GEO 1762 : Real-time Emergency Prediction Decision Support Tool
City of Barrie

Author: Athuri Vivekanantharasa (200583830)

Ramya Kappagantu (200578811)

Raju Thapa (200584151)

Date: 2024/08/09

Table of Contents

1. Introduction

1.1 About the Client

City of Barrie is a vibrant, progressive and growing government municipality which is making continuous efforts to provide better facilities to the local community. They undertake various strategic projects and initiatives to support Barrie's growth, environment, transportation and many more, some of which are Bayview Drive Transportation improvements, parking pay station device updates to meet Payment Card Industry(PCI) security standards.

1.2 Background of the topic

Last year when wildfires from Quebec hit Barrie, The Department of Barrie Fire and Emergency Services couldn't differentiate between emergencies that were caused due to an actual fire accident and that of a wildfire which is causing campfire smoke. People were constantly contacting 911 whenever they saw a wildfire and they couldn't find a way to prioritise those where there was actually fire. This called for a solution to predict the severity of fire and the type of emergency so that necessary actions can be taken by the department.

1.3 Describe the Client's need

The Department of Barrie Fire and Emergency Services would like to develop a real-time tool which can actually predict the emergency situations based on the existing data sources or any additional data sources if needed. The tool should predict the severity of the fire and type of the emergency for example, if it is a wildfire or actually a fire accident so that they can have the fire apparatus available to take necessary and immediate action.

1.4 Project Rationale

Why the Project is significant, strategic or relevant

- Predicting fire emergencies will help the fire department to strategically develop firefighting equipment to high-risk areas. This ensures that the resources are used efficiently and effectively, minimising the response time and maximising the effectiveness.
- Fires can impact the environment causing loss of biodiversity, habitat destruction, air and water pollution and contribution to climate change through the release of greenhouse gases. If we are able to predict the fire beforehand, it helps in better assessment and management of these environmental impacts, helping to protect ecosystems and wildlife.
- Fires can cause risk to public safety and health due to smoke inhalation, respiratory issues and injuries. Predicting fire emergencies will help the authorities to implement measures to protect public health such as warning the public well in advance, issuing health advisories, providing shelters and coordinating medical assistance.

1.5 Objective

To predict the type and severity of the fire emergency using data collected from current and historical weather sources, CAD and RMS current and historical incident data and current road conditions including closures.

Scope of Project and Out of Scope

1. **In Scope:**
    a. The project predicts where the next emergency area will be so that the department can take necessary action and resources available.
    b. The project predicts the type of fire emergency and severity of the fire emergency.
2. **Out of Scope:**
    a. The project does not find an optimum route to the fire emergency based on the specific conditions determined.

1.6 Functional Requirements

Emergency Prediction:

1. **Prediction of Emergency Area and Category:**
    - The system should predict the next emergency area (DAUID) and the category of the emergency for the current date.
    - Users should be able to select future dates from the UI.
    - Upon selecting a date and clicking the search button, the system should update the map to display predictions for the chosen date.
    - The map should also display the prediction for the current date in real-time, updating every 15 minutes.
2. **Real-Time Prediction Updates:**
    - Predictions for the current date should update every 15 minutes.
    - The updated prediction should be displayed in a card on the UI, showing the latest emergency area, category, and prediction date.

Map Visualization:

1. **Dissemination Area Polygons:**
    - The map should display each dissemination area as a polygon.
    - Each polygon should have color coding based on severity levels to indicate different levels of emergency severity.
2. **Severity Levels:**
    - The map should visualize severity levels using different colors, with a clear legend explaining what each color represents.

User Management:

1. **Admin User Functionality:**
    - Only admin users should have the capability to add more users to the system.
    - The system should include a UI for login and logout, with appropriate user-friendly alerts and notifications.
2. **User Authentication:**
    - The system should manage user authentication and ensure secure login and logout functionality.
    - Alerts should be implemented for successful or unsuccessful login attempts, as well as user registration.

<u>AI Models Implementation:</u>

1. **Model Selection and Training:**
    - The system should support the implementation of at least three different AI models for emergency prediction.
    - From the models implemented, two should be selected for the final deployment based on performance metrics.
    - Documentation of the training process, including details of datasets used, data cleaning, and data aggregation, should be maintained.

<u>Severity Index Calculation:</u>

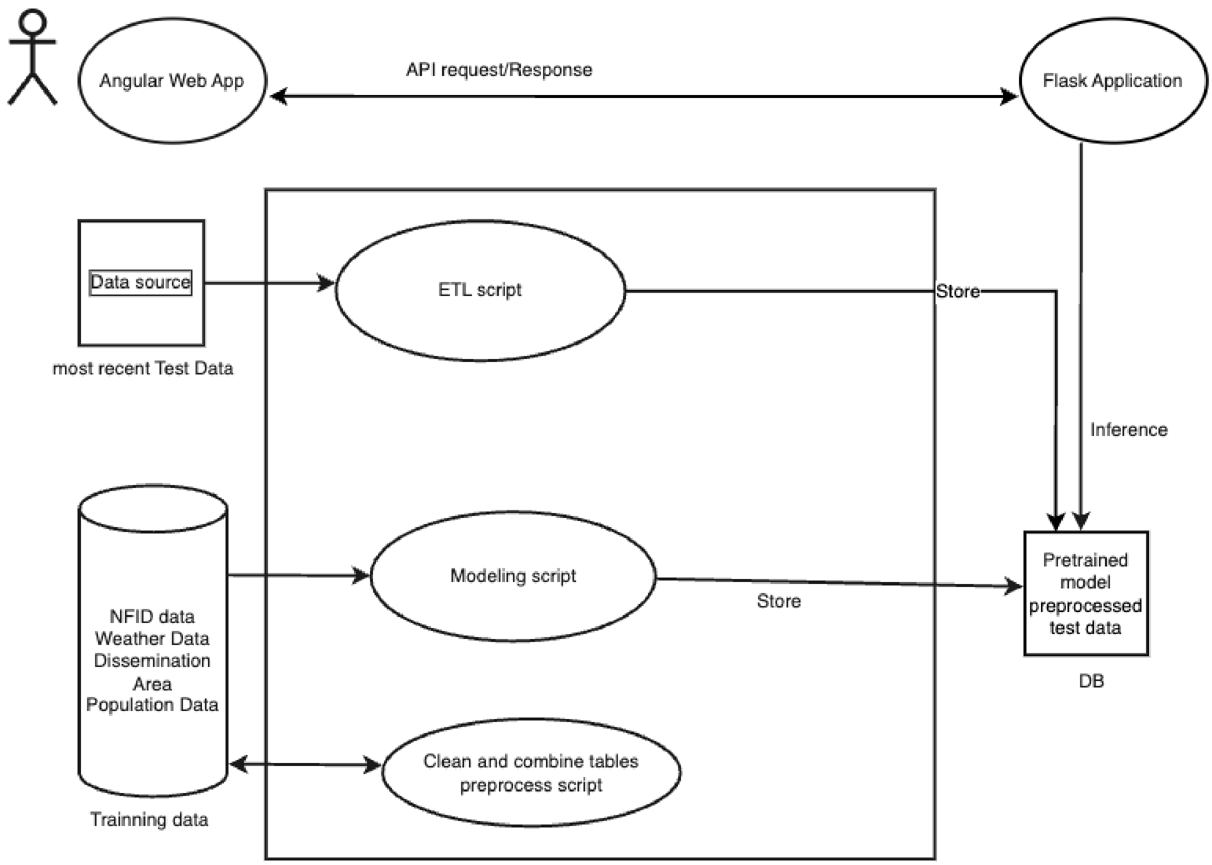
1. **Severity Index Documentation:**
    - The system should calculate the severity index, considering various parameters that influence the severity level of emergencies.
    - Documentation should be provided on how severity levels are predicted and the parameters considered in the calculation.

