(Your cover page should include the information below, but feel free to format and style as desired)

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering

University of Toronto

ECE496Y Design Project Course

Group Final Report

Title: **Digital Twin Model for Police and Emergency Events in Toronto**		
Team #:	2024102	
Team members:	Name:	Email:
	Kartik Tyagi	kartik.tyagi@mail.utoronto.ca
	Shreyan Mahalanabis	shreyan.mahalanabis@mail.utoronto.ca
	Lung Guang Jeffrey Li	lungguang.li@mail.utoronto.ca
	Jerry Jiarui Wang	jerryjr.wang@mail.utoronto.ca
Supervisor(s):	Alberto Leon-Garcia	
Section #:	9	
Administrator:	Eric Lefebvre	
Submission Date:	March 21, 2025	

Group Final Report Attribution Table

This table should be filled out to accurately reflect who contributed to each section of the report and what they contributed. Provide a **column** for each student, a **row** for each major section of the report, and the appropriate codes (e.g. 'RD, MR') in each of the necessary **cells** in the table. You may expand the table, inserting rows as needed, but you should not require more than two pages. The original completed and signed form must be included in the hardcopies of the final report. Please make a copy of it for your own reference.

Section	**Student Names**			
All				

Abbreviation Codes:
Fill in abbreviations for roles for each of the required content elements. You do not have to fill in every cell. The "**All**" row refers to the complete report and should indicate who was responsible for the final compilation and final read through of the completed document.

RS – responsible for research of information
RD – wrote the first draft
MR – responsible for major revision
ET – edited for grammar, spelling, and expression
OR – other
"All" row abbreviations:
FP – final read through of complete document for flow and consistency
CM – responsible for compiling the elements into the complete document
OR - other
If you put OR (other) in a cell please put it in as OR1, OR2, etc. Explain briefly below the role referred to:
OR1: enter brief description here
OR2: enter brief description here

Signatures
By signing below, you verify that you have read the attribution table and agree that it accurately reflects your contribution to this document.

Name	**Shreyan Mahalanabis**	**Signature**	**Shreyan Mahalanabis**	**Date:**	**03/21/2024**
Name		**Signature**		**Date:**	
Name		**Signature**		**Date:**	
Name		**Signature**		**Date:**	

Voluntary Document Release Consent Form

To all ECE496 students:

To better help future students, we would like to provide examples that are drawn from excerpts of past student reports. The examples will be used to illustrate general communication principles as well as how the document guidelines can be applied to a variety of categories of design projects (e.g. electronics, computer, software, networking, research).

Any material chosen for the examples will be altered so that all names are removed. In addition, where possible, much of the technical details will also be removed so that the structure or presentation style are highlighted rather than the original technical content. These examples will be made available to students on the course website, and in general may be accessible by the public. The original reports will not be released but will be accessible only to the course instructors and administrative staff.

Participation is completely voluntary and students may refuse to participate or may withdraw their permission at any time. Reports will only be used with the signed consent of all team members. Participating will have no influence on the grading of your work and there is no penalty for not taking part.

If your group agrees to take part, please have all members sign the bottom of this form.

Consent Statement

We verify that we have read the above letter and are giving permission for the ECE496 course coordinator to use our reports as outlined above.

Team #: ___2024102_____ Project Title:

____

Supervisor(s): ________________________ Administrator: ________________________

Name		**Signature**		**Date:**	
Name		**Signature**		**Date:**	
Name		**Signature**		**Date:**	
Name		**Signature**		**Date:**	

Updated: March 19, 2022

Group Highlights

(1 page)
The one-page ‘Group Highlights’ summary should indicate what you have achieved as a team.

As a team, we can look back and be proud of the work we have done. This sense of achievement is the greatest reward we could ask for. While we faced challenges at the start of the semester, having to balance school work, capstone, and job applications, we gradually found our rhythm. As the semester progressed and we had a couple of meetings together, what once seemed like an impossible task, suddenly seemed attainable.

To complete our project, we applied a divide-and-conquer approach. Initially, we split the work among Jerry and Kartik to complete the model while Shreyan and Jeffrey worked on data preparation. Later on, we collaborated effectively and cleared the backlog together. Beyond the technical aspects of the project, we made sure to communicate effectively and cover each other's weaknesses. Through our countless in-person meetings, and many nights staying up working on documents and practicing presentations, at the end of the day, we can say it was all worth it.

Individual Contributions

(1 page per student)
The one-page ‘Individual Contributions’ can be an updated version of the 'Individual Contributions, from the last Design Review Meeting, with the emphasis now on your final personal contributions and achievements.

Jeffrey

My contributions to the project centered on dataset preparation for the model. I did research across Open Data portals to find appropriate datasets to use in our analysis. I performed correlation analysis and removed redundant low-correlated features while keeping highly correlated ones. One of my main focuses was analyzing points of interest (POIs) across Toronto. I used the OpenStreetMaps API to query geospatial data and extract information such as traffic signals, food places, and buildings of interest. To ensure data quality, I cleaned and pre-processed the datasets by removing missing values and converting text entries to numeric formats. Other contributions included police dataset preparation which was not used in our final model.

Shreyan

My contributions to the project include data collection from the Toronto Open Data Portal. I collected demographic data for each ward. I also worked on data cleaning and data transformation to make the data ready to use to input into the model. This includes linearly interpolating data for missing years in the census data as well as cleaning the weather dataset and normalizing the final dataset. I also researched the best model architecture to use as well as implemented the final chosen architecture. During the implementation, I researched and implemented techniques like parallel GPU training and gradient accumulation to run the model at the largest depth possible.

Kartik

My primary contributions focused on data collection, model development, and building a recommendation system. I started by collecting weather data from various weather stations to obtain daily weather information for each ward. This involved compiling and organizing the data into a comprehensive dataset, ensuring it was clean and consistent for use in model training. For the model architecture, I initially worked on creating a Convolutional Neural Network (CNN) to evaluate embeddings from the tabular dataset. However, after further refinement, we replaced the CNN with a transformer layer to create a more effective FT transformer model, which improved the model's ability to handle complex spatiotemporal patterns. Additionally, I developed a recommendation system that classified wards by risk level based on transformer output. This system also identified the minimum number of patrol units needed and optimized their patrol routes to cover high-risk areas efficiently. My work contributed to enhancing the model's ability to predict and respond to emergency events with greater accuracy and efficiency.

Acknowledgments

We would like to thank our Administrator, Eric Lefebvre, for his full support and advice on improving the communication part of our capstone project. We would also like to thank our Supervisor, Professor Alberto Leon-Garcia, for his continued support in guiding the technical aspect of the project and providing us with guidance whenever required.

Executive Summary

(0.5 – 0.75 page)
The executive summary is about ½ to ¾ of a page that presents the key aspects of the project in a clear, concise, and professional manner. It should stand-alone, meaning it must provide a comprehensive overview without referring to the main document (e.g., no page numbers or cross-references). Instead, it should focus solely on summarizing the project itself.

With the rise in population in Toronto over the years, the frequency of emergency events are following the same upward trend. This increase in emergency events has put significant pressure on current response systems such as the Toronto Fire Service. A more proactive solution is to predict emergency events based on past trends and allocate resources more effectively. However, given the complex spatial and temporal relations in the data we require specialised tools to analyse these relationships. A Transformer allows for the input of categorical data which cannot be accommodated in state of art models like XGBoost. We make predictions based on historical data from 2016-2021 in Toronto. We then present this prediction on a website intended for Toronto Fire Services, providing them with information on high-risk areas at a daily level and providing real time recommendations to proactively respond to emergency events.

