood patterns. In contrast, variations in the weighting scheme and distance metric produced only marginal differences in performance and did not materially alter the ranking of configurations. This confirms that selecting $k = 25$ is both empirically justified and sufficient for achieving optimal KNN performance in our forecasting task. Check Fig. 4 for the visualization.

3.4 FNN

A Fully-Connected Neural Network (FNN) is one of the most powerful function-approximation tools among the models we studied. Unlike linear and polynomial regression, an FNN can automatically learn useful patterns from data without requiring manually designed feature interactions.

We conducted a grid search over two neural architecture hyperparameters: the *hidden layer configuration* and the *activation function*. The tested architectures varied in depth and width (e.g., one to three layers with 32–256 neurons per layer), while activation functions included `relu`, `logistic` (sigmoid), and `tanh`. For each configuration, we trained a separate network using the same train–test split and evaluated performance using R^2 and RMSE. This exhaustive evaluation ensured that model quality was determined by systematic exploration rather than ad-hoc selection.

The comparative results show that deeper or wider architectures do not necessarily yield better performance; instead, moderate capacity balances expressiveness and generalization. Among all configurations, a two-layer network with `(64, 32)` neurons and a `logistic` activation achieved the best performance, obtaining the highest R^2 and lowest RMSE. Other activations and larger networks did not yield consistent improvements and sometimes degraded performance due to overfitting. These findings justify selecting the `(64, 32)` logistic network as the optimal FNN configuration for NSI forecasting. Check Fig. 6 for the visualization.

3.5 LSTM Neural Network

We implemented a LSTM neural network to help us predict the NSI target feature of the next consecutive month of a given 12 month sequence for a neighbourhood. The main modification to the "feed forward" is the addition of the LSTM cells; it allows us to remember information over time via a hidden state vector and cell state vector. We chose this model as we believe the relationship is not linear. Because the data is sequential we decide to use a LSTM NN with an input length of 12, to represent a sequence of a year for a particular neighbourhood, we use this context to predict the 13th month. We experiment with 2 LSTM layers and a fully connected layer, and note our best results in the following hyperparameter tuning process.

As our input has 9 features our input size follows. Our output size is just 1 for the NSI. Then we just tune the hidden vector size and number of layers. We use [32, 64, 128, 256] as our values for the hidden vector size. At first we tried with a hidden size of 32, 64 time steps, but we found that increasing to 128 time steps gave slightly better accuracy; this indicates that 128 time steps captures patterns that 64 cannot, 256 gave slightly

worse performance so we chose 128. For the number of layers we tested both 2 and 3 their performance were similar with 2 layers having slightly better performance, this leads us to choose our hyperparameters.

4. Main Results

Table 1: Model Performance Comparison (Poly Reg & SARIMA in appendix)

Model	**RMSE**	**R^2**
Linear Regression	0.0558	0.8143
Polynomial Regression (degree 2)	0.0560	0.8100
KNN	0.0561	0.8100
SARIMA	0.0568	−0.4963
FNN	0.0558	0.8139
LSTM	**0.0527**	**0.8295**

Linear Regression achieves the best RMSE (0.0558) while offering interpretability and computational efficiency, making it ideal for operational deployment. LSTM attains the highest R^2 (0.8476) through superior modeling of temporal dependencies, though at the cost of interpretability and computational overhead. KNN and Polynomial Regression show moderate performance with no clear advantages. SARIMA explicitly captures seasonality with excellent MAPE (2.41%), though its negative R^2 inadequately reflects time series performance.

We select LSTM as the primary model and Linear Regression as a complementary alternative. LSTM achieves both the best RMSE (0.0527) and highest R^2 (0.8295), demonstrating superior predictive accuracy and ability to capture complex temporal dependencies in NSI patterns. Its sophisticated sequential modeling makes it the strongest performer overall. Linear Regression offers a close second with RMSE of 0.0558 and R^2 of 0.8143, providing the key advantage of interpretability—coefficients directly reveal feature impacts on NSI for actionable policy insights. We recommend LSTM for applications prioritizing maximum predictive performance, and Linear Regression when model transparency and computational efficiency are critical.

5. Future Work

First, we want to integrate the premise of the crime into our NSI, as this could encode additional information about the overall crime of a neighbourhood (e.g. Apartment + Assult has weight 10 because of high danger level). About LSTM NN, we would like to experiment with different combinations of features as well as possibly using random features to try to furter maximize the performance of the LSTM NN. For SARIMA model, currently we used the city-wide NSI to determine the hyperparameters of the model and implemented the same model on all 158 sequences corresponding to 158 neighbourhoods. Instead, we can explore VARIMA to train a single model fitting to 158 sequences.