2. System Overview

System Description

The system is a real-time emergency prediction and visualization tool that integrates machine learning models, interactive map displays, and robust user management. It predicts emergency locations and categories, visualizes severity levels on a map, and allows admin-controlled user management. Key components include an AI-driven prediction engine, severity index calculations, and real-time data processing, all designed to assist in proactive emergency response planning and management.

Architecture Diagram



3. Data Description

3.1 Fire Alarm Sprinkler Data

The dataset in focus contains comprehensive information regarding fire alarm and sprinkler systems installed in various buildings. It includes key attributes that detail the presence, status, and specifics of these fire safety systems.

Data Preprocessing:

Added new columns longitude and latitude : The geopy library was used to convert building addresses into latitude and longitude, involving external geocoding services essential for spatial analysis.

The dataset underwent rigorous cleaning and transformation, including filling missing values, standardising dates, and geocoding location data. This multi-source approach ensures the dataset is comprehensive, detailed, and well-structured for analysis, providing a robust foundation for trend analysis, spatial analysis, and predictive modelling of fire safety systems in buildings. **However, we were unable to join this data with the CAD & RMS dataset because the details pertain only to commercial buildings, and there is no key available to link it with the incidents recorded in the CAD & RMS dataset.**

3.2 CAD & RMS Data

The dataset contains 40815 rows, where each row represents a fire incident. The dataset contains 20 individual variables (features) which may serve as predictors of the next fire incident. The dataset has 2 sheets which includes the following features:

Sheet 1 - Incident Data

- unique_Id : This variable uniquely identifies each fire incident. It can be ultimately considered as a primary key in the database.
- inci_no : This variable uniquely identifies each fire incident recorded. It can be used to track, manage, and reference specific incidents from the moment they are reported until they are resolved and recorded in the system.
- inci_type : It can be used to categorise the type of fire incident based on which we can decide the impact of the fire incident. This is a nominal variable ranging from 1 to 913.
- alm_type : It can be used to define the type of alarm that was used during the fire incident. This is a nominal variable ranging from 1 to 11 and also contains 4 missing values.
- e911_used : This variable can define whether 911 has been contacted during the fire incident. It contains a lot of inconsistent data and is out of scope for our project.
- Property_Loss_Value : This variable quantifies the amount of external property loss (buildings, vehicles, fences and gates, irrigation systems) that has happened due to the fire incident. It is measured in Canadian Dollars(CAD).

- Content_Loss_Value : This variable quantifies the amount of damage or destruction of the personal belongings, furnishings and other movable items within a property as a result of the fire incident. This type of loss is different from Property_Loss_Value (structural damage that pertains to the buildings itself). It is also measured in Canadian Dollars (CAD).
- Property_Value : This variable defines the total actual value of the structural property like building, vehicle, irrigation systems where the fire incident happened. It is also measured in Canadian Dollars (CAD).
- Content_Value : This value defines the total actual value of the movable items within the property before the fire incident happened. It is also measured in Canadian Dollars (CAD).
- Civilian_Fatal : This refers to the number of deaths of non-firefighters as a result of the fire incident. This variable can be used to identify fatalities among the general public who got affected by the fire incident. We have values 0 and 1 for this variable in our data.
- Civilian_Injuries : This refers to the number of people who had injuries due to the fire incident. They can be occupants of the affected property, bystanders or visitors. We have the values 0,1 and 2 recorded in our data.
- prop_use : This variable is used to categorise the type of property involved in the fire incident, which can be used to define the property and content damage. This is a nominal variable with values ranging from 101 to 999.
- alm_dttm : This variable defines the datetime when 911 had received the call.
- disp_dttm : This variable defines the datetime when the fire trucks were dispatched.
- arv_dttm : This variable defines the datetime when the first fire truck arrived at the location of the incident.
- ctrl_dttm : This variable defines the datetime when the fire incident was controlled. This is considered as a transition time and is out of scope for the project, since it doesn't provide any useful predictions.
- clr_dttm : This variable defines the time when the last fire truck has cleared from the incident. This time is usually considered as the end time of the incident.
- DAUID : This variable uniquely defines each dissemination area in Barrie. A dissemination area is a small area composed of one or more neighbouring dissemination blocks and is the smallest standard geographic area for which all the census data are disseminated. (Reference: Statcan)
- NEAR_X : This variable defines the longitude of the location where the fire incident happened. The values range between -79.74 and -79.58.
- NEAR_Y : This variable defines the latitude of the location where the fire incident happened. The values range between 44.29 to 44.42.

Sheet 2 - Incident and Property Usage Category codes with description

1. category : This variable defines the category that differentiates between types of incidents (INCI TYPE) and property use (PROP USE)

2. grp : This variable represents a subgroup within the category value, which provides further breakdown. The values range from A to J for INCI TYPE and 100 to 940 for PROP USE
3. code : This variable is a unique identifier that is assigned to each specific type of fire incident (INCI TYPE) and property use (PROP USE). This is a nominal variable where the values range from 1 to 999.
4. descript : This variable provides a brief description of the specific type of the incident or property use represented by the code. A few examples include fire, pot on stove(no fire), other false fire calls.

Data Preprocessing

A few preprocessing techniques were applied on CAD and RMS data to extract useful features that can be used for machine learning modelling to predict the next fire emergency.