Table of Contents

Introduction

This report presents the motivation, design, implementation, and testing of the Digital Twin Model for Police and Emergency Events in Toronto, developed as part of the final-year design course, ECE496. It provides a comprehensive overview of the project's objectives, methodologies, and outcomes, concluding with recommendations for improvements and future work.

Over the past 30 years, Ontario has seen an increase in population from 7.8 million in 1971 to 15.6 million in 2023, and the Government of Ontario is predicting an increase to over 22.1 million by the year 2051 [G]. With this ever-increasing population, the number of emergency events in the city has also increased. According to the City of Toronto, the number of major crime indicators has risen from 32,461 in 2014 to 49,359 in 2023 [H].

The above trend in population, crime and emergency events has put significant pressure on current response systems such as Toronto Police Service (TPS), Toronto Fire Services (TFS), and Emergency Medical Services (EMS). For example, Toronto Fire Services reported an average response time of 6.1 minutes for emergency calls in 2025 [A], while Toronto Police Service recorded an average response time of 18.1 minutes for the same year [B]. Although this is an improvement from the average response time of 22 minutes in 2024 for TPS, it is important to note that this improvement was obtained through significant investments, such as deploying 146 new police officers in frontline operations and promoting 80 Sergeants and Sergeant of 50 employees to strengthen the frontline supervision [B]. However, despite these tremendous efforts, the improvement seen is only marginal. Increasing population and increasing demand for emergency services highlight the need for more new and active solutions.

Predictive modeling and policing offer a promising solution to tackle this critical issue. Predictive modeling is a mathematical process used to predict future events or outcomes by analyzing patterns in a given set of input data [C]. Essentially, it asks the question "Have I seen this before? If so, what do I expect to happen next?" Predictive policing is one such application of predictive modelling and has already been implemented in various cities around the world. One of the earliest adopters of predictive policing was the Los Angeles Police Department (LAPD) in 2008, and the New York Police Department (NYPD) in 2012 [D]. Cases in the United States show that crime rate decreases when predictive policing software is used [E].

While predictive policing has shown promise in reducing crime in the United States, its usage in Toronto has not been as prevalent. A study reveals that Toronto is laying the groundwork for data-driven policing but remains cautious about fully adopting predictive technologies [F]. This gap highlights the need for a robust predictive system that can ensure public safety through data-driven forecasting and resource optimization.

To address this need, the goal of our project is to develop a predictive model to detect anomalies in emergency data and forecast trends, thereby improving response strategies. Simply put, we want to be able to quickly, accurately, and continuously predict the emergency events that may occur in each of the Toronto wards on any given day. By simulating scenarios and testing response strategies, this system will enhance officer preparedness, optimize resource allocation, and ultimately improve public safety in Toronto.

[G] Government of Ontario, "Ontario population projections," Government of Ontario, Sep. 2023. [Online]. Available: https://www.ontario.ca/page/ontario-population-projections#section-3.

[H] City of Toronto, "City of Toronto Open Data Dashboard," Power BI. [Online]. Available: https://app.powerbi.com/view?r=eyJrIjoiMGQyZGFlYTEtZjdiOS00M2VmLWI5NGQtYTI5N2UwNDUyODg3IiwidCI6Ijg1MjljMjI1LWFjNDMtNDc0Yy04ZmI0LTBmNDA5NWFlOGQ1ZCIsImMiOjN9.

[A] Toronto Fire Services, 2023 Annual Report, City of Toronto, 2024. [Online]. Available: https://www.toronto.ca/wp-content/uploads/2024/06/9764-24-00134-Toronto-Fire-Ann-Report-2024-June5-AODA-4-PAC1-2-Dot-op-Rot-FPspread-3.pdf. [Accessed: 11-Feb-2025].

[B] Toronto Police Service, "Chief Reports Improved Response Time," Toronto Police Service, Apr. 30, 2024. [Online]. Available: https://www.tps.ca/media-centre/stories/chief-reports-improved-response-time/. [Accessed: 11-Feb-2025].

[C] https://www.techtarget.com/searchenterpriseai/definition/predictive-modeling

[D] https://www.brennancenter.org/our-work/research-reports/predictive-policing-explained

[E] https://www.tandfonline.com/doi/full/10.1080/01900692.2019.1575664#d1e114

[F] https://citizenlab.ca/2020/09/algorithmic-policing-in-canada-explained/

What follows is a detailed outline of our Project Requirements.

ID	Project Requirement	Description
1	Data Ingestion	**Constraint:** The system must comply with the specific formats of input sources to accurately collect and process real-time data. These sources include police reports, emergency dispatch logs, and various datasets provided by the City of Toronto Open Data portal.
2	Spatio-Temporal Analysis	**Functional requirement:** The model must analyze spatial and temporal patterns to predict the probability of crime or emergency incidents at specific times and locations. This prediction is expressed as a probability, rather than a definitive outcome, to avoid unintended delays in emergency response.
3	Emergency Response Recommendation	**Functional requirement:** The model will generate recommendations for emergency preparedness, indicating the required types and quantities of equipment and personnel. These recommendations will support both police and fire departments, ensuring that they remain adequately prepared for potential emergencies.
4	Real-time Performance and Model Accuracy	**Objective requirement:** The model must handle large volumes of data (e.g., social media and network activity) in near real-time, optimizing computing resources to achieve target processing times. During tests, social media and network activity data from the corresponding timeframe will be used (e.g., testing for emergencies in February 2022 will use February 2022 data), with a processing time target of under 10 seconds. Also to ensure that the predictions are able to correctly distinguish wards with a lower level of emergency events from wards with a higher level of emergency events, we want the RMSE error to be less than 3

Final Design

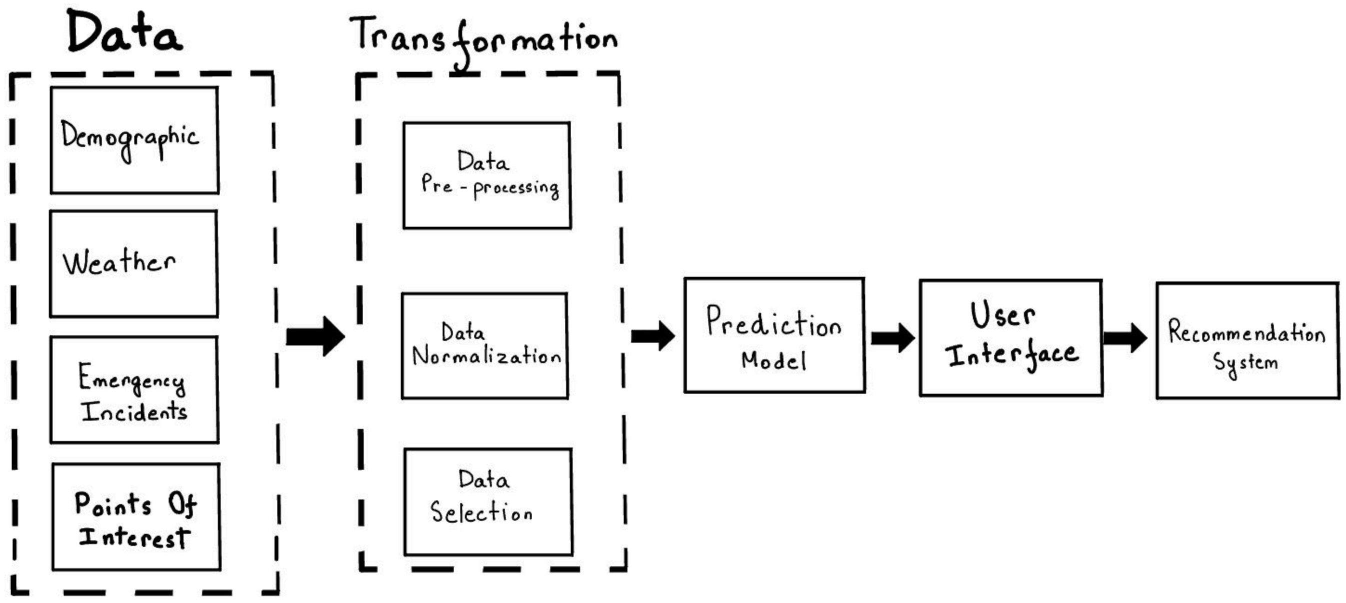
(8 pages)
This is the core section of the report, where you document the final design of your project. While much of the content may be drawn from previous reports and discussions, it should be updated and refined to reflect the completed design. Each team member's contributions should be integrated seamlessly into a well-structured and cohesive report.
Depending on the complexity of your project, this section may consist of multiple chapters with sub-sections, which can vary in title, style, and approach based on the nature of the work. Each sub-section should clearly identify the primary author.
Key Components to Include

