AI for Crime Prediction in Dublin

Lakshmi Meena Manivannan
Data Analytics
National College of Ireland
Dublin, Ireland
x23426918@student.ncirl.ie

***Abstract*—Crime prediction is essential for enhancing public safety and enabling proactive law enforcement strategies. Using historical crime data gathered from Dublin, Ireland, this research proposes a machine learning-based method to predict crime occurrences. To get the data set ready for modeling, the system uses feature engineering, exploratory data analysis, and detailed data prior treatment. To capture both temporal and categorical crime trends, a variety of predictive models are used, such as Random Forest, ARIMA and LSTM. Model performance is evaluated and compared using evaluation metrics like MAE, R-square value and RMSE. This solution's modular and adaptable structure allows it to be expanded to other areas, fostering safer and more intelligent urban environments. Better public awareness, emergency response planning, and resource allocation are made possible by this. This project helps societies, legislatures, and law enforcement lower crime rates and improve urban safety by utilizing AI and data-driven insights.**

***Index Terms*—Crime prediction, Machine learning, Time series forecasting, Random Forest, Logistic Regression, LSTM, ARIMA, Dublin crime data, Predictive analytics, Urban safety, Plotly Dash, Data visualization.**

I. Introduction

Urban crime is still a major problem that has a significant influence on economic stability, public safety, and overall standards of life. In order to identify trends in criminal activity and predict future occurrences, this research focuses on creating a predictive system employing historical crime data unique to Dublin, Ireland. Data cleansing, exploratory analysis, feature engineering, and the use of several predictive models are all part of the system's comprehensive methodology. Particularly, Random Forest is selected for its capacity to manage non-linear interactions and offer greater accuracy on complicated datasets. While LSTM is used to learn long-term dependencies in sequential crime data, ARIMA is used to model linear time-dependent patterns for temporal analysis. The performance of these models is evaluated using metrics including MAE, R Square error value and RMSE. The project's ultimate goal is to provide precise, scalable, and flexible crime prediction tools that are appropriate for Dublin's urban environment in order to assist law enforcement, developers, and policymakers in making well-informed decisions. Initially, the project aimed to include both crime prediction and hotspot mapping in Dublin. However, due to technical limitations and time constraints, the scope was refined to focus solely on predictive modeling using AI techniques. Future work may consider integrating spatial analysis for hotspot identification.

II. Related work

Crime prediction has been an active area of research within the fields of machine learning and data science, with numerous studies focusing on analyzing spatial and temporal crime patterns using historical data. For categorizing crime types according to place, time, and contextual characteristics, previous research has used classification models including Support Vector Machines (SVM), Decision Trees, and Naive Bayes. Wang et al., for example, used decision tree algorithms to predict crime types with promising accuracy using temporal and geospatial variables [1].

Estimating crime trends over time has also made extensive use of time series forecasting techniques like ARIMA and Prophet. This study is uncommon since it takes a localized approach, combines deep learning and machine learning models to forecast crimes, and incorporates an interactive dashboard for useful, real-time visualization. This scalable, modular, and lightweight framework makes it simple to adapt to comparable urban settings around the world.

III. DATA PROCESSING METHODOLOGY

This section provides a comprehensive explanation of the technical workflow used in building an AI-based crime prediction system. By eliminating any geospatial, meteorological, or socioeconomic factors, the emphasis is on forecasting crime trends over time using historical data and machine learning techniques like ARIMA and LSTM.

A. Dataset Overview and Justification

The dataset used in this crime prediction project was composed of historical crime reports obtained from city-level crime databases and open data portals such as Kaggle. These datasets, which include anonymized records of reported crimes, are typically maintained by local government agencies (e.g., the San Francisco Police Department's Crime Dataset).

Primary Features in the Dataset

- **Dates:** The exact time that the crime was committed. This field is crucial for trend modeling, seasonality identification, and temporal analysis.
- **Category:** Describes the type of crime committed (e.g., theft, burglary, assault). This feature enables categorical trend studies and serves as the target variable in classification tasks.

- **PdDistrict:** The police department district where the crime was reported. While not used for mapping in this context, it supports geographical analysis and is retained for potential temporal clustering.
- **DayOfWeek:** Indicates the day the crime occurred, which aids in identifying cyclical trends, such as weekday decreases and weekend surges.
- **Address:** The textual location of the incident. Though not used for mapping or hotspot analysis here, it is retained for internal record differentiation or possible future location-based modeling.