- Drop unwanted columns: Since the data contains a lot of columns like e911_used, alm_type, ctrl_dttm that don't contribute in predicting the next fire emergency, these columns were dropped from the data using pandas drop() function. OID_ has also been removed from the data since we already have inci_no which can uniquely identify each fire incident.
- Missing Values: After dropping the unwanted columns, the data was analysed to check if there are any missing values using pandas isnull() function. There were 408 missing values in arv_dttm, 4 missing values in clr_dttm and 245 missing values in DAUID. The missing values in arv_dttm(Arrival Datetime) could be because the incident never occurred or it was resolved over the phone. On further analysis, the 4 missing values in clr_dttm occurred when the arv_dttm was missing. This justifies the fact that, since the fire department didn't arrive at the location, they don't need to clear the location. 245 missing values in DAUID could be because the locations(found by latitude and longitude) may be outside of Barrie for which dissemination areas weren't recorded or they could be on any highway or unidentified roads which couldn't be mapped to any dissemination area. In order to verify those missing values, we have used the shape file of dissemination areas of Barrie from Statcan and mapped all the latitude and longitude values with their corresponding DAUID and used it in our aggregated data for model training. Section 4 explains more about how DAUID was calculated for each latitude and longitude.

3.3 Weather Data

After obtaining the CAD and RMS data, FireAlarmSprinkler data from the client, an initial time-series forecasting has been done using Prophet, LSTM (Long Short-Term Memory) and VAR (Vector Autoregression) models. The models weren't able to provide accurate time to time predictions because of the limited data across different timeframes and lack of other factors that can help in causing fire incidents. In order to improve the model accuracy and provide accurate predictions, Barrie weather data was extracted from meteoblue. Barrie city has been selected and weather data between 2020 and 2022 has been extracted on a daily basis. The data contains columns like timestamp, Temperature, Relative Humidity, Snowfall Amount, Wind Speed and Wind Direction.

3.4 Census Data

Datasource link: Statistics Canada. Table 98-10-0129-01 Marital status, age group and gender: Canada, provinces and territories, census divisions, census subdivisions and dissemination areas.

Frequency: Occasional Table: 98-10-0129-01
Release Date: July 13, 2022
Geography: Canada, Province or Territory, Census Division, Census Subdivision, Dissemination Area
Universe: Population 15 years of age and older, 2021 Census — 100%

Data Description: This dataset provides detailed information on marital status categorized by age group and gender for various geographic regions in Canada, including provinces, territories, census divisions, census subdivisions, and dissemination areas. The data is sourced from the 2021 Census, covering the entire population aged 15 years and older

Data preprocessing

Column Removal:

- Removed the following columns: `'VECTOR'`, `'COORDINATE'`, `'SCALAR_ID'`, `'SCALAR_FACTOR'`, `'UOM'`, `'UOM_ID'`, `'STATUS'`, `'SYMBOL'`, and `'TERMINATED'` as they are deemed unnecessary for the analysis.

Gender Column Removal:

- Removed the `'Gender (3)'` column as it contained only a single unique value, `'Total - Gender'`, which does not provide detailed gender information and simplifies the analysis.

Rows Removal:

- Excluded rows corresponding to Canada and Barrie location data to focus solely on population data by Dissemination Area (DAUID), as the Canada and Barrie rows represent aggregated data.

Severity Indicator Identification:

- Identified only one Severity Indicator: Percentage of Population 65 Years and Over.
- Grouped data by Dissemination Area (DAUID) and filtered for rows corresponding to the age group above 65 to focus on this specific indicator.

4. Finding DAUID by location

Finding Dissemination Area Code using Latitude and Longitude of the Location

First, we need the shapefiles for Barrie to get the dissemination area borders. We downloaded the shapefile zip from the following link: Statistics Canada Shapefiles

We extracted latitude and longitude for each incident using CAD & RMS data and determining the DAUID that corresponds to that location. We then added a new column for DAUID in the CAD & RMS data and removed the previous DAUID column. We used this updated DAUID for our analysis and implementation.

Note: According to the Statistics Canada population data by dissemination area, there are only 243 DAs in their records. However, the CAD&RMS dataset contains 246 unique DAs. According to the 2021 Statistics Canada data, the following 5 dissemination areas are not included in Barrie but are present in the CAD records. Additionally, there are 2 DAs that are in the Statistics Canada records but have no data available in the CAD dataset. For our project, we are considering a total of 248 DAs (243 + 5).

DAUIDs only in CAD dataframe:

- ***35430639***
- ***35430715***
- ***35430961***
- ***35431133***
- ***35431377***

DAUIDs only in Population dataframe:

- ***35430682***
- ***35430710***

5. Severity Index Calculation Methodology

Predicting the Severity of Fire Incidents Using 2021 Census Data, National Fire Information Database (NFID) Data, and CAD & RMS Data

5.1. Severity Indicators

- Fire Incident Rate - Completed
- Percentage of Non-Working Smoke Alarms in Fires - Data Not Available
- Civilian Injuries Rate per Incident - Completed
- Casualty Rate per Incident - Completed
- Property Damage Cost per Fire Incident - Completed
- Response Time - Completed

5.2. Population Risk Factors

- Percentage of Population Under 6 Years
- Percentage of Population 65 Years and Over - Completed
- Percentage of Lone-Parent Families
- Unemployment Rate
- Percentage of Movers

<u>Fire Incident Rate</u>

Definition: The number of fire incidents occurring within a specific dissemination area. **Purpose**: This indicator helps understand the frequency of fire incidents in different areas. High fire incident rates could indicate a need for better fire prevention measures and public education. **Calculation**: Total number of fire incidents in a given area.

<u>Percentage of Non-Working Smoke Alarms in Fires</u>

Definition: The proportion of fire incidents where smoke alarms were not working out of the total fire incidents.
Purpose: This indicator highlights the effectiveness of smoke alarm maintenance and installation in different areas. A high percentage of non-working smoke alarms could suggest a need for better public awareness campaigns or stricter regulations on smoke alarm installations. **Calculation**: (Number of fire incidents with non-working smoke alarms / Total number of fire incidents) * 100.

<u>Civilian Injuries Rate per Incident</u>

Definition: The average number of civilian injuries per fire incident. **Purpose**: This indicator measures the impact of fire incidents on human health and safety. High rates of civilian injuries per incident can indicate the severity of fires and the effectiveness of emergency

response services.
Calculation: Total number of civilian injuries / Total number of fire incidents.

Casualty Rate per Incident

Definition: The average number of civilian fatalities per fire incident.
Purpose: This indicator provides insight into the deadliness of fire incidents in different areas. Higher casualty rates may indicate more severe fire incidents or delayed emergency response times.
Calculation: Total number of civilian fatalities / Total number of fire incidents.

Property Damage Cost per Fire Incident

Definition: The average monetary value of property damage per fire incident.
Purpose: This indicator assesses the financial impact of fire incidents. High property damage costs per incident can indicate more severe fires, possibly due to the type of buildings affected or the effectiveness of fire control measures.
Calculation: Total property damage cost / Total number of fire incidents.