• System-Level Overview – Provide a high-level description of the system and its purpose.
• System Block Diagram – Present a clear, labeled block diagram illustrating system components and their interactions.
• Module-Level Descriptions and Designs – Detail the functionality, structure, and design of each module, explaining how they integrate into the system.
• Final Design Assessment – Analyze the effectiveness, limitations, and performance of the final design.
Documentation Guidelines
• Use diagrams, schematics, tables, flowcharts, pseudo-code, photos, equations, and simulation models to clearly present your design.
• Be selective—include key design details in the main report while placing extensive documentation (e.g., full schematics) in the appendix. Always reference the appendix when applicable.

System Level Overview

The project aims to make emergency response proactive instead of reactive. Historical demographic, weather, and emergency event data is input into the prediction model, which outputs the most probable number of emergency incidents at a given location (at the ward level) and at a given time (at the day level). The end user can then view this prediction through a user interface (a website) that conveys easy-to-understand graphics to help the user make a response strategy. The user can also upload weather and demographic data to update the model and get daily predictions. Finally, the user also receives a recommended strategy to address the predicted emergencies effectively.

System Block Diagram



Module-Level Descriptions and Designs

Data

To train the prediction model, we input three historical compiled datasets. We sourced the demographic and emergency incidents (fire incidents) datasets from the Toronto Open Data Portal [] and the weather dataset from the Government of Canada's Historical Weather Data repository[]. We also created a dataset compiling points of interests across the city of Toronto using data from OpenStreetMap [].

- Demographic Data
  - Census data at a ward level (following the latest 25 Ward model used by the Toronto Fire Department) in Toronto for 2016 and 2021.
  - Includes data on income, education, age, type of residence, etc.
- Weather Data
  - Data from 2016 - 2021 at a day level from a weather station in each of the 25 wards in Toronto.
  - Includes data on maximum and minimum temperature, precipitation, wind speed, etc.
- Emergency Incidents
  - Count of Fire Incidents at a day level from 2016-2021 per ward.
  - Includes the type of fire and medical incidents, average response time, etc.
- Points of Interest Data
  - We collected Points of Interest (POI) from the OpenStreetMap for amenity, building and highway categories using the Overpass API [].

Data Transformation

The data collected is cleaned and transformed to help the prediction model better understand how fire incidents are related to demographics and the weather of a ward on any given day. The data is also sorted based on how correlated a certain feature is to fire incidents and only the top 50 features are selected as input to the model.

- Data Pre-Processing
  - Any rows in either dataset are removed if they contain missing values
  - Text entries are converted to a numeric format
  - Since we only have census data from 2016 and 2021, we linearly interpolate the years in between to provide a larger range of historical data to train the model.
  - Removal of cultural and ethnic data from the demographic dataset
- Data Normalization
  - Using Quantile Transformer, all of the data, given a ward and a day is converted to a normal distribution to account for different datatypes (median income $38,000 vs population count 37,000). This also allows the prediction model to expect normally distributed values between [-1,1] up to 15 significant digits.
  - Categorical data like day and week numbers is one-hot encoded to remove any force the prediction model to view them as individual entities rather than numbers.
- Data Selection
  - After preprocessing and normalization, correlation analysis was used to identify the most relevant features for the predictive model. Pearson's Correlation Coefficient [] was applied to measure relationships between variables. A heatmap visualized correlations, highlighting features strongly linked to emergency events. As shown in Figure 2, Low Income and Single detached homes are highly correlated with fire incidents. See Appendix [A] for the entire correlation analysis of our dataset.
  - High-correlation features (e.g., population density, median income, crime rates) were selected whereas low correlation or redundant features were removed to reduce noise.

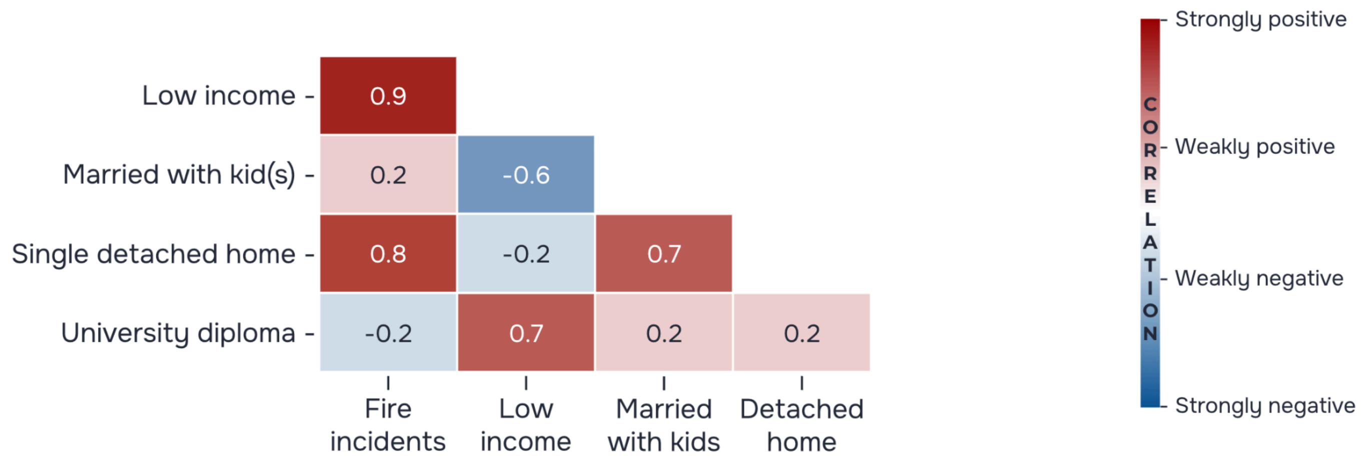
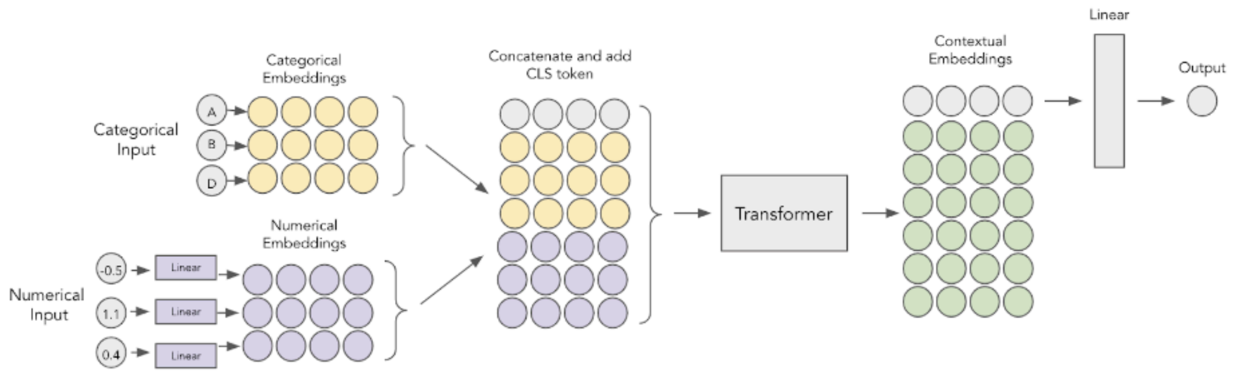