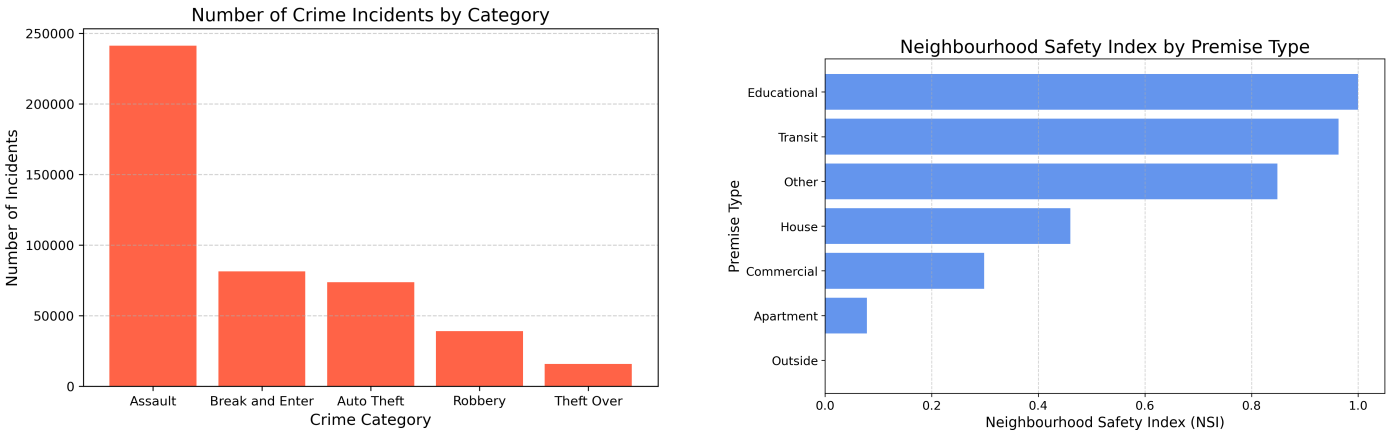
6. Originality

Our approach differs from some other similar attempts in our use of the LSTM neural network. Kang and Kang (2017) make use of a deep neural network to predict crime in a given location given the current conditions. Our approach differs in our use of the LSTM and past 12 month sequence of both temporal and spatial data to predict the crime of the next month.

In Kim et al. (2018), KNN regression is used to predict crime rates. In their approach they use very similar features in computing the distance. However we find an average location of crimes in a neighbourhood and use lagging features to help us find distances. This way we try to encode some sort of previous data about the sequence in our KNN regression approach. Additionally we define the average x and y of crimes in a neighbourhood in a month, which differs from the other approach, so closer neighbourhoods to the "wanted" neighbourhood will help influence our prediction. Our R-squared was 0.75 and performed much better than the other approach.

7. Appendix

7.1 Other Data Exploration



(a) Crime Category Distribution

(b) NSI by Premise Type

Figure 2: Exploratory visualizations of neighbourhood safety in Toronto.

- **Crime Category Distribution (Fig. 2a):** Assault constitutes the largest proportion of incidents, far exceeding other categories such as Break and Enter or Auto Theft. Theft Over is comparatively infrequent. This imbalance highlights the need for severity weighting rather than relying on raw incident counts.
- **Safety by Premise Type (Fig. 2b):** Crimes occurring outdoors and in apartment buildings correspond to lower NSI values, indicating higher-risk environments. In contrast, educational institutions and transit locations show comparatively higher safety scores. This variation supports the inclusion of contextual features in the model design.

7.2 Other Models

7.2.1 Polynomial Regression

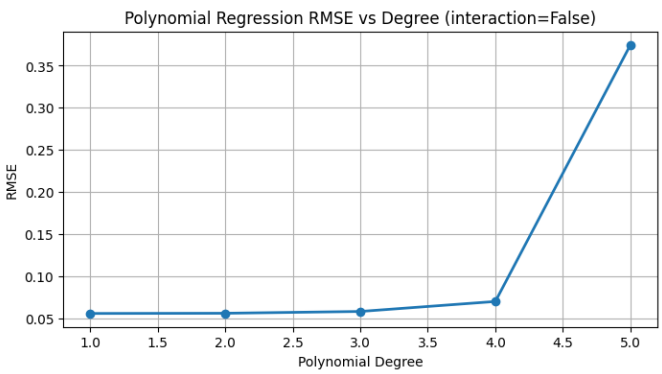
Polynomial Regression extends linear regression by introducing nonlinear transformations of the input features, enabling the model to capture curved temporal patterns in NSI that a purely linear model cannot represent. We include this model in our comparison study because it offers a lightweight way to introduce nonlinearity without the complexity of neural models, while remaining interpretable and suitable for regression tasks.