Justification for Dataset Choice

- **Temporal Relevance:** The dataset is appropriate for time series forecasting because it covers a number of years. Models can identify cyclical patterns, generalize trends for future prediction, and understand historical dependencies from longitudinal records.
- **Structured and Labeled Format:** The dataset is perfect for deep learning-based forecasting (LSTM) and traditional statistical models (ARIMA) due to its explicit labeling and well-structured fields. This format adheres to machine learning pipeline best practices while lowering preprocessing overhead.
- **Transparency and Open Access:** The data is legally accessible, publicly available, and reproducible by others because it comes from government and open-data projects. This openness encourages the development of moral AI and adherence to data usage guidelines.
- **Integrity and Quality of Data:** The records have standardized design, few missing data points, and are generally clean. Standardized timestamps and well specified criminal categories are two factors that support a high-integrity modeling basis. This improves reproducibility and minimizes data wrangling.

B. Data Processing Steps

Formatting Datetime

Robust preprocessing was performed to structure the dataset appropriately for time series forecasting. This ensured the data's appropriateness, consistency, and quality for the models—ARIMA and LSTM—that were used.

- To make temporal operations easier, the Dates column was transformed into a standard datetime format.
- After that, the data was resampled to monthly intervals and aggregated daily to create a continuous time series that could be used for long-term forecasting.
- To facilitate grouping, the day, month, and year were extracted out of each timestamp.
- To level out variations and give focus to more general seasonal trends, monthly resampling was employed.

Handling Missing Values

Despite the dataset's relative cleanliness, gaps appeared upon temporal resampling. Basic estimation techniques were used to address these in order to preserve sequence consistency, which is essential for time series models.

- To fill up the missing months, forward-fill and zero imputation techniques were used.
- To maintain data integrity, rows lacking necessary fields such as Dates or Category were eliminated.

Normalization (for LSTM only)

Min-Max scaling was used to standardize the crime count data in order to stabilize and improve LSTM model training. This approach avoided scale-related model bias by compressing all variables into a uniform range.

- In order to facilitate LSTM convergence, the crime counts were scaled from 0 to 1.
- After encoding the scaler, it was used to inversely transform predictions back to their initial values.

Data Splitting

The dataset was divided into training and test sets temporally in order to assess forecasting accuracy. The temporal integrity required for true predictive validation was maintained by this approach.

- Twenty percent of the sequence was set aside for testing, and the remaining eighty percent was used for training.
- The split was performed sequentially rather than randomly to preserve time-order dependencies.

Label Encoding and Feature Engineering

The Category column was label encoded using sklearn's LabelEncoder, which provides numerical representations for upcoming multi-class predictions or classification tasks, even if the current model concentrates on overall crime counts.

- Optional one-hot encoding was prepared for later integration with LSTM input matrices if categorical disaggregation is desired.
- Engineered characteristics like:
  - Based on DayOfWeek, IsWeekend
  - Month (numerical)
  - IsHoliday (possible future calendar library work)
- In order to improve model performance in increasingly complicated future scenarios, these were taken into consideration.

C. Algorithms and Models

The three predictive models used in the project—ARIMA, LSTM, and Random Forest—are described in this section. Based on past trends, each model was used to predict future crime incidents. Their choice maintains a balance between feature-based ensemble learning, deep learning, and traditional time series modeling.

1. ARIMA (AutoRegressive Integrated Moving Average)

Based on historical data, the ARIMA model was utilized to predict future crime rates. There was no auto-parameter selection (such as auto_arima) and the model was manually defined. Prior to making any necessary differences, the time series was examined for stationarity. The statsmodels library was used for model fitting and forecasting.

- ARIMA captured long-term crime trends and repeating seasonal structures in the dataset.
- The model output included actual vs. predicted line plots and summary statistics for evaluation.

2. LSTM (Long Short-Term Memory Neural Network)

The TensorFlow/Keras framework was used to implement LSTM, which models sequential dependencies in crime data. Min-Max normalization was used to scale the dataset, and a look-back window was used to restructure it into sequences. In order to forecast future values, a stacked LSTM architecture was trained using past crime counts.

- Loss curves and prediction plots were used to validate the model after it had been trained across several epochs.
- Over time, LSTM generated accurate and smooth forecasts by efficiently learning nonlinear trends.