Average Response Time

Definition: The average time taken for emergency services to respond to fire incidents.
Purpose: This indicator evaluates the efficiency of emergency response services. Shorter response times generally lead to less severe outcomes in fire incidents.
Calculation: Sum of response times / Total number of fire incidents.

5.3 Methodology

Steps for Calculating Fire Severity Index

1. Standardization

- Convert all population characteristics and severity data into rates or percentages.
- Standardize these measures into z-scores to make them comparable across different DAs.

2. Identify High-Severity DAs

- Use z-scores to identify DAs in the top 10% for each indicator (both population risk factors and severity indicators).
- Convert these into categorical variables (1 = high severity risk, 0 = not high severity risk).

3. Create Severity Index

- Sum the categorical variables for each DA to create a composite severity score, combining both population risk factors and fire severity indicators.

4. Categorization

- Categorize DAs based on their composite severity score:
    - **5**: Very High Severity (6-8 indicators in top 10%)
    - **4**: High Severity (4-5 indicators in top 10%)
    - **3**: Moderate Severity (3 indicators in top 10%)
    - **2**: Low Severity (1-2 indicators in top 10%)
    - **1**: Very Low Severity (0 indicators in top 10%)

6. Modelling Approach

Three models were initially selected for consideration: LSTM, Prophet, and VAR. Ultimately, LSTM and Prophet were chosen for final implementation. However, further improvements are necessary to determine the best model between LSTM and Prophet.

In our implementation of Prophet, we have treated DAUID and category as numerical features. Although Prophet is primarily designed for time series forecasting and is best suited for continuous data, we utilised it to identify patterns and trends in historical data. By encoding these inherently categorical features numerically, Prophet allows us to uncover temporal patterns. For future enhancements, we suggest combining Prophet with a dedicated classification model to better handle the categorical nature of the data and improve predictive accuracy.

6.1 Long Short-Term Memory Model((LSTM))

The aim of this project is to predict the next emergency area (DAUID) and type of emergency using the CADRMS dataset. Given the temporal nature of emergency events, a Long Short-Term Memory (LSTM) model was chosen for its effectiveness in handling time series data and its ability to capture long-term dependencies. Initially, the model was trained solely on CAD and RMS data. However, the performance was suboptimal due to its reliance on historical data till December 31, 2023, for current and future predictions. Therefore, an aggregated dataset including weather data was created and used to improve the model's performance.

Two versions of the model were developed: one that returned a single prediction at a time and another that provided multiple predictions (specifically, 10 predictions) for each time step with the aggregated dataset.

Data Preparation:

- **Dataset:** The dataset comprised various features, including incident details, geographic coordinates, weather conditions, and timestamps. The data was aggregated and resampled at 15-minute intervals to capture temporal dependencies.
- **Categorical Encoding:** Categorical features such as DAUID and incident categories were encoded using Label Encoding to transform them into numerical formats suitable for model input.
- **Scaling:** Numerical features were standardised using StandardScaler to ensure that all features contributed equally to the model's learning process.

Model Architecture:

- **LSTM Layers:** Both models utilised Bidirectional LSTM layers to capture temporal patterns in the data from both forward and backward directions. BatchNormalization and Dropout layers were applied to prevent overfitting.
- **Dense Layers:** The LSTM outputs were connected to dense layers, which then fed into separate output layers for predicting DAUID and incident category.
- **Activation:** Softmax activation was used in the output layers to manage the multi-class classification problem inherent in both DAUID and category predictions.

Model Comparison and Aggregated Dataset:

- **Initial Model:** The initial LSTM model was trained solely on the CAD and RMS data. The performance was less than expected because it relied heavily on historical data from December 31, 2023, for predicting current and future events.
- **Aggregated Dataset:** To enhance the model's predictive capabilities, an aggregated dataset that included weather data was created. This dataset combined CAD and RMS data with weather parameters such as wind speed, temperature, and precipitation.
- **Extended Predictions:** To improve performance, the model was configured to make predictions for 1, 4, 8, 10, and 24-hour intervals together, providing a more comprehensive prediction range.
- **Model Comparison:**
    - **CAD and RMS-Only Model:** This model was trained solely on the CADRMS dataset.
    - **Aggregated Dataset Model:** This model was trained on the aggregated dataset, which included both CAD and RMS and weather data.
    - **Results Comparison:** The model trained on the aggregated dataset performed better due to the additional context provided by weather parameters, which can influence the frequency and type of emergencies.

Single Output Model:

- **Functionality:** This model was configured to generate a single prediction for the next time step. It is optimised for scenarios requiring immediate and accurate predictions.
- **Performance:**
    - **DAUID Prediction:** Achieved an accuracy of 48.36%.
    - **Incident Category Prediction:** Achieved a high accuracy of 92.81%.
- **Use Case:** Ideal for real-time applications where the focus is on predicting the immediate next event with high confidence.

Multiple Output Model:

- **Functionality:** This version of the LSTM model was designed to return 10 predictions at a time. This setup is valuable for scenarios requiring predictions over a broader time horizon, such as predicting several future events at once.
- **Performance:**
    - **Prediction Accuracy:** The model's accuracy for DAUID and category predictions generally declined as the number of predicted time steps increased. The first prediction

(1st output) was the most accurate, with accuracy decreasing for subsequent predictions (e.g., 4th or 10th output).
    - **Comparison Across Outputs:** The accuracy and performance of the first few predictions were higher, making them more reliable for immediate use, while later predictions provided a broader, albeit less accurate, forecast.
- **Use Case:** Suitable for planning over extended periods, where having a sequence of predictions is more critical than the accuracy of any single prediction.

Data Recency and Multi-Iteration Predictions:

- **Prediction Horizon:** since most recent data is from December 31, 2023, predicting current (August 2024) or future events requires multiple iterations. Each prediction corresponds to a 15-minute time step, necessitating numerous iterations to bridge the time gap.
- **Impact on Performance:** The need for iterative predictions affects the model's performance, as each subsequent prediction is based on the previous one, leading to potential error accumulation.
- **Choosing 10 Predictions:** To optimise performance and manage the computational load, the model was designed to return 10 predictions at a time. This approach reduces the number of iterations required and helps maintain efficiency.

Future Considerations:

- **Updating Test Data:** As more recent data becomes available (beyond December 2023), it is essential to update the test data to include the latest 16 time steps from CAD RMS, combined with corresponding weather data. This update will enhance the model's accuracy by providing a more current and relevant dataset.
- **Data Aggregation:** Creating a new aggregated dataset with the most recent CAD RMS and weather data is crucial for maintaining model performance. This aggregation will ensure that the model's predictions are grounded in the most up-to-date information.