Figure 2: Simplified heat map showing correlation between features and incidents

Prediction Model

The prediction model architecture used is a Fast-Tab Transformer. The data input has categorical (day/week number) and numerical features (income, population count). The model first extracts categorical and numerical data embeddings through a linear layer and then concatenates them to feed into a transformer. The transformer then extracts contextual embeddings using its attention mechanism to find the relationship between each feature and the target value as well as the relationship between the features. This is especially important because a combination of some features might be relevant to fire incidents. For example - a ward with low income might have more homeless people who might try to warm themselves by lighting a fire on a cold day leading to an emergency. On the other hand, these features such as the temperature and income level on their own might not contribute much towards the prediction.

The transformer has 4 attention heads and a depth of 4 layers. To avoid overfitting, we have set the attention and feed-forward dropout to 0.2. So we have 4 units of attention per layer to encapsulate the relationship among features. We have 4 layers in total and 20% of the neurons are turned off during training to force the other neurons to work equally as hard to find the predicted value. This prevents a group of neurons from putting the model on its back during training.



The model is trained on Kaggle using dynamic parallel computing to concurrently utilize two Nvidia T4 GPUs. Each training iteration uses a batch size of 16 data points and gradient accumulation is used on top of that with 4 accumulation steps to mimic a batch size of 64 data points (model trained on 4*16=64 data points every iteration) for 100 epochs. In each training iteration, the model makes a prediction using its computer parameters and compares it to the actual number of incidents to compute its loss (using mean squared error). If the loss is high then the model makes a big update to its parameters and vice versa. In the beginning, every guess is random but over time, as the model makes mistakes and learns from them, it starts to learn the parameter distribution and come up with better predictions for the number of fire incidents in a day.

User Interface

Recommendation System

The recommendation system takes the output of the transformer - a list of 25 numbers, representing the predicted number of emergencies to occur - as an input. Then it outputs the minimum number of patrol units and their optimal patrol paths to cover all ‘high risk’ wards. The recommendation system first classifies each ward as either ‘high risk’, ‘medium risk’, or ‘low risk’ based on a ‘high threshold’ and ‘low threshold’. The values for the two were found by evaluating past data on a curve and finding the 70% (high) and 33% (low) values. Once the classification was made, the high risk wards were placed in a graph network, with connections to adjacent wards. In the end, we had a graph of all high risk wards and taking the minimum number of components gave the minimum number of patrol units. Additionally, components of a smaller length were then connected to adjacent medium risk or low risk wards (priority given to medium risk wards). These components then form the path of the patrol units which is ultimately outputted along with the minimum number of patrol units. The recommendation has been implemented and will be tested on custom test cases and implemented into the UI.

Final Design Assessment

Limitations in Data Processing

We remove data points with missing values instead of making an educated guess about what that value could be. This is not a big issue because we have around 250 MB of data overall but this process leads to removing the information we have on other features in the same datapoint. For eg - in the weather dataset, we have values for snow on the ground for a few wards but we choose to remove that feature from all wards to allow for data integrity. Snow on the ground could be influential in predicting traffic accidents which fall under the fire incidents dataset.

Effectiveness in Data Processing

We normalize the data into a normal (gaussian) distribution. This step makes it possible for our data to have different data types. The model cannot read and interpret what the feature headings mean so a median is treated the same as a count or a percentage. When we normalize the data, all values fall between -1 and 1 so the model does not have an exploding gradient problem. This problem occurs when the model is trying to learn weights through its prediction error. If the values input are too big, then the error function starts to explode into larger and larger values without a considerable impact on learning new weights. Keeping input values bounded solves this problem. Lastly, when the values of a feature say, median income, are close to each other for every ward, say, a range of $10,000, the model sees very little difference in the wards and picks up on no useful relationship. Using a normal distribution, we are essentially finding the distance of each value from the mean of that feature for every ward. Therefore, it provides the model information to compare these distances and find the relationship of each ward for that feature.

Limitations of the Prediction Model

Transformers are large models that require a long time to train and are nig in size. The greatest limitation to making our model better is the complexity or the size of the model. If the model is too simple it cannot pick up the relationships among features and the target and starts to memorize whatever training data you give it. While we use techniques like gradient accumulation to circumvent this problem, it is not possible to get the same results as a bigger transformer model. Given our limitation to two Nvidia T4 GPUs, we can only fit a model of a small size (30 GB). While the state of art transformers have billions of parameters, (GPT-4 has

1.8 Trillion parameters), we can only have a few hundred thousand parameters, limiting the capability of making an accurate prediction.

Effectiveness of the Prediction Model

The transformer architecture allows the use of categorical data (day/month no.) as input which would not be possible if we used other model architectures. We also optimize the training and inference speed of the transformer by using 2 GPUs in parallel. This also provides us the capability of using a model which is at most 30GB in size.

Results of the Prediction Model

The model predicts the incident count with a Root Mean Squared Error of 13.6 which means that it predicts around 14 incidents more or less than the actual value for all 25 wards per day. On average, there are 27 fire incidents per ward from 2016 to 2021 per day with a maximum average of around 80 incidents in ward 13.

Conclusion

final design

The project is designed to be able to predict the number of fire incidents in a day at a ward level. We compile census, weather, and, POI (points of interest) data and apply data transformations to feed it into a Fast-Tab Transformer prediction. The prediction model utilizes techniques like parallel computing and dropout to optimize and increase prediction accuracy. The prediction model gets a Root-Mean Squared Error (RMSE) of 13.6 on its prediction for the number of fire incidents in a day at each ward. We then display these predictions on a user-friendly website and further provide recommended actions to best address the predicted fire incidents. We also discuss the limitations of our approach and where our design is effective at tackling challenges faced during implementation.