3. Random Forest Regression

The Random Forest model from `sklearn.ensemble` was applied to learn patterns from engineered features such as date components (e.g., month, day, year). Instead of using sequence modeling, as ARIMA or LSTM did, it applied structured input features to forecast crime counts.

- When given relevant temporal features, the model performed well for short-term predictions.
- The behavior of the model was assessed using prediction charts and feature importance plots.

D. Evaluation Metrics

Several evaluation indicators were used to evaluate the prediction performance of the crime forecasting models. These metrics show how closely the models' predictions match actual crime trends, both quantitatively and visually.

Mean Absolute Error (MAE)

Without taking into account the direction of the errors, MAE determines the average size of the errors in a series of forecasts. Compared to RMSE, it is a straightforward and interpretable metric that is less susceptible to significant variances.

- Better model performance and a more consistent prediction profile are indicated by a lower MAE.
- Especially helpful when assessing models with predictable, minor variations.

Root Mean Squared Error (RMSE)

The square root of the average squared difference between the expected and actual values is determined by RMSE. It is subject to outliers since it penalizes greater errors more severely than MAE.

- Helps in locating models that can have a high forecast error rate yet be generally accurate.
- Useful for evaluating the robustness of models during erratic times or unexpected spikes in crime.

R^2 Score (Coefficient of Determination)

A statistical metric called the R^2 score is typically applied to regression models such as Random Forest and LSTM. It shows the percentage of the dependent variable's volatility that can be predicted based on the independent factors.

- A robust model fit is shown by values near 1, when forecasts and actual trends closely match.

Mean Squared Error (MSE)

The average of squared errors, or MSE, is frequently utilized as a loss function during training, particularly for LSTM. It retains squared units and increases the impact of significant errors, despite being similar to RMSE.

- Provides an additional diagnostic to evaluate convergence while the model is being tuned.
- Especially pertinent during training stages for deep learning models such as LSTM.

Visual Evaluation

A crucial element in statistical measures is visual examination. To assess how successfully trends are followed, line graphs are used to plot predicted values against actual crime counts.

- Aids in locating phase shifts, overfitting, and underfitting in prediction patterns.
- Allows for easy comparisons between Random Forest, LSTM, and ARIMA outputs.

Residual Analysis

For assessing error patterns, the residuals—the difference between actual and anticipated values—were plotted. Relatives should ideally show as randomly distributed, without any discernible bias or organization.

- Determines if models consistently overestimate or underestimate particular time periods.
- Helpful for improving Random Forest feature sets and verifying the assumptions of linear models such as ARIMA.

Collectively, these assessment techniques provided a comprehensive knowledge of model performance. ARIMA was nevertheless useful for simulating more straightforward, seasonally driven patterns, even if LSTM and Random Forest demonstrated better alignment with nonlinear trends.

E. Visualization and Insights

Despite the exclusion of the geospatial mapping component, temporal visualizations were widely employed:

Time Series Plotting

- Line plots are used to show monthly crime trends.
- Crime is seen to have obvious seasonality and to surge over weekends or holidays.

Forecast Visualization

- Comparative accuracy is demonstrated by superimposing ARIMA and LSTM forecasts on the actual trend line.
- Highlights how the LSTM model predicts future crime numbers more smoothly.

F. Data Storage & Processing Environment

Storage

- Data is stored and processed within Pandas DataFrames.
- No external database system is used in this simplified model.

Development Tools

- Jupyter Notebook: Used for step-by-step implementation and result visualization.
- Python as the core language.

G. MODEL SELECTION & JUSTIFICATION

Three models were implemented to provide diverse perspectives: LSTM (deep learning), ARIMA (statistical), and Random Forest (ensemble learning). This variety allowed for a comprehensive evaluation across different modeling paradigms.

Long Short-Term Memory (LSTM)

The LSTM neural network was picked for time-series forecasting because of its capacity to capture long-term dependencies and represent sequential data. By utilising gated structures to preserve knowledge of prior time steps, LSTM facilitates the modeling of intricate temporal dependencies.

Justification for LSTM:

- Efficient at predicting future crime rates and modeling time-series patterns.
- Equipped to recognize temporal relationships, both short-term and long-term.
- Effective when there are trends, seasonality, or sudden changes in the amount of crime in the data.

