Les Cahiers du GERAD ISSN: 0711–2440

Comptes rendus du 14e atelier de résolution de problèmes industriels de Montréal, 13–17 mai 2024

Proceedings of the 14th Montréal Industrial Problem Solving Workshop, May 13–17, 2024

Odile Marcotte, éditrice

G–2024–76

Novembre 2024

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2024
– Bibliothèque et Archives Canada, 2024

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2024
– Library and Archives Canada, 2024

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Préface

Le CRM, IVADO et le GERAD organisèrent conjointement le Quatorzième atelier de résolution de problèmes industriels de Montréal, qui eut lieu du 13 au 17 mai 2024 à HEC Montréal. Les organisateurs en sont très reconnaissants à HEC Montréal, qui est maintenant un des partenaires académiques du CRM. J'aimerais aussi remercier Jean-François Plante et Juliana Schulz, coprésidents du comité scientifique de l'atelier, Janosch Ortmann, Sylvain Perron, Samuel Perreault, Mike Lindstrom et Gilles Caporossi, coordonnateurs d'équipes, et Dany Plourde et Mariam Tagmouti, conseillers chez IVADO. Finalement j'aimerais souligner la contribution exceptionnelle du CRM et du GERAD à l'organisation matérielle de l'atelier, qui fut assurée par Sakina Benhima, Marie Perreault, Marilyne Lavoie et Karine Hébert.

Odile Marcotte
Conseillère spéciale aux partenariats, CRM
Professeure associée, UQAM et membre associé, GERAD

Foreword

The CRM, IVADO, and GERAD jointly organized the Fourteenth Montreal Industrial Problem Solving Workshop, which took place on May 13-17, 2024, at HEC Montréal. The organizers are grateful to HEC Montréal, which is now a CRM academic partner, for the use of its premises. I would like to thank Jean-François Plante and Juliana Schulz, cochairs of the workshop Scientific Committee; Janosch Ortmann, Sylvain Perron, Samuel Perreault, Mike Lindstrom, and Gilles Caporossi, team coordinators; and Dany Plourde and Mariam Tagmouti, advisors at IVADO. Finally I wish to highlight the contribution of the CRM and GERAD to the workshop logistics, especially the exceptional work of Sakina Benhima, Marie Perreault, Marilyne Lavoie, and Karine Hébert.

Odile Marcotte
Special Advisor, Partnerships, CRM
Adjunct Professor, UQAM and Associate Member, GERAD

CENTRE DE RECHERCHES MATHÉMATIQUES
IVADO
GERAD
GROUPE D'ÉTUDES ET DE RECHERCHE EN ANALYSE DES DÉCISIONS

Contents

1 Air Canada: Contact centre staffing forecasting

Samah Abdalrhaman [a, d]
Ali Barooni [b, d]
Pedram Peiroasfia [c, d]
Samad Farih [e]
Vivian Chan [e]
Sebastian Cosgrove [e]
Janosch Ortmann [a, d, f]

[a] *UQAM*
[b] *Polytechnique Montréal*
[c] *HEC Montréal*
[d] *GERAD*
[e] *Air Canada*
[f] *CRM*

November 2024
Les Cahiers du GERAD

Abstract: *This report summarizes the findings of the team working on Problem 1 of the 2024 CRM industry workshop (IPSW), proposed by Air Canada. We study the problem of scheduling staff for the customer service centres. Our main contributions are a supervised machine learning model that predicts the number of incoming requests and an optimization model that suggests a staff schedule based on the expected number of requests.*

1.1 Introduction

Founded in 1937 as Trans-Canada Air Lines, Air Canada is the largest airline in the country. Headquartered in Montréal, Québec, the airline operates a vast network of domestic and international routes, serving over 200 destinations across six continents.

Air Canada's cargo operations have been an integral part of its business since its early days. The cargo division, known as Air Canada Cargo since 1977, has evolved significantly over the decades, expanding its capabilities and reach. By 2021 Air Canada Cargo was Canada's largest provider of air cargo transportation services as measured by cargo capacity, with a presence in over 50 countries and hubs in Montréal, Toronto, Vancouver, Chicago, London, and Frankfurt.