Testing and Verification

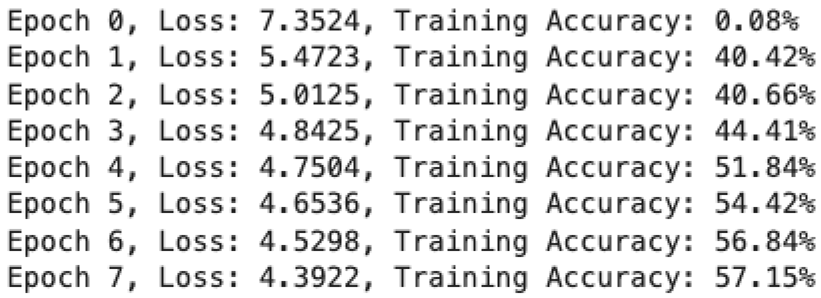
As discussed, the goal of our project is to be able to accurately, quickly, and continuously predict the emergency events that may occur in each of the Toronto wards on any given day. Our main motivation to pursue the project is that this prediction will allow us to recommend to the authorities high-risk and low-risk zones, while also recommending minimum patrol units and optimal patrol routes. As discussed in the introduction, we have four major categories for our project requirements:

1. Data Handling and Input/Output Flexibility
2. Spatial and Temporal Predictions
3. Incident Response Recommendation
4. Response Time and Accuracy

Data Handling and Input/Output Flexibility

In our journey to build the Fast Tab Transformer to predict emergencies, we encountered a road block very early on - Data gathering and handling. In order to predict the emergencies, our model needed a wide variety of datasets: past police emergencies dataset, past fire emergencies dataset, past medical records dataset, weather dataset, and demographic dataset. Our immediate concern was for our transformer to be able to effectively analyze all of this data, it would either need to be able to input and train on all of them in parallel, or we focus on data processing to ensure we create one large readable dataset for the model. Therefore, our training process should satisfy either of the two criteria to ensure we have a fair and effective training process. To ensure that our model is able to train and read the model as expected, we tried training on a small portion of the training and validation data and track the training error for each training cycle. Ideally, the training error should decrease, showing that our model is able to learn on the provided dataset.

- **Test Case ID:** 1.1
- **Test Case Title:** Tracking training error rate
- **Test Objective:** To verify that the dataset allows the model to learn quickly
- **Preconditions:** The dataset is compiled as expected and transformer model architecture is built and ready to be trained
- **Expected Results:** Training error should keep decreasing as it runs through the training cycle
- **Comments/Notes:** Even if the test passes, we will continue running the test to see if we can improve the quality of our dataset



```
Epoch 0, Loss: 7.3524, Training Accuracy: 0.08%
Epoch 1, Loss: 5.4723, Training Accuracy: 40.42%
Epoch 2, Loss: 5.0125, Training Accuracy: 40.66%
Epoch 3, Loss: 4.8425, Training Accuracy: 44.41%
Epoch 4, Loss: 4.7504, Training Accuracy: 51.84%
Epoch 5, Loss: 4.6536, Training Accuracy: 54.42%
Epoch 6, Loss: 4.5298, Training Accuracy: 56.84%
Epoch 7, Loss: 4.3922, Training Accuracy: 57.15%
```

Image 3.1: Decreasing training error

In addition to our transformer, we built a UI system that is able to display the results of our predictions along with the recommendations based on those predictions. Considering the fact that we are unsure of the exact expectations of the clients for our UI model, our layout and display needs to be adaptable. This means we want to be able to adjust the layout and design quickly and add or remove functionalities from it while preserving design consistency. To achieve this goal, we have to ensure 2 things:

- UI is accessible and easy to use to minimize the need for changes
- Ensure that the UI is simple and intuitive. This guarantees that any changes in the design will be based on client expectations and no further adjustments will be needed to be made to preserve consistency.

To address the first bullet point, we can ensure the UI is accessible if the use of colors and contrasts along with the data visualization is such that it is very easy to understand, even if one has bad eyesight or is suffering from color blindness.

- **Test Case ID:** 1.2
- **Test Case Title:** Color contrast through focus group testing
- **Test Objective:** To ensure that the UI is readable by those with weak eyesight or color blindness
- **Preconditions:** The UI is built
- **Expected Results:** All testers verify that they have no problem following through with the contents of the UI

Additionally, we can add features like keyboard navigation for easier use, improving the accessibility of our UI module. This brings us to the point of ease of use, which mainly refers to the UI being intuitive, such that the navigation can be easily understood without guidance. These will be verified by conducting user testing ourselves and with the help of select alpha testers, who have minimal knowledge of the project.

- **Test Case ID:** 1.3
- **Test Case Title:** Keyboard navigation
- **Test Objective:** To ensure our UI is accessible
- **Preconditions:** The UI is built
- **Expected Results:** Keyboard navigation is possible

One look at the UI (Refer to Image 4) and some knowledge of what the project is about, and we are able to tell what this image shows - the 25 wards of Toronto along with the number of emergency events expected to occur in the ward on that day. This is exactly what we want for the interface to be simple and intuitive. There is no need to switch to a different page to view information not visible. There are no hidden features the user may need to explore the UI to find. And, the displayed information resembles maps and weather maps viewed everyday. It is simple, and intuitive, and ensures design consistency no matter what changes may be suggested by the client.

- **Test Case ID:** 1.4
- **Test Case Title:** Simple UI Design
- **Test Objective:** To verify that all information is instantly available on our UI in a readable manner
- **Preconditions:** The UI is built
- **Expected Results:** We are able to view the wards, predictions, and recommendations at once

Spatial and Temporal Predictions

A key functional requirement for our model is to be able to predict emergencies in Toronto given ‘space’ and ‘time’. At first we struggled with how to define the ‘space’ and ‘time’ for our model. We figured that it was linked to data handling and how we want to process our data to help our model best understand any correlations. Upon evaluation of our datasets, we found to have sufficient information (information from all datasets) at the ward level.

- **Test Case ID:** 2.1
- **Test Case Title:** Defining predictions at ward level
- **Test Objective:** To ensure we are able to correctly compile the datasets around the ward feature to allow predictions at a ward level
- **Preconditions:** All datasets gathered are presented at a ward level
- **Expected Results:** The final compiled dataset is at the ward level

Additionally, we wanted the number of past emergencies to be a major factor to be considered by our transformer model. And so, to avoid overwhelming reports of no emergency events (which would cause our model to learn to predict no emergency events will ever occur), we decided to define ‘time’ as the day. Therefore, we want to be able to predict the number of emergency events that may occur in each ward at a given day.

- **Test Case ID:** 2.2
- **Test Case Title:** Defining predictions at day level
- **Test Objective:** To ensure we are able to correctly compile the datasets around the day feature to allow predictions at a day level
- **Preconditions:** All datasets gathered are presented at a day-level
- **Expected Results:** The final compiled dataset is at the day level

On the note of adjusting the focus of the transformer, we want to ensure that we only pass on relevant features. This means we have to find dependencies amongst the current features through correlation analysis, a machine learning methodology that allows us to find how strongly one feature impacts the other. If the dependency is strong, that means only one of the two features is required for analysis by the transformer, since the other can be derived. Thus, we need to perform correlation analysis and ensure any irrelevant features are deleted from the dataset.

- **Test Case ID:** 2.3
- **Test Case Title:** Correlation Analysis
- **Test Objective:** To find irrelevant or redundant features to improve the quality of data for the transformer
- **Preconditions:** The dataset is compiled as expected
- **Expected Results:** Able to identify minimum independent features and pass them to transformer

On the UI side, we want to present these findings in a readable manner. For ease of understanding, we decided it's best to visually represent the wards and display the number[1]. This means we need to be able to visually see the city of Toronto divided into 25 wards (See image 5.2).

- **Test Case ID:** 2.4
- **Test Case Title:** Visual representation of wards
- **Test Objective:** To ensure a greater understanding of the UI
- **Preconditions:** The UI is built
- **Expected Results:** All 25 wards can be viewed as part of Toronto

Additionally, we want to be able to present our output of the transformer and recommendation models.