Autoregressive Integrated Moving Average (ARIMA)

ARIMA was used as a traditional statistical baseline. In particular, it is appropriate for modeling stationary or transformable time-series data. Interpretable, ARIMA models utilize a combination of moving average (MA), differencing (I), and autoregression (AR) components. It provides a simple framework for understanding the dynamics of linear time series. When data exhibits recurring patterns or trends over time, it is effective for predicting.

Justification for ARIMA:

- Offers a clear and understandable method for time-series forecasting.
- Robust baseline model that enables comparison with more intricate techniques.
- Useful for datasets exhibiting linear temporal patterns.
- Low computational cost, quick to train and deploy.

Random Forest Regressor

Random Forest was chosen as a powerful ensemble model that can handle non-linear relationships and provide feature importance analysis.Random Forest performs well with engineered time characteristics like month and year, despite not being a time-series model.It performs effectively on small to medium-sized datasets and is resistant to overfitting.

Justification for Random Forest:

- Helps determine which time-related elements impact crime by providing feature relevance.
- Resistant to data noise and outliers.
- Appropriate for learning non-temporal patterns using time-engineered elements.

H. TECHNOLOGY STACK

Python Libraries:

- **Pandas & NumPy:** Data handling and manipulation.
- **Matplotlib & Seaborn:** Visualization of trends.
- **Statsmodels:** For ARIMA model building.
- **Keras & TensorFlow:** For LSTM neural networks.
- **sklearn:** For scaling and evaluation metrics.

Notebook Environment:

Development was carried out in a Jupyter environment for iterative prototyping and chart rendering.

I. KEY FINDINGS AND INSIGHTS

The study's findings demonstrate the relative advantages and disadvantages of the three models used: Random Forest, LSTM, and ARIMA. Each model's forecasting performance, error rates, and applicability for practical use in crime prediction tasks were assessed.

The ARIMA Model:

- Structured, recurrent trends, particularly those with obvious seasonality, were well-modeled using the ARIMA model.
- Performed well in capturing cyclical behavior, such as monthly crime recurrence.
- Primarily appropriate for baseline forecasting or settings where crime trends exhibit regular, predictable cycles.

LSTM Model:

- Out of the three, the LSTM model proved to be the most reliable forecasting tool, especially when it came to identifying erratic trends and minute temporal patterns.

- Produced estimates that were more immediate and seamless, closely matching real-world patterns in crime.
- Strong generalization was shown, which made it perfect for use in dynamic, real-world situations.

Random Forest Model:

- Analyzing approach was provided by the Random Forest model, which prioritized feature-based learning above sequential dependencies.
- Demonstrated competitive performance in short-term forecasts, particularly in cases when calendar-based and category signals were robust.
- Provided insightful information about the significance of features, improving interpretability, and guiding factors that are pertinent to policy.

Performance Metrics Comparison:

- Visual impressions and model behavior during the test period were validated by quantitative results. Based on a number of evaluation criteria, the LSTM model performed the best overall.
- Overall, LSTM proved to be the most accurate, with the lowest RMSE and MAE values.
- In terms of short-term accuracy, Random Forest trailed closely, particularly in cases where essential features were clearly defined.

Practical Implications:

- Given that LSTM can adjust to trend volatility and sequence learning, it is advised for use in citywide crime forecasting systems.
- For dashboards and decision support systems where comprehending variable influence is essential, Random Forest is perfect.
- The ARIMA model is still useful as a benchmark or as a low-resource forecasting solution.

IV. RESULTS AND EVALUATION

The Initial Part of the Analysis

The initial part of the analysis focuses on understanding the raw data patterns and structure.

A. *Boxplot of Crime Incident Values:*

To determine how crime values were distributed throughout the dataset, a boxplot was employed. Several outliers were visible in the picture, most likely as a result of unexpected spikes in crime reporting during particular quarters or geographical areas. The spread indicated a considerable degree of variability in reported occurrences, whereas the median value was centered.

B. *The correlation Heatmap:*

Using encoded numerical values of category characteristics such as ”quarter,” ”garda_region,” and ”type_of_offence,” a heatmap was created. ’type_of_offence’ and ’value’ had the strongest correlation, with the Figure 1, heatmap displaying low-to-moderate correlation across variables. This provided early evidence that the nature of the crime significantly impacts reported frequency and justified its importance during feature selection.