We performed an exhaustive grid search over two hyperparameters: the *polynomial degree* and whether to include *interaction terms*. We evaluated polynomial degrees $d = \{1, 2, 3, 4, 5, 6, 7\}$ under both interaction settings (`True`/`False`), training a separate model for each combination using the same train–test split and recording its performance. All results were stored and visualized using RMSE and R^2 curves, ensuring that the comparison reflects every configuration rather than a heuristic subset.

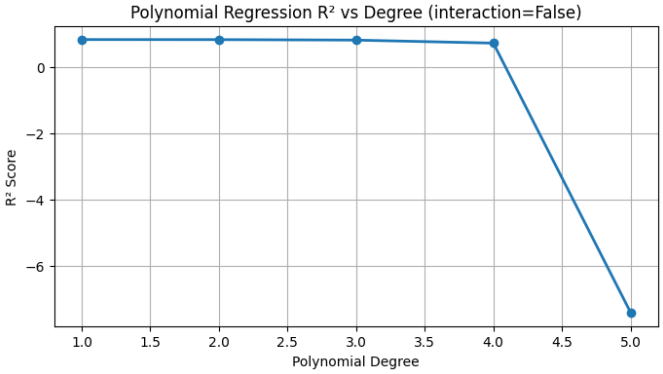
The tuning results show that performance improves when increasing the degree from $d = 1$ to $d = 2$, but deteriorates for $d \geq 3$: RMSE rises and R^2 falls, indicating overfitting. Interaction terms provided no consistent benefit and often worsened the results. By jointly selecting the configuration with the highest R^2 and the lowest RMSE, we identified a second-degree polynomial *without interaction terms* as the optimal model. This confirms that modest nonlinearity is sufficient for NSI prediction and that additional complexity harms generalization. Check Fig. 3 for visualization.

7.2.2 ARIMA

SARIMA extends ARIMA by incorporating seasonal patterns, enabling the model to capture the 12-month cyclical behavior in NSI. Unlike regression approaches requiring manual feature engineering, SARIMA automat-



(a) Polynomial RMSE



(b) Polynomial R2

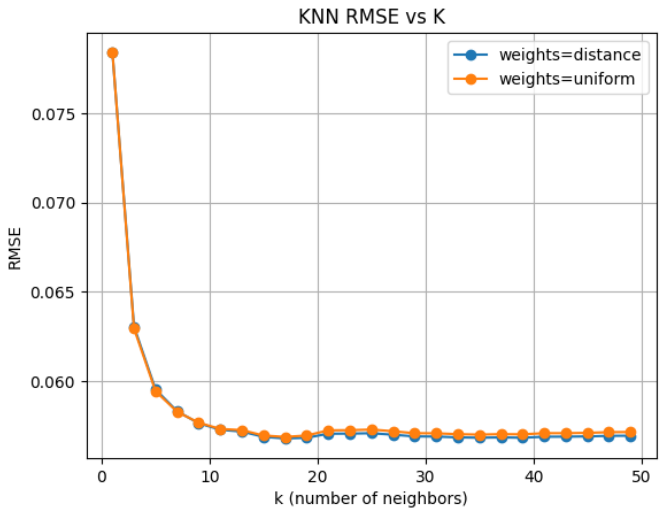
Figure 3: Hypertuning for Polynomial Regression.

ically models temporal dependencies through autoregressive and moving average components while differencing removes trends.