- **Test Case ID:** 2.5
- **Test Case Title:** Predictions are clearly visible and understood by alpha testers
- **Test Objective:** To ensure readability of transformer output on UI
- **Preconditions:** The transformer model is trained and the UI is built
- **Expected Results:** All alpha testers should verify that they were able to clearly understand and follow the UI information including the predictions

- **Test Case ID:** 2.6
- **Test Case Title:** Recommendations are clearly visible and understood by alpha testers
- **Test Objective:** To ensure readability of recommendation system output on UI
- **Preconditions:** The transformer model is trained and the UI is built
- **Expected Results:** All alpha testers should verify that they were able to clearly understand and follow the UI information including the recommendations

All of this should be done without cluttering the UI to keep it clean and simple. We must also consider the fact that these emergencies can be either fire-related or police-related. Not classifying them as such would go against 'keeping it simple', so it is best to identify the numbers and types of emergencies.

- **Test Case ID:** 2.7
- **Test Case Title:** The type of emergency is visible
- **Test Objective:** To ensure readability and improve understanding of the output information
- **Preconditions:** The transformer outputs the types of emergencies as well and the UI is built

- **Expected Results:** Able to clearly distinguish the number of police and fire emergencies expected to occur on the UI screen

Incident Response Recommendation

As described earlier, our main motivation behind the project is to be able to recommend high risk areas and patrol plans to reduce the emergency response rates, and hopefully, be able to save more lives. One key feature for that is to develop a recommendation system which, when given the list of predicted number of emergencies in each ward, is able to:

1) Classify the wards as High Risk or Low Risk
2) Identify the minimum number of patrol units required to cover all High-Risk wards
3) Identify the optimal path for each patrol unit

In order to test whether our recommendation system is able to provide us with the correct results, we will be testing it across many different custom sets of transformer outputs to ensure it passes even edge cases (for example, when there are no high-risk wards, our model shouldn't say don't patrol).

- **Test Case ID:** 3
- **Test Case Title:** Accuracy of recommendation system
- **Test Objective:** To ensure that the recommendations provided are correct estimates based on our transformer predictions
- **Preconditions:** The recommendation system is ready and custom test cases have been built
- **Expected Results:** All test cases pass

Response Time and Accuracy

While it is important to verify all our functional requirements, it is important to test the objective requirements to see if our model would perform as well as intended. We have two main objective requirements:

- Response Time
- Accuracy

Response time is a very important objective. As time goes on, the 'past' data will only increase, meaning, we will constantly have new data to process and train on. In order for our model to seemingly function uninterrupted by new data, we must be able to process this data in 'real-time'. A more concrete definition says we must be able to process 1 day's worth of data in less than 1 minute.

- **Test Case ID:** 4.1
- **Test Case Title:** Transformer processing speed
- **Test Objective:** To ensure that the transformer is able to process new data in 'real-time'
- **Preconditions:** The transformer architecture is ready and has been partially trained
- **Expected Results:** Able to train on a new set of 1-day data in less than 1 minute

However, the response time of the transformer alone is not sufficient, because the user will be more interested in the interface response rate. So, what the focus of the objective should be is user interaction. This means that the response time of the UI itself must also be in real-time and should be able to update the output information within 1 second so as to keep the user's train of thought uninterrupted[2].

- **Test Case ID:** 4.2
- **Test Case Title:** Processing Speed of UI

- **Test Objective:** To ensure the UI is able to update itself with changes in transformer output while keeping the user's train of thought uninterrupted
- **Preconditions:** UI is built and the transformer is partially trained
- **Expected Results:** The time taken for the UI to update is less than 1 second

The UI is not the only thing affected by a change in transformer output - the recommendation system must also be able to calculate in 'real-time', which for a path-finding algorithm like this is less than 0.5 seconds.

- **Test Case ID:** 4.3
- **Test Case Title:** Processing speed of the recommendation system
- **Test Objective:** To ensure that the recommendation system is able to find the minimum patrol units and optimal path in 'real-time' for a path-finding algorithm
- **Preconditions:** The recommendation system is ready
- **Expected Results:** Recommendation system outputs within 0.5 seconds

In addition to response time, accuracy is a key objective that we have been working towards the entire year. We faced many roadblocks in building a spatio-temporal transformer model. We first found that a transformer is not optimal for analyzing tabular data. To get around that, we worked with multiple layers so that the first acts as the encoder for the second to help the second understand tabular data. However, to be able to accurately distinguish between high-risk and low-risk wards, we want our RMSE error to be below 3, an ambitious goal, which if successful, will mean we successfully created an AI model to predict real-world emergencies.

- **Test Case ID:** 4.4
- **Test Case Title:** Accuracy of transformer model
- **Test Objective:** To ensure that transformer predictions are accurate enough to avoid confusion between high-risk and low-risk wards
- **Preconditions:** The transformer is trained
- **Expected Results:** The final RMSE error should be less than 3

Testing Table

The testing and verification results discussed above can be exemplified in the following 'Testing Table':

ID	Requirement	Method	Result and Proof
1.1	The transformer should show a constant decrease in training error	TEST: Track the training error	PASS: Verified, the training error decreased. See image 3.1
1.2	UI is visible and understood by people with bad eyesight or color blindness	TEST: Focus group testing	PASS: All testers verified. See appendix C
1.3	Keyboard navigation is present	Evaluation of the final product	UNTESTED: No keyboard navigation is required. See section 2
1.4	All information is present at once	Evaluation of the	PASS: Verified, all

		final product	information was displayed
2.1	All datasets should be compiled together at the ward level	Evaluation of the final dataset	PASS: Verified, datasets are at the ward level
2.2	Weather data compiled at the day level	Evaluation of the final dataset	PASS: Verified, dataset is at the day level
2.3	Perform correlation analysis and remove irrelevant features	ANALYSIS: identify the minimum independent features	PASS: Verified, was performed, and checked with the supervisor
2.4	Visual representation of wards in UI	Evaluation of the final product	PASS: Verified, Toronto was divided into 25 wards
2.5	The output of the transformer is visible and readable	TEST: Alpha Testing	PASS: All testers verified. See appendix C
2.6	The output of the recommendation system is visible and readable	TEST: Alpha Testing	UNTESTED: See section 5
2.7	Output the types of emergencies as well	Evaluation of the final product	FAIL: Only output is number. See section 2
3	The recommendation system should be able to give accurate suggestions	TEST: test across various custom sets of inputs	UNTESTED: See section 5
4.1	Train on 1 day data in less than 1 minute	TEST: test on 1-day data	PASS: Verified, took 19.73 seconds. See Appendix D
4.2	UI displays within 1 second	TEST: Send data to UI	FAIL: Took 1.5 seconds
4.3	Recommendation given in less than 0.5 seconds	TEST: Check the time of run	FAIL: Took 4.2 seconds
4.4	Achieve a RMSE < 3	TEST: Check the RMSE error	FAIL: Current error of 13.6. See section 2

References

References are essential for accurate documentation and must be provided for any quoted or copied material. Use original sources such as books, journals, and standards whenever possible. List references in order of appearance, on a separate page, with proper indentation for clarity. Follow the IEEE Citation Guidelines [2].