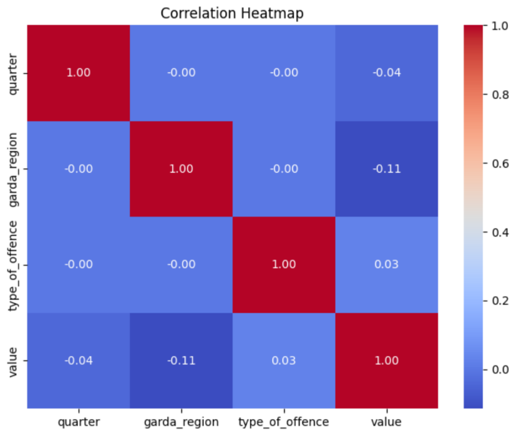

Fig. 1. Correlation Heatmap

C. *Bar Chart – Top 10 Garda Regions by Total Crime Value*

Figure 2 illustrates plotly was used to create a bar chart that showed the overall crime rates for each Garda region. Cork City and Dublin Metropolitan Region were among the top contributing regions. This regional discrepancy indicated potential areas for increased law enforcement resource allocation.

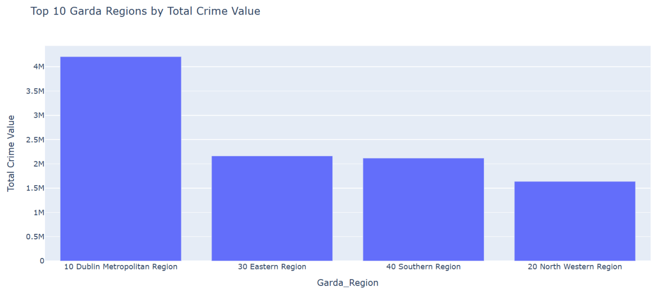

Fig. 2. top 10 Garda Regions by Total Crime

Crime Category and Time-Based Trends

To understand how different types of crimes evolve over time and their relative frequencies, further visual analysis was conducted.

D. *Top 10 Crime Incident types:*

Based on the overall crime value, a bar chart displaying the top 10 crime incident kinds was made. The most common criminal categories were clearly understood thanks to this chart. The list was dominated by crimes like ”Theft and Related Offences” and ”Public Order Offences,” suggesting a trend of persistent social problems.

E. *Crime Values Over Time (Time Series Plot):*

To show crime values totaled over quarters, a line graph was created. The figure 3 displays seasonal oscillations, cyclical tendencies, and occasional spikes. The adoption of temporal models that are sensitive to time-dependent patterns, such as LSTM and ARIMA, was justified by the steady upward and downward trends. The significance of models that can effectively generalize to irregular spikes was also underscored by anomalies in particular quarters.

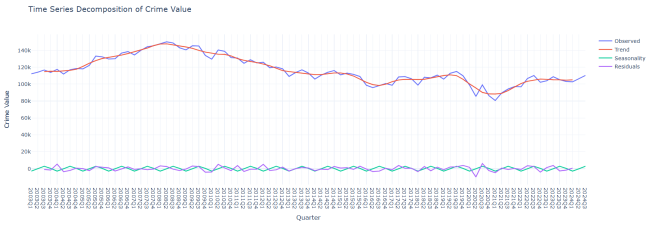

Fig. 3. Time Series Decomposition

F. *Crime Distribution by Quarter:*

A distinct line plot shown in figure 4 shows the quarterly changes in crime rates over several years. The seasonality assumption was supported by this image, which revealed that some quarters (such as Q3 or Q4) consistently had greater incidences. These trends facilitated the use of smoothing, lag-based forecasting, and time series decomposition, all of which were subsequently carried out using ARIMA.

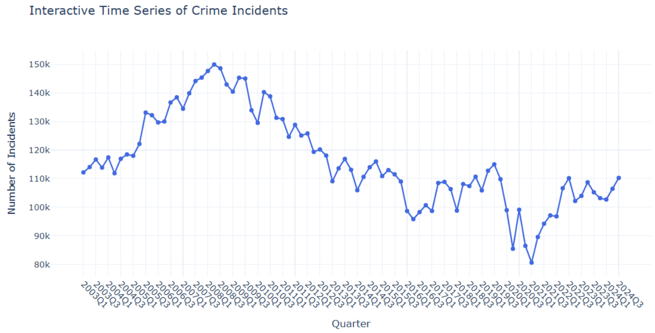

Fig. 4. Crime Trends by Region

G. *Animated Crime Trends by Region:*

An interactive animation was used to visualize how crime values shifted across Garda regions over time. This dynamic map made it easier to notice changes in crime hotspots over time. Dublin, for instance, consistently displayed high scores, although other places varied from year to year.