Reason for Choosing LSTM:

The LSTM model was chosen for this task for several reasons:

- **Ability to Handle Sequential Data:** LSTM can effectively model time series data, capturing the temporal dependencies essential for predicting future emergencies.
- **Long-Term Dependency Capture:** LSTM's architecture allows it to remember information over long sequences, which is crucial for accurately predicting emergencies based on historical data.
- **Proven Performance:** LSTMs have been widely used and proven effective in various time series prediction tasks, making them a reliable choice for this project.
- **Improvement with Aggregated Data:** The inclusion of weather data enhanced the model's performance, demonstrating LSTM's capability to integrate multiple data sources for more accurate predictions.

6.2 PROPHET Model

To forecast the next emergency area (DAUID) and the type of emergency using the CADRMS dataset, the Prophet model was utilised due to its strengths in time series forecasting, particularly in capturing trends, seasonality, and holiday effects. By encoding categorical features such as DAUID and incident types into numerical values, the model was able to reveal temporal patterns in the data. Although this approach yielded valuable insights, it also highlighted the limitations of Prophet when applied to categorical data. To enhance predictive accuracy, future work should consider integrating Prophet with a classification model better suited to handling categorical variables, thus leveraging the strengths of both approaches.

1. How Prophet Works

Prophet is an open-source time series forecasting tool developed by Facebook, designed to handle datasets with strong seasonal effects and historical data spanning multiple seasons. Prophet uses an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.

Trend Component: Prophet models the trend in the data using either a linear or logistic growth curve. The linear trend assumes that the time series will grow or shrink consistently over time, while the logistic trend allows for a saturation point, which is useful when forecasting growth that is constrained.

Seasonality Component: Prophet automatically detects and models yearly, weekly, and daily seasonality. Users can also add custom seasonalities if their data exhibits more complex patterns.

Holiday Effects: Prophet can incorporate the effect of holidays on the time series by allowing users to specify a list of holidays that might influence the time series.

Changepoints: One of Prophet's strengths is its ability to automatically detect changepoints i.e. points in the data where the trend significantly shifts. This makes it particularly useful for time series that undergo sudden changes.

Uncertainty Intervals: Prophet also provides uncertainty intervals, which are calculated using Bayesian methods. These intervals help quantify the confidence in the model’s predictions.

2. Data Preprocessing

Data preprocessing is a crucial step in ensuring the quality and suitability of data for use in the Prophet model.

Time Data Cleaning: The time data was cleaned to remove inconsistencies, such as unnecessary decimal points in seconds and hours that exceed 24. Prophet requires that the ds (DateTime) column be in a consistent format, so these steps are essential.

Datetime Conversion: All time-related columns (e.g., Received Datetime, Dispatched Datetime) were converted into a consistent date time format. This ensures that Prophet can accurately process and model the time series data.

Categorical Encoding: Categorical features like incident_category description and DAUID were encoded into numerical values using Label Encoding. Prophet expects the y column to be numerical, so this conversion is necessary for the model to process the data.

Time Series Structuring: The data was structured into two separate data frames: one for predicting DAUID and another for incident category description. This structuring is necessary because each data frame is used to train separate Prophet models, one for each prediction task.

3. **Model Architecture and Hyperparameters**

Prophet's architecture revolves around its ability to model time series data using trend, seasonality, and holiday effects.

Trend Component: Configurable to be linear or logistic, with changepoints where the trend is allowed to change.

Seasonality Component: Yearly, weekly, and daily seasonalities are automatically modelled, with options to add custom seasonal patterns.

Hyperparameters:

Changepoint prior scale: Controls the flexibility of the trend component, determining how sensitive the model is to detecting changepoints.

Seasonality prior scale: Governs the flexibility of seasonal components, controlling how much the seasonality patterns can change over time.

Holidays prior scale: Manages the effect of holidays on the forecast, allowing the model to adjust the significance of holiday events in the time series.

4. **Handling Categorical Features in Prophet**

In our implementation of Prophet, DAUID and incident category description were treated as numerical features by encoding them. While Prophet is primarily designed for continuous time series forecasting, this approach allows us to leverage Prophet's capabilities to identify temporal patterns and trends in historical data, even though these features are inherently categorical.

By encoding categorical features numerically, Prophet can still uncover underlying patterns that occur over time, such as seasonal trends or the impact of specific events. However, it is important to recognize that the Prophet is not inherently designed for classification tasks. Therefore, while this method provides valuable insights, it may not fully capture the complexity of categorical data.

For future improvements, it is recommended to combine this approach with a dedicated classification model. A classification model, specifically designed for handling categorical data, would be better suited to predicting categories like incident_category_description and DAUID. Integrating such a model with Prophet could enhance predictive accuracy by allowing each model to play to its strengths.

5. Training the Model

The model was trained using an 80-20 split of the data, where 80% of the data was used for training and 20% was used for testing.

Model Fitting: The Prophet model was fitted using the training data. During the fitting process, the model identified the best-fit parameters for the trend, seasonality, and holiday effects, based on the historical data provided.

Hyperparameter Tuning: A grid search was used to explore different combinations of hyperparameters (e.g., changepoint prior scale, seasonality prior scale, holidays prior scale). The best parameters were selected based on the model's performance on the test set, measured by Mean Squared Error (MSE).

6. Metrics for Evaluation

The performance of the Prophet model was evaluated using several key metrics:

Mean Squared Error (MSE): This measures the average squared difference between the predicted and actual values. It heavily penalizes larger errors, making it a robust metric for evaluating prediction accuracy.

Mean Absolute Error (MAE): This metric measures the average absolute difference between predicted and actual values, providing a straightforward interpretation of model accuracy.

R-squared (R^2): R^2 indicates how well the model explains the variance in the data. Values closer to 1 indicate better performance, while negative values suggest that the model is performing worse than a simple mean-based model.

Directional Accuracy: This metric assesses whether the model correctly predicts the direction of change (increase or decrease) in the target variable. It is particularly useful for understanding the model's effectiveness in capturing trends.

7. Fine-Tuning the Model

Fine-tuning involved several steps to optimize the model's performance:

Hyperparameter Tuning: Techniques like grid search were employed to explore different combinations of key hyperparameters. The objective was to identify the combination that resulted in the lowest MSE on the test set.

Model Selection: The model with the best-performing hyperparameters (based on MSE) was selected as the final model. This model was then retrained on the full training dataset to ensure that it captured the most information possible before making predictions.

Regularization: Regularization parameters like changepoint prior scale and seasonality prior scale were adjusted to prevent overfitting, ensuring that the model generalized well to unseen data.

8. Reason for Choosing Prophet

Prophet was chosen for this task due to its:

Flexibility and Robustness: Prophet is well-suited for time series data that exhibits strong seasonality, trends, and holidays, making it ideal for the CADRMS dataset, which likely has such characteristics.

Ease of Use: Prophet is designed to work out of the box with minimal configuration, making it accessible even to those without deep expertise in time series forecasting.