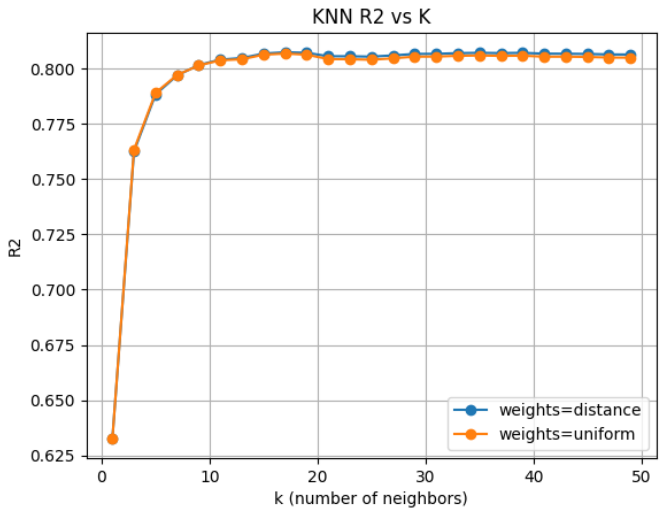
We evaluated five SARIMA configurations varying the order (p,d,q)(p, d, q) (p,d,q) and seasonal order $(P, D, Q, 12) : (1, 1, 1) \times (1, 1, 1, 12), (2, 1, 2) \times (1, 1, 1, 12), (1, 1, 2) \times (1, 1, 1, 12), (0, 1, 1) \times (0, 1, 1, 12), (1, 1, 0) \times (1, 1, 0, 12)$. Each model used an 80-20 train-test split, and performance was assessed using AIC, BIC, and MAPE.

The simplest configuration (0,1,1)×(0,1,1,12) achieved the lowest AIC (-527.62) and best test MAPE (2.41%), outperforming more complex alternatives. Adding autoregressive terms increased AIC without improving forecasts, indicating overfitting. The seasonal moving average term proved highly significant ($p < 0.001$), confirming the importance of 12-month patterns. This demonstrates that modest complexity with seasonal error correction is sufficient for NSI prediction, and that additional parameters harm generalization. The prediction of the city-wide NSI is the following Fig. 5:

7.3 Other Figures For Justification of Hypertuning

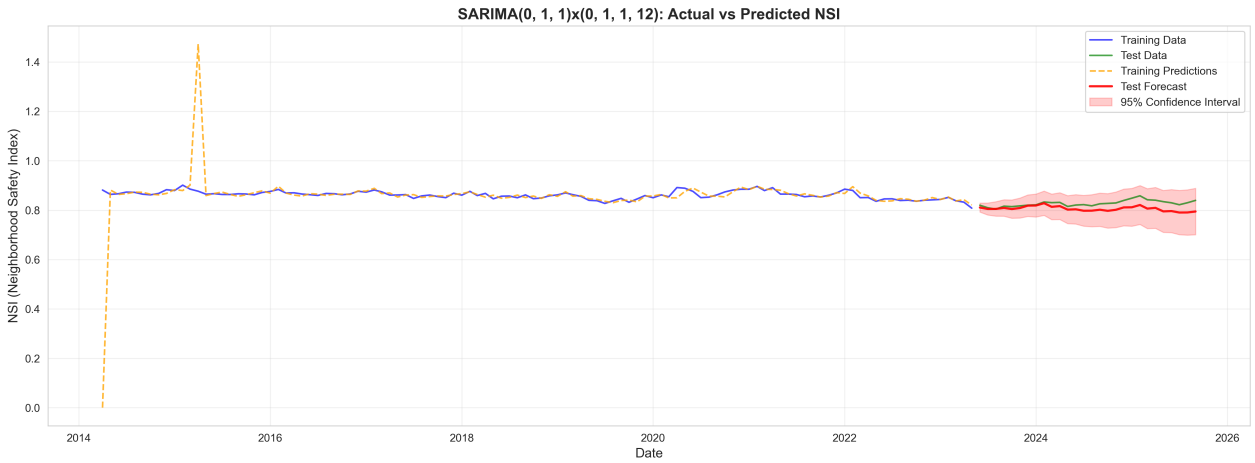


(a) KNN RMSE

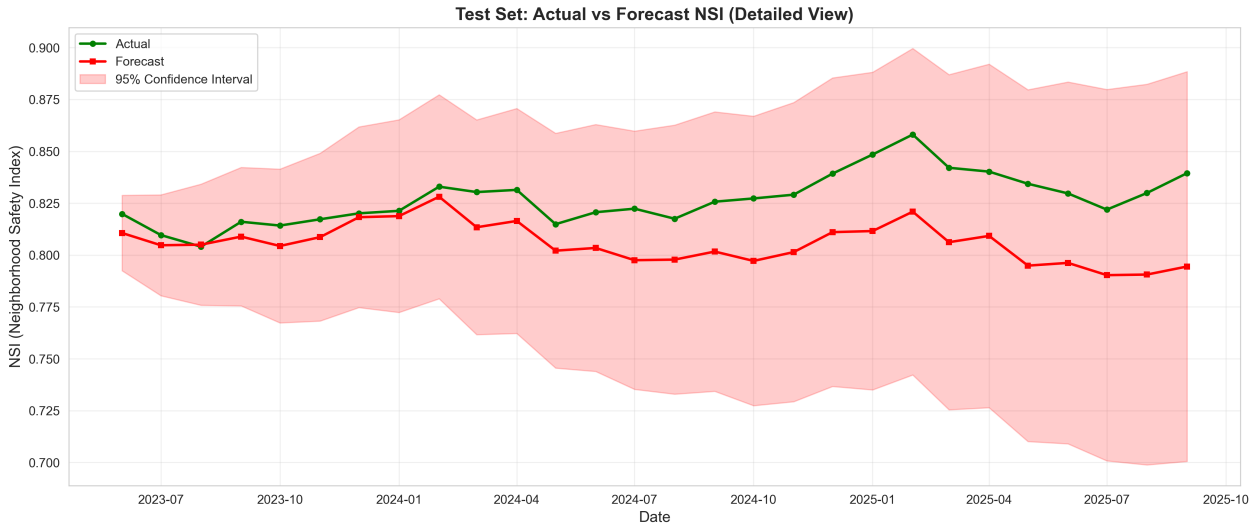


(b) KNN R2

Figure 4: Hypertuning for KNN.



(a) Predict vs Actual (Whole Serie)



(b) Predict vs Actual (Test Serie)

Figure 5: Prediction by SARIMA

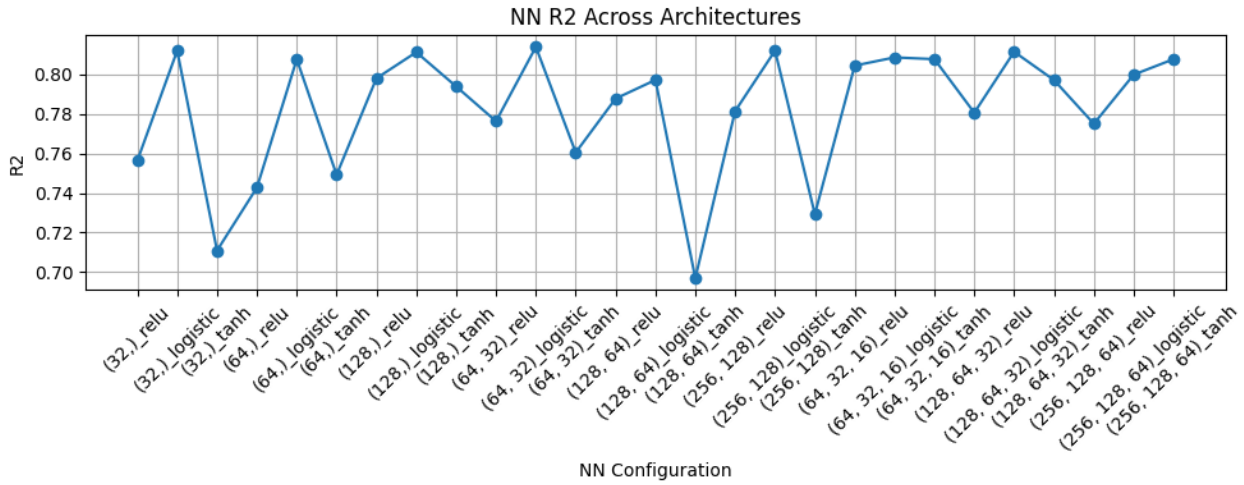


Figure 6: Hypertunning for FNN Model

Table 2: Linear Regression Feature Coefficients (Top and Bottom 5)

Feature	**Coefficient**
Intercept	3.334816
`NSI_3M_Avg`	0.503455
`C(HOOD_158)[T.020]`	0.162466
`C(HOOD_158)[T.012]`	0.159333
`C(HOOD_158)[T.019]`	0.155635
⋮	
`C(REPORT_MONTH)[T.12]`	−0.000935
`C(HOOD_158)[T.147]`	0.000570
`y`	0.000003
`x`	0.000002

8. Acknowledgment

We thank the professor and TAs for their guidance and support over the past semester. This project would not be possible without them. We hope you enjoyed our report and presentation as much as we enjoyed working on this project.

References

Kang, H.-W., & Kang, H.-B. (2017). Prediction of crime occurrence from multi-modal data using deep learning. *PLOS ONE*, *12*, 1–19. https://doi.org/10.1371/journal.pone.0176244

Kim, S., Joshi, P., Kalsi, P. S., & Taheri, P. (2018). Crime analysis through machine learning. *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 415–420. https://doi.org/10.1109/IEMCON.2018.8614828