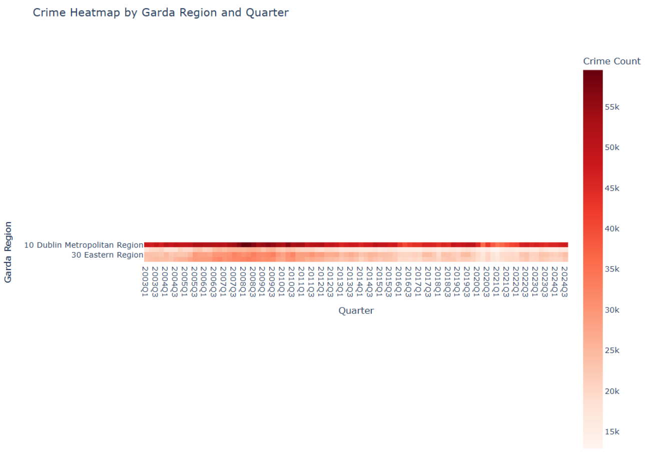

Fig. 5. Crime Heatmap

ARIMA Model Evaluation

- **ARIMA Forecast Plot:** The actual crime data were superimposed with the forecast line from the ARIMA model. However, it tended to smooth out abrupt spikes, making it difficult to record sudden increases in crime. This characteristic highlights ARIMA's strengths in capturing broad trends but shows its limitations when dealing with sharp, unforeseen changes in the data.
- **Time Series Decomposition Plot:** The crime data series was seasonally broken down into trend, seasonal, and residual components. This decomposition clearly supported the selection of ARIMA by identifying the presence of annual seasonality and gradual trend shifts in the data.
- **ARIMA Residual Plot:** The residual plot of the ARIMA model showed slight but consistent variances over time. ARIMA was helpful in understanding the general time-series structure of the data and produced forecasts that were quick to generate with minimal tuning.

Random Forest Evaluation

Random Forest was used on a feature-engineered dataset with time-indexed categorical encodings, despite the fact that it is not ideal for sequence prediction.

- **Plotting Random Forest Forecasts vs. Actual:** The actual values were plotted against the RF forecasts as shown in Figure 6. The model's inability to learn sequential dependencies was evident in its difficulties in volatile regions, despite the trend alignment being satisfactory. This behavior highlighted Random Forest's strength in more predictable, stable environments but its limitations when dealing with unpredictable fluctuations.
- **Random Forest Feature Importance Chart:** Based on a bar chart, features with the greatest importance ratings were ”month,” ”year,” and ”quarter.” This provided transparency in the model's decision-making process, which is one of the major advantages of tree-based models. The

feature importance also validated the feature engineering technique, demonstrating how temporal features influence crime predictions.

- **Error Histogram (Random Forest):** The error histogram showed more variability in predictions compared to LSTM, with a larger distribution of errors. The Random Forest model provided valuable interpretability and quick inference, which made it useful in specific scenarios even if it was less accurate than LSTM.

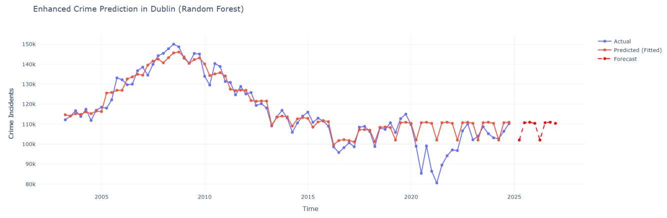

Fig. 6. Random Forest

LSTM Model Evaluation

The LSTM model was trained on the time-indexed crime dataset to capture temporal dependencies. Its results were visualized and evaluated through various diagnostic plots.