Automatic Handling of Missing Data and Outliers: Prophet can automatically handle missing data and outliers, reducing the need for extensive data cleaning and making the model more resilient to irregularities in the data.

Ability to Incorporate Domain Knowledge: The model allows users to add custom seasonalities, holidays, and other events, making it possible to incorporate domain knowledge into the forecasting process, which can significantly improve accuracy.

Utilizing the Prophet model for predicting emergency areas (DAUID) and incident types allowed for the identification of important temporal patterns within the CADRMS dataset. By numerically encoding categorical features, Prophet effectively modelled trends and seasonal variations, offering valuable insights into the underlying data structure. However, the model's application to categorical data also highlighted its limitations, suggesting that a combined approach with a dedicated classification model could further enhance predictive accuracy. This strategy would leverage Prophet's strengths in time series analysis while addressing the complexities of categorical data, resulting in more robust and accurate forecasts.

6.3 VAR Model

The Vector Autoregression(VAR) model is a multivariate forecasting algorithm that is used when two or more time series influence each other. It is modelled as a system of equations with one equation per time series variable.

Understanding VAR according to our problem:

A univariate time series is a series that contains only a single variable depending on time, whereas multivariate time series have more than one variable that depends on time. Each variable depends not only on its past values but also has some dependency on other variables. In our problem, we are trying to predict the type of next fire incident (incident_type_description) and where it will happen (dissemination area). Since, we have two variables depending on time, our data comes under multivariate time series.

Steps followed to build a VAR model:

- Data Preprocessing and EDA
- Stationarity Test
- Granger Causality Test for correlation
- Train Test Split
- Grid Search for order p
- VAR model training with order p
- Forecast on test data

1. **Data Preprocessing and EDA:**
   a. Label encoding the incident_category_description: Since our target variables are incident_category_description and DAUID, the target variables have been converted from categorical columns to numerical columns using label encoding in python so that they can be fitted into a machine learning model for further predictions.
   b. Resampling the data: In our project, we are targeting to provide real-time predictions for every 15 minutes. In order to provide the model with as much historical data as possible, the data has been resampled with a frequency of 15 minutes to ensure consistency for the VAR model and maintain continuity in the data. Resampling can also be used to smooth out noise in the data and highlight the underlying trend. This will ultimately help in improving the model's accuracy by focusing on the significant patterns rather than the noise.
   c. Removing null values formed while resampling: In case, there were no predictions found in a particular time period while sampling, we will get a null value. This can again lead to incorrect predictions if not removed. Hence, all those rows with missing values/null values are eventually dropped before model training.

2. **Stationarity Test:**

   Stationarity is a statistical property where each time series has constant mean and variance over time. One of the common methods to perform a stationarity test is **Augmented Dickey-Fuller test (ADF Test)**.

   In the ADF test, the null hypothesis is considered time-series to be non-stationary. We need to choose a significance level p (usually 0.05) based on which we can know if a time-series is stationary. If p-value is less than the significance level then we reject the null hypothesis and consider the time series to be stationary. We will be passing each and every time series (variables) to test whether they are stationary.
   What if a time-series is not stationary:
   If a time series is not stationary and is used to train a VAR model, it can result in spurious correlations, which makes it appear as if there are relationships between variables when, in fact, there could be no relationship. This can ultimately lead to incorrect conclusions about the dynamics of the time series. The model parameters may become unstable. This can lead to poor out-of-sample forecasting performance.

   Address non-stationarity:

   a. Apply differencing to the time series data to make it stationary. This involves subtracting the previous observation from the current observation.
   b. Apply transformations such as logarithms or square roots to stabilise the variance.

3. **Granger Causality Test for correlation:**

   The Granger Causality Test is a statistical hypothesis test for determining whether one time series offers useful information to the other time series. We need to choose a significance level p (usually 0.05) based on which we can know if both the time series used are correlated. If p-value is less than the significance level then we reject the null hypothesis and consider that one time series to granger causes the other. We will be passing each and every time series (variables) to test whether they are stationary.

   Since we are going to predict the incident_category_description and DAUID, we are going to check the other time series alongside incident_category_description and DAUID and choose those time series which granger cause incident_category_description and DAUID.

4. **Train Test Split:**

   The data we have is available from January 2020 to December 2023, of which the data from January 2020 to December 2022 was used as training data and rest of the data as test data.

5. **Grid Search for order p:**

Selecting the order in the VAR model refers to determining the appropriate number of lags i.e past values to include in the model. The correct order is crucial as it impacts the model's accuracy and the validity of inferences drawn from it.

Factors used to select the order of optimal lag:

- **Akaike Information Criteria (AIC):** Measure of relative quality of statistical models for a given set of data. The model with lowest value of AIC is chosen as best.
- **Bayesian Information Criteria (BIC):** It provides a stronger penalty for model complexity and tends to select simpler models when sample sizes are large. The model with lowest BIC value is chosen as best.
- **Hannan-Quinn Information Criterion (HQIC):** Similar to AIC but imposes a stronger penalty for the number of parameters. The model lowest value of HQIC is chosen as best.

The lag number for which the AIC, BIC and HQIC is the lowest is chosen as the p-value.

6. **VAR model training with order p:** The model is then fitted with the optimum p value obtained in the previous step using model.fit() method.
7. **Forecast on test data:**

   model.forecast() method is used to forecast the future data using the previous records. The number of previous records is equal to the p-value which has been obtained previously. The step parameter in the forecast method specifies the number of time periods ahead for which you want to generate the forecast.

8. **Performance Evaluation:**

   Once we obtain the predicted incident_category_description and DAUID using model.forecast() method, we will be comparing it with the test data and calculate the mean of all those records where the predicted value is equal to the original value to get the accuracy of the model.

9. **Performance Improvement:**
   a. **Using one-hot encoding for target variables:** Label encoding is usually preferred when there is a natural ordering between the variables. For example, 'first', 'second' and 'third'. In our project, the target variables 'incident_category_description' and 'DAUID' don't have any natural ordering between them and nothing regarding their ordering was explicitly mentioned. In fact, if we assume a natural ordering between categories, this may result in poor model performance. Hence, as an effort to improve the model performance, one-hot encoding has been performed for target variables where one column is created for each category of the variable.
   b. **Changing the significance value for ADF test and Granger Causality test:** The significance level (p-value) is usually considered 0.05. For performance

improvement, different levels of p-values have been studied and finally, a p-value of 0.1 has been used to reduce the risk of false negatives to happen. In our project, false negatives happen when the model predicts that a fire incident doesn't happen at a particular time, whereas a fire incident occurs at that time. This condition is very risky due to the damage and impact it can cause. Hence, a p-value of 0.1 has been used for both ADF and Granger Causality tests to train and forecast the model.

Why is the VAR model used?