[1]E. Cleary, "Visualization to Improve the Speed of Understanding," *simpleshow*, Aug. 30, 2017. https://simpleshow.com/blog/visualization-improve-speed-understanding/
[2]J. Nielsen, "Response Time Limits: Article by Jakob Nielsen," *Nielsen Norman Group*, Jan. 01, 1993. https://www.nngroup.com/articles/response-times-3-important-limits/

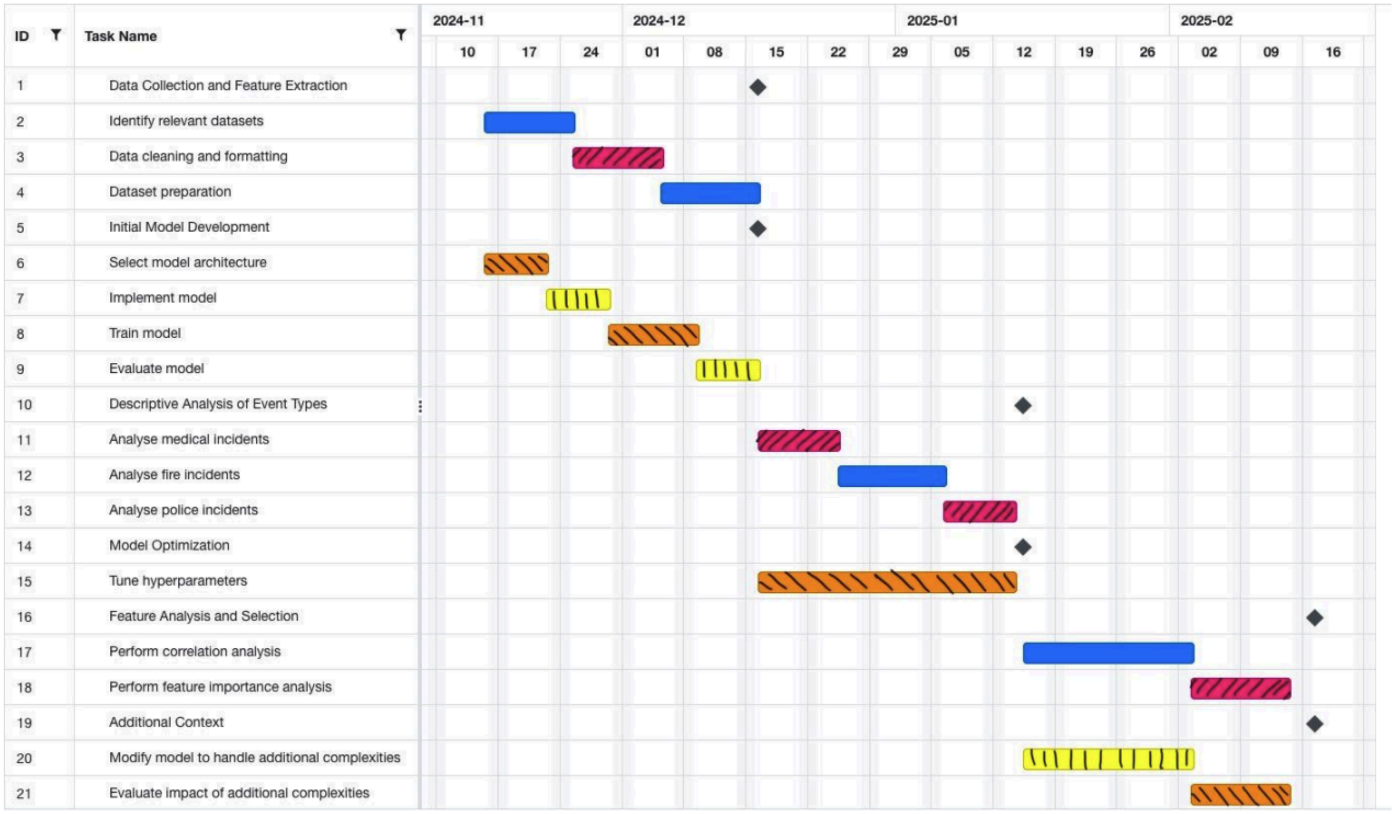
Appendices

Appendices provide supplementary material to enhance the reader's understanding of your project. Keep these key points in mind:
• The main document should be self-contained; appendices should only provide additional details.
• Each appendix must be stand-alone and understandable without referencing the main text or other appendices.
• Appendices may include figures, tables, maps, photos, raw data, code snippets, or background information. Only include relevant material.
• Appendices must appear in the Table of Contents. If there is only one, title it "Appendix".

Appendix A: Gantt Chart History

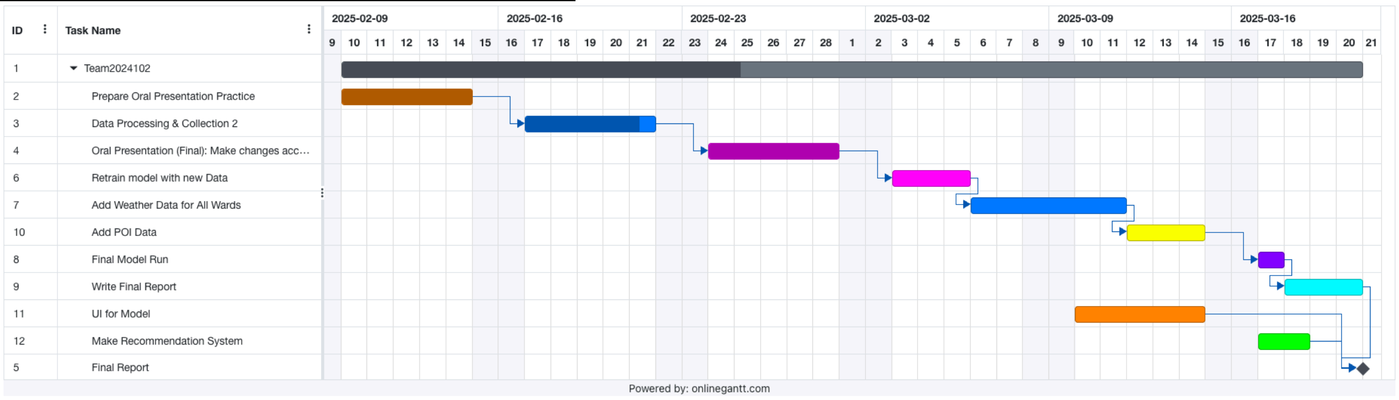
Include the final Gantt chart along with previous versions to show project progress.

Gantt Chart from September,2024 to Feb 9th, 2025



ID	Task Name
1	Data Collection and Feature Extraction
2	Identify relevant datasets
3	Data cleaning and formatting
4	Dataset preparation
5	Initial Model Development
6	Select model architecture
7	Implement model
8	Train model
9	Evaluate model
10	Descriptive Analysis of Event Types
11	Analyse medical incidents
12	Analyse fire incidents
13	Analyse police incidents
14	Model Optimization
15	Tune hyperparameters
16	Feature Analysis and Selection
17	Perform correlation analysis
18	Perform feature importance analysis
19	Additional Context
20	Modify model to handle additional complexities
21	Evaluate impact of additional complexities