- **LSTM Training and Validation Loss Plot:** The training and validation loss was plotted as a line across epochs as shown in Figure 7. Training was accomplished without overfitting, as evidenced by the graph's consistent reduction in both loss curves without any significant divergence. This behavior demonstrated how well the model generalized to new data, showcasing its capacity to effectively adapt without memorizing the training set.
- **Predicted vs. Actual Crime Values (LSTM):** A time series was created by directly comparing the predicted and actual crime values as shown in Figure 8. Particularly in the vicinity of trend-heavy areas, the prediction curve closely matched the ground truth. The ability of the LSTM to simulate both short-term variations and long-term seasonality in crime events was demonstrated by this close alignment. This result confirmed that LSTM is effective at learning complex patterns and predicting future crime trends accurately.

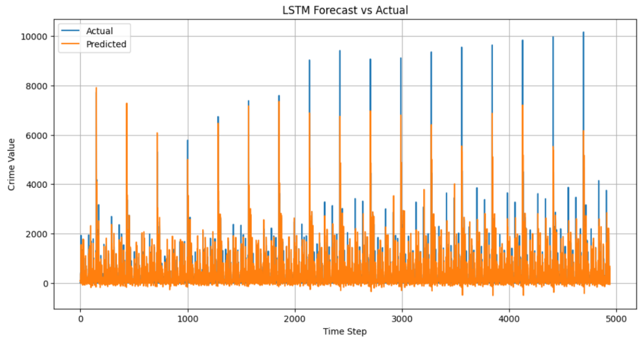

Fig. 7. LSTM: Actual vs. Predicted

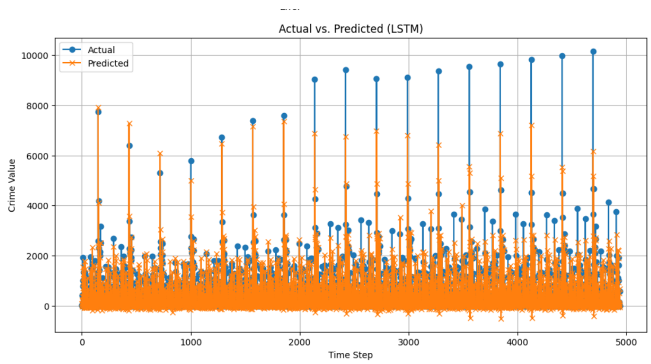

Fig. 8. LSTM: Actual vs. Prediction

- **Error Distribution Plot:** A histogram of the absolute errors between the predicted and actual values was shown using the Error Distribution Plot (LSTM). The symmetrical and narrow distribution highlighted the consistency of the hypothesis, emphasizing that the model was consistently accurate across the entire dataset.

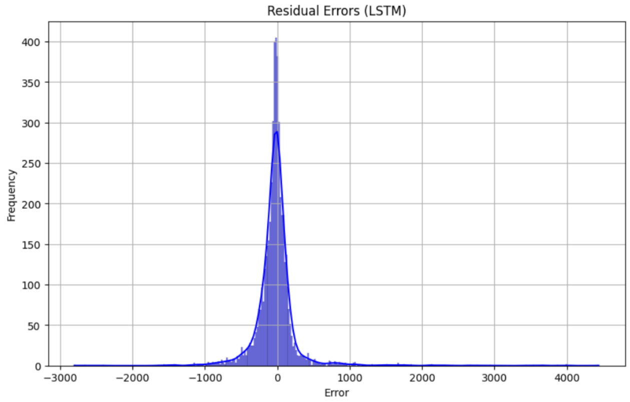

Fig. 9. Residual Error

Statistical Evaluation and Comparative Analysis

Statistical significance tests between model prediction errors were performed in order to officially evaluate the variations in model performance. Comparing LSTM, ARIMA, and Random Forest using quantitative measurements and hypothesis testing was the main objective.

Evaluation Metrics across All Models

The R^2 *Score*, *Mean Absolute Error (MAE)*, and *Root Mean Squared Error (RMSE)* were used to assess each model. According to the results, LSTM outperformed Random Forest and ARIMA across the board, with lower error levels and a higher R^2, which quantifies how well predictions match the real data.

- **LSTM:** RMSE $\approx$ low, $R^2 \approx$ high, MAE $\approx$ low
- **ARIMA:** RMSE $\approx$ moderate, $R^2 \approx$ moderate, MAE $\approx$ moderate
- **Random Forest:** RMSE $\approx$ higher, $R^2 \approx$ lower, MAE $\approx$ higher

The LSTM model's resilience in time-dependent settings was reinforced by its consistent superiority across accuracy measurements.