- **Modelling multivariate time-series:** VAR is particularly useful when you have multiple time-series variables that influence each other. Unlike univariate models, VAR can capture the dynamics and relationships between several time-series simultaneously.
- **Capturing Lagged Relationships:** VAR models incorporate lagged values of all variables in the system, allowing them to capture temporal dependencies not only within a single time series but also between different time series.
- **Flexibility and Simplicity:** VAR model doesn't require strong assumptions about the underlying data structure, such as the need for exogenous predictors. It is relatively simple to estimate and interpret, making it a practical choice for multivariate time series analysis.

Possibilities for Poor Performance - Future Considerations:

- **Insufficient Data:** The data like CAD and RMS, weather data used for predictions could be insufficient for the model to understand the patterns and predict the fire emergencies. Considering more data features and real-time data might help in improving the model predictions.
- **Data preprocessing issues:** The missing data can be handled in other ways or data can be scaled using Standard Scaler/MinMax Scaler so that all the features are considered equally.
- **Linear Relationships:** VAR model assumes linear relationships between variables. In case, there is a non-linear relationship between weather data and incident_category_description/DAUID, the model will be performing weakly.

7. User Interface

How to Use the System's Interface

1. **Access the System:** Open your web browser and navigate to http://localhost:4200 (or the deployed URL if the application is hosted on a server).
2. **Login:**
   - Enter your credentials to log in. If you don't have an account, request an admin to create one for you.
   - Admin users can add new users via the admin panel.
3. **Home Page:**
   - The home page displays the current prediction for the next emergency area and category. This information updates every 15 minutes.
4. **Date Selection:**
   - Use the date picker to select a future date. Click the "Search" button to view predictions for that date.
5. **Map Visualization:**
   - The map shows dissemination areas as polygons, with different colors indicating the severity levels of potential emergencies.
   - Hover over or click on the polygons to view details about the predicted emergency type and severity level.
6. **Admin Panel:**
   - Admin users can access the admin panel to manage user accounts. This includes adding new users and managing existing ones.

8. Testing

Test Case ID	Description	Steps	Expected Results
1	Verify UI Elements	1. Open the application. 2.Verify that the date picker for selecting a date is present.. Verify that there is a button or trigger to get predictions for the selected date. 4. Verify that the current date predictions section is present. 5. Verify that there is a map or list to display severity levels.	All elements should be present and correctly displayed.
2	Select Date and Get Predictions	1. Open the application. 2. Select a date using the date picker. 3. Click on the button or trigger to get predictions for the selected date. 4. Verify that predictions are displayed.	Predictions for the selected date should be displayed correctly.
3	Get Current Date Predictions	1. Open the application. 2. Verify that predictions for the current date are automatically displayed or can be triggered.	Predictions for the current date should be displayed correctly.
4	Predictions Every 15 Minutes	1. Open the application. 2. Select a date and get predictions. 3. Verify that the predictions are updated every 15 minutes.	Predictions should refresh or update every 15 minutes.
5	Map Severity Levels	1. Open the application. 2. Select a date and get predictions. 3. Verify that severity levels are correctly mapped and displayed on the map or list.	Severity levels should be correctly mapped and displayed.
6	Validate Severity Levels	1. Open the application. 2. Select a date and get predictions. 3. Manually verify that the severity levels match expected values.	Severity levels should match expected values.
7	Error Handling for Invalid Dates	1. Open the application. 2. Select an invalid date (e.g., a future date beyond prediction capability). 3. Attempt to get predictions.	The application should display an appropriate error message.
8	UI Responsiveness	1. Open the application on various devices (e.g., desktop, tablet, mobile). 2. Perform the steps to get predictions and map severity levels.	The UI should be responsive and functional across all devices.
9	Data Accuracy	1. Open the application. 2. Select a date and get predictions. 3. Compare the displayed predictions with known accurate data (e.g., from a reliable data source).	The displayed predictions should match the known accurate data.
10	Performance Test	1. Open the application. 2. Simulate multiple users selecting dates and getting predictions simultaneously. 3. Monitor the application's performance.	The application should perform well without significant delays or crashes.
11	Login	1. Open the application. . 2. Enter the provided login username and password. 3. Click login.	Homepage should open without any error message or delay.
12	Login Validation Condition 1	1. Open the application. . 2. Wrong entry of username and password. 3. Click login.	Error popup is shown as wrong credentials entered.

13	Login Validation Condition 2	1. Open the application. . 2. Empty form submission. 3. Click login.	Error popup is shown as username and password required.
14	Add User	1. Open the application. . 2. Click add user from home page. 3. Enter all details and add user.	User is added successfully.
15	Add User Condition 1	1. Open the application. . 2. Entered username already exists 3. Click add user	Error is shown mentioning username already exists.
16	Add User Condition 2	1. Open the application. . 2. Empty form submission 3. Click add user	Error popup is shown as username and password required.
17	Authentication and authorization	1. Open the application. . 2. Try to access any unthorized api url using browser or other application.	Error is returned in response asking proper tokens and headers.
18	Prophet Model Result	1. Open the application. . 2. Login to homepage. 3. Checked prophet model result card.	Result is properly displayed in the prophet model card.

9. Security and Privacy

Security features are essential in ensuring that an authentication system remains protected against unauthorized access and vulnerabilities. The following are the data security measures that are implemented in this project to make the application fully secured and protected.

1. **User Authentication and JWT (JSON Web Tokens):** The system employs JWTs to securely transmit information between the client and server. These tokens are digitally signed, ensuring the integrity and authenticity of the data exchanged. By using JWTs, user credentials are securely verified without exposing passwords. The tokens are configured with expiration times, minimizing the risk of misuse, even if a token were to be intercepted.
2. **Role-Based Access Control (RBAC):** Role-Based Access Control (RBAC) is implemented to enhance security by restricting access to certain features of the application based on user roles. For example, administrative users have full access to manage users, while regular users have limited permissions. This system ensures that even if a user's credentials are compromised, the potential damage is contained due to their restricted role-based permissions.
3. **Data Protection and Encryption:** Sensitive data, such as passwords, is protected using strong encryption algorithms like PBKDF2 (Password-Based Key Derivation Function 2).The communication between the client and server is secured using HTTPS, ensuring that data in transit is encrypted and protected from interception. JWTs used in the system can also be encrypted to provide an additional layer of security.
4. **Frontend Security Features:** The frontend is designed with security in mind, incorporating features such as CSRF tokens and secure storage practices. JWTs are stored securely using HttpOnly cookies to protect against common web vulnerabilities like cross-site scripting (XSS) and cross-site request forgery (CSRF).

The authentication system is well-equipped to defend against unauthorized access and attacks by including various security elements, guaranteeing the safety of the system and its users. By securing user data and privacy, this implementation not only improves the application's security posture but also builds user trust.