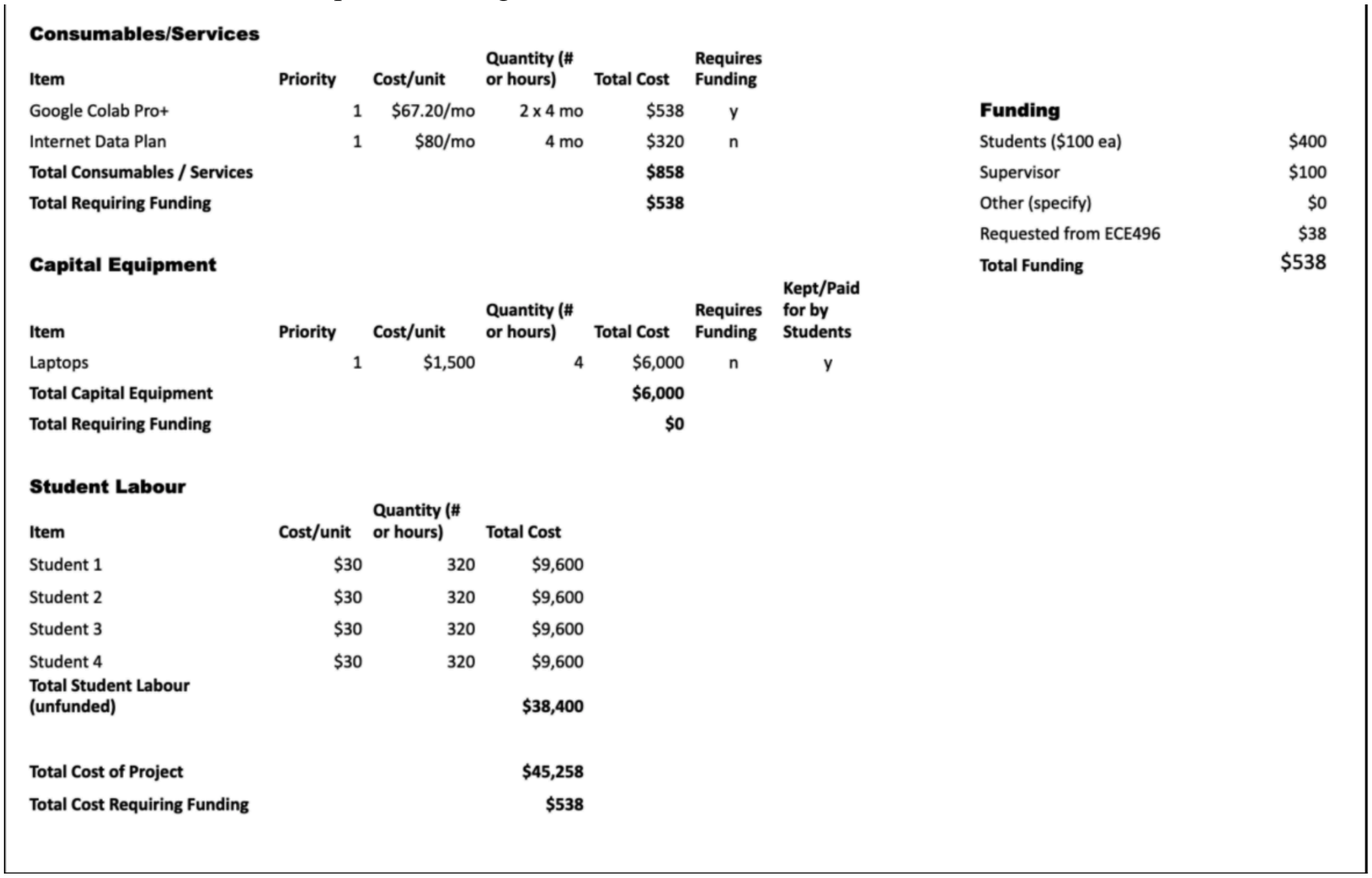
Gantt Chart from Feb 10th to Mar 21st, 2025



ID	Task Name
1	Team2024102
2	Prepare Oral Presentation Practice
3	Data Processing & Collection 2
4	Oral Presentation (Final): Make changes acc…
6	Retrain model with new Data
7	Add Weather Data for All Wards
10	Add POI Data
8	Final Model Run
9	Write Final Report
11	UI for Model
12	Make Recommendation System
5	Final Report

Appendix B: Financial Plan

Provide the final financial plan and budget table



Consumables/Services

Item	Priority	Cost/unit	Quantity (# or hours)	Total Cost	Requires Funding
Google Colab Pro+	1	$67.20/mo	2 x 4 mo	$538	y
Internet Data Plan	1	$80/mo	4 mo	$320	n
Total Consumables / Services				**$858**	
Total Requiring Funding				**$538**	

Funding

Students ($100 ea)	$400
Supervisor	$100
Other (specify)	$0
Requested from ECE496	$38
Total Funding	$538

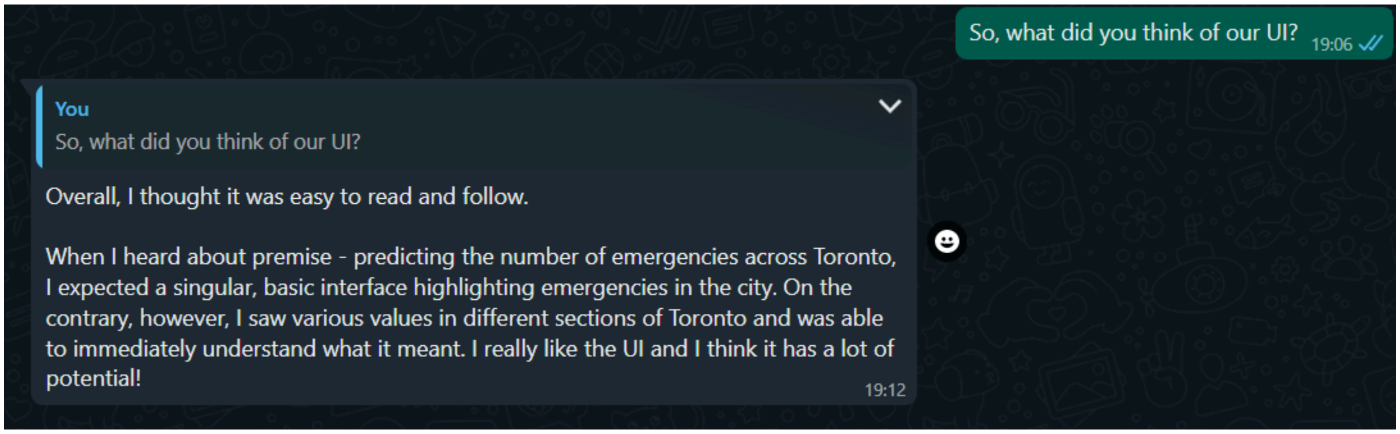
Capital Equipment

Item	Priority	Cost/unit	Quantity (# or hours)	Total Cost	Requires Funding	Kept/Paid for by Students
Laptops	1	$1,500	4	$6,000	n	y
Total Capital Equipment				**$6,000**		
Total Requiring Funding				**$0**		

Student Labour

Item	Cost/unit	Quantity (# or hours)	Total Cost
Student 1	$30	320	$9,600
Student 2	$30	320	$9,600
Student 3	$30	320	$9,600
Student 4	$30	320	$9,600
Total Student Labour (unfunded)			**$38,400**
Total Cost of Project			**$45,258**
Total Cost Requiring Funding			**$538**

Appendix C: Alpha Testing of UI



So, what did you think of our UI? 19:06

You
So, what did you think of our UI?

Overall, I thought it was easy to read and follow.

When I heard about premise - predicting the number of emergencies across Toronto, I expected a singular, basic interface highlighting emergencies in the city. On the contrary, however, I saw various values in different sections of Toronto and was able to immediately understand what it meant. I really like the UI and I think it has a lot of potential! 19:12

Appendix D: Processing Speed Calculation

Training on 1 year's worth of data takes approximately 2 hours:
Let time taken for 1 day = x seconds
So, we have, 1:365::x:7200
Thus, x = 7200/365 = 19.726 seconds (approximately)