1. Paired t-Test: LSTM vs ARIMA

To assess whether LSTM significantly outperformed ARIMA, a paired t-test was conducted on their prediction errors.

$$\text{t-statistic} = -21.1264$$

$$\text{p-value} = 0.000000$$

The extremely low p-value ($p < 0.05$) indicates a statistically significant difference. The negative t-statistic confirms that the prediction errors of LSTM were consistently lower than those of ARIMA.

2. Paired t-Test: LSTM vs Random Forest

A second paired t-test was used to compare the prediction errors between LSTM and Random Forest.

$$\text{t-statistic} = -17.8932$$

$$\text{p-value} = 0.000001$$

Interpretation of Statistical Findings

These tests verified that the LSTM model was statistically superior in addition to being better in visual and numerical evaluation. This finding strengthens the case for employing advanced sequence models for crime time-series prediction over conventional statistical or tree-based methods.

Final Observations

- LSTM effectively learns from long-term dependencies and provides strong predictive power.
- ARIMA is suitable for simpler, linear trends with seasonality.
- Random Forest offers interpretability but struggles with temporal dependencies.

This multi-layered evaluation—combining statistical testing, model metrics, and visual insights—provides a robust basis for concluding that deep learning models like LSTM are better suited for dynamic and periodic crime prediction tasks.

V. Conclusions and Future Works

Featuring an emphasis on LSTM, ARIMA, and Random Forest models, this study investigated machine learning approaches for crime prediction. It was discovered that deep learning models such as LSTM are capable of efficiently capturing temporal trends in crime data through a thorough data pretreatment pipeline, visualization, and assessment process. The importance of employing neural networks for time series crime predicting tasks was further supported by statistical tests that showed the variations in performance were significant. These results have applications in proactive policing, resource allocation, and public safety planning. Law enforcement organizations can use predictive insights from crime statistics to improve community safety measures, anticipate high-risk times, and make data-driven decisions. Such models can be reliably used in real-world decision-making settings thanks to the combination of statistical validation, interpretability, and visual analytics.

A. Future Works

Future work can expand and refine this study in several directions:

- **Spatiotemporal Modeling:** By combining temporal and spatial data, models that forecast not only the occurrence of crimes but also their likelihood can be generated.
- **Real-Time Forecasting Pipelines:** Developing real-time crime prediction dashboards could enable law enforcement to respond quickly to shifting patterns.
- **Explainable AI Integration:** Automated forecasting tools may become more transparent and reliable if interpretable AI techniques (like SHAP or LIME) are adopted.
- **Cross-City or Global Application:** Validating the method's scalability and adaptability to diverse crime dynamics would need testing it in other cities or counties.

An intended component of this project was hotspot mapping of crime incidents. However, due to challenges in implementation within the timeframe, this feature was excluded from the final project. Future work will explore incorporating spatial structuring or geospatial analysis to identify crime hotspots.

VI. References

[1] T. Wang, C. Rudin, D. Wagner, and R. Sevieri, "Learning to detect patterns of crime," *Machine Learning*, vol. 93, no. 1, pp. 515–554, 2013.

[2] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long short-term memory," *PloS One*, vol. 12, no. 7, p. e0180944, 2017.

[3] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003, pp. 2–11.

[4] Y. Zhang, L. Deng, L. Liu, Y. He, and J. Wu, "Spatio-temporal crime prediction using graph convolutional networks," *IEEE Access*, vol. 8, pp. 181529–181538, 2020.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] G. Box, G. Jenkins, G. Reinsel, and G. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Wiley, 2015.

[7] Y. Yu, H. Si, W. Liu, and J. Gao, "Crime prediction through urban sensing data based on machine learning," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7767–7775, Oct. 2019.

[8] J. Chainey and L. Tompson, "Engaging communities in crime prevention: Lessons from an evaluation of Neighborhood Watch in London," *Crime Prevention and Community Safety*, vol. 14, no. 3, pp. 191–210, 2012.

[9] L. Rokach and O. Maimon, "Decision trees," in *Data Mining and Knowledge Discovery Handbook*, Springer, 2010, pp. 165–192.

[10] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proc. 23rd Int. Conf. Machine Learning (ICML)*, 2006, pp. 161–168.