10. Challenges and Solutions

Challenge 1: Using Dissemination Areas (DAUID)

To identify the next emergency area, we used dissemination areas by converting locations to DAUID. However, several challenges emerged:

- **Data Aggregation:** While working with the CAD RMS emergency data, we faced difficulties in aggregating the data due to missing DAUID values. The client-provided data had some locations without DAUIDs, raising concerns about the dataset's accuracy, as every location should ideally have an associated DAUID.

Solution: We addressed this issue by using shapefiles from the Statistics Canada website to determine the DAUIDs based on longitude and latitude. We then added a new DAUID column to the CAD RMS data, ensuring that every location had a corresponding DAUID, thereby improving the completeness and accuracy of the dataset.

Challenge 2: Handling Sprinkler System Data

Another challenge was dealing with sprinkler system data, which was only available for commercial buildings. This data could not be directly integrated with CAD RMS emergency data because it was not related to DAUIDs and represented different aspects of emergency management.

Solution: Since the sprinkler system data was not directly applicable to our DAUID-based emergency predictions, we treated it as a separate dataset. This allowed us to focus on CAD RMS data for predicting emergency areas, without trying to force a connection where none existed.

Challenge 3: Handling Incident Categories

The original dataset contained around 800 different incident types, which were too numerous and granular for effective predictions.

Solution: We streamlined the incident categories by using the "category description" field, which grouped the incident types into more manageable and relevant categories as per client suggestions. This made the dataset more suitable for prediction tasks, improving the model’s performance.

Challenge 4: Real-Time Data Integration

We faced limitations in integrating real-time data due to data privacy and security concerns. This restriction prevented us from comparing our prediction results with actual real-time data.

Solution: Given the lack of real-time data, we relied on the most recent data available from December 2023 for making current and future predictions. However, this required multiple iterations of predictions to bridge the time gap, which in turn affected the model's performance.

Challenge 5: DAUID Accuracy

The DAUIDs used in the predictions were based on four years of data, which might not reflect the current accuracy due to data limitations.

Solution: While improvements are still needed in the model, we achieved good accuracy for single predictions, particularly with incident category predictions, which were 90% correct. However, DAUID predictions require further refinement for better accuracy.

Challenge 6: Severity Index Calculation

Calculating the severity index was challenging due to incomplete data. Some critical factors could not be included because the necessary data was unavailable.

Solution: We calculated the severity index using available census and CAD RMS data. While this provided a basic measure, the accuracy of the severity index could be improved with more comprehensive data.

Challenge 7: Preprocessing Large Datasets for Predictions

When creating sequences for 10 or more predictions together, the machine couldn’t handle the large preprocessed files required for test data.

Solution: To manage this limitation, we restricted testing to 1, 4, 8, and 10 predictions at a time. This allowed us to run the model within the constraints of available computational resources, though it also highlighted the need for more efficient data handling methods for larger predictions.

11. Future Work

1. Enhanced Data Integration:

- Incorporating Additional Data Sources:
  - Integrate more diverse data sources, such as social media feeds, traffic data, and economic indicators, to provide a more comprehensive view of the factors influencing emergency events. This could enhance the predictive capabilities of the models by considering a wider range of variables.
- Real-Time Data Integration:
  - Establish secure, real-time data pipelines that can continuously feed updated information into the prediction models. This would enable the system to provide more accurate and timely predictions, adapting quickly to changes in real-world conditions.

2. Advanced Modeling Techniques:

- Hybrid Models:
  - Combine different modelling approaches, such as LSTM and Prophet, to leverage the strengths of each. For instance, LSTM could be used for capturing complex temporal dependencies, while Prophet could identify trends and seasonality. This hybrid approach could lead to more robust and accurate predictions.
- Incorporation of Machine Learning Techniques:
  - Explore the use of other machine learning algorithms, such as Gradient Boosting Machines (GBM) or Random Forests, either standalone or in combination with time series models, to improve prediction accuracy, particularly for categorical and spatial data.
- Hierarchical and Multi-Stage Models:
  - Develop hierarchical models that first predict broader categories or regions before refining predictions to more specific levels, such as individual DAUIDs or incident types. This could improve accuracy, especially in cases where data is sparse or highly granular.

3. Improving Accuracy and Handling of Categorical Data:

- Better Handling of Categorical Data:
  - Develop more sophisticated methods for handling categorical variables, especially when using models like Prophet, which are traditionally designed for continuous data. This might include using encoding techniques that better capture the relationships between categories.
- Reducing Prediction Uncertainty:

  - Implement techniques to quantify and reduce prediction uncertainty, such as using ensemble methods or Bayesian approaches. This would provide more reliable predictions and better inform decision-making.

4. Scalability and Computational Efficiency:

- Model Optimization:
  - Optimise models to be more computationally efficient, particularly when dealing with large datasets or generating multiple predictions simultaneously. Techniques like model pruning, quantization, and parallel processing could be explored.
- Cloud-Based Deployment:
  - Consider deploying models on scalable cloud platforms to handle larger datasets and provide faster, real-time predictions. This would also enable more complex model architectures to be used without being limited by local computational resources.

5. Enhanced User Interaction and Decision Support:

- Interactive Dashboards:
  - Develop more advanced and interactive dashboards that allow users to explore different prediction scenarios, adjust parameters, and visualize the impact of different variables on predictions. This could improve user engagement and the practical application of the models.
- Decision Support Systems:
  - Integrate the predictive models into a broader decision support system that not only predicts emergencies but also suggests optimal resource allocation, response strategies, and risk mitigation measures. This would enhance the practical utility of the predictions.

6. Continuous Learning and Model Adaptation:

- Implementing Continuous Learning:
  - Introduce continuous learning mechanisms that allow models to update and improve over time as new data becomes available. This could help the system adapt to changing patterns and improve its predictive accuracy over time.
- Feedback Loops for Model Improvement:
  - Establish feedback loops where user input or real-world outcomes are fed back into the model to refine predictions and improve accuracy. This would create a more dynamic and responsive system.

References

ArcGIS. (2021). *Canada population estimates by dissemination area (DA).* Retrieved August 13, 2024, from https://experience.arcgis.com/experience/4611a367f9714f278358ef8983f44a23

Statistics Canada. (2022, December 15). *Dissemination area.* Retrieved August 13, 2024, from https://www.statcan.gc.ca/search/results/site-search?fq=stclac:2&q=Dissemination%20area&wb-srch-sub=

Statistics Canada. (2022, July 13). *Population estimates by dissemination area.* Retrieved May 13, 2024, from https://www150.statcan.gc.ca/t1/tbl1/en/cv.action?pid=9810012901

ArcGIS. (2021). *Map viewer: Canada dissemination areas.* Retrieved August 13, 2024, from https://www.arcgis.com/apps/mapviewer/index.html?layers=0d0ae0aa40d44e71a2ac48533a0c2de4