Initially focused on domestic routes, Air Canada Cargo quickly grew to include international markets, leveraging the airline's extensive route network. During the COVID-19 pandemic, the cargo division played a crucial role in transporting essential goods, including medical supplies and vaccines, demonstrating its reliability and adaptability. Since 2021 a fleet of eight dedicated cargo aircraft (767F) has been providing extra capacity.

The current project concerns scheduling of customer service centre staff for Air Canada Cargo.

The report is structured as follows:

- In Section 1.2 we give a detailed context and problem statement, including an overview of the available data;
- Section 1.3 consists of a descriptive analysis of that data;
- Section 1.4 outlines our predictive and prescriptive methodology;
- The results of our analysis are presented in Section 1.5;
- Section 1.6 is the conclusion of our report.

1.2 Problem description

Air Canada Cargo Customer Service Centres handle customer service inquiries for customers with freight originating from, transiting through, or departing from Canada. We offer service through telephone and email and are reviewing alternative contact channels such as chat. Outside of standard customer service inquiries, the service centres handle new cargo bookings and speciality bookings such as Active Containers and Live Animals (horses, dogs, cats etc.). The service centres in Toronto (YYZ), Montréal (YUL), and Moncton (YQM) are open from 0600 – 2200 EST, seven days a week and offer bilingual service to our customers.

With the ability to bid schedules only once per year, we need to forecast contacts and generate schedules that are linked across the three Customer Service Centres. Currently Air Canada relies on Excel spreadsheets and "heat mapping" to determine the heaviest contact volumes of the day and when staffing should be scheduled. This is only based on historical data, however, and takes into account neither language requirements (English or French) nor future expected freight volumes.

The objective is to create a model that could estimate required staffing based on contact volumes, taking into account our service level requirements and current staffing levels (number of total staff). This model should be based on an average estimated weekly.

1.2.1 Goals of the project

Forecast the number of required staff: Based on the available data, we need to identify the total staff demand for each hour. This will allow us to forecast the number of staff required per hour for the next 365 days. The forecasting horizon is crucial, as we must provide the schedules for staff at the beginning of the year.

Optimize the schedule based on the Forecast: The goal is to minimize the number of staff required while reducing waiting times and meeting all demands as efficiently as possible. By achieving this, we can enhance customer satisfaction and decrease the costs associated with overstaffing.

1.2.2 Available data

Together with the challenge, Air Canada provided four data sets. The first, *VoiceCallComplete*, contains information about all the requests (calls and emails) received by the customer service centre. Some of these contain a Case reference number, which means that the request led to a case being opened. All of the cases are collected in the *AllCases* file, merged from six individual cases files. To benchmark against the current schedule, we were provided with the *AWD-Connect* data set. Finally we also received a *AirwaysBills* data set containing all data about all the airway bills (cargo bookings) originating in Canada between March 2023 and April 2024.

A more detailed description of the data sets, including the attributes, can be found in Appendix 1.A.

1.3 Data presentation

In this section, we briefly present the data that was made available to us and discuss the pre-processing steps that we considered necessary for further analysis.

1. **VoiceCallComplete_.csv:** Initially, the start and end date/time columns for the calls were modified to concatenate the date and time into two separate columns. This allowed for the calculation of the call duration by subtracting the start date/time from the end date/time and converting the result into minutes.

   Certain columns such as 'ToPhoneNumber', 'CallDisposition', 'CallType', and 'From Phone # (area Code)' were deemed non-essential for our preliminary modelling and analysis and therefore we excluded them.

   Following consultations with representatives of Air Canada Cargo, an upper bound of 60 minutes was set for the call durations to enhance the robustness of the model by focusing on more representative cases rather than outliers. This upper bound impacted only 0.4% of the data. Also records with call durations of less than 30 seconds, constituting 10.4% of the remaining data, were removed as these short durations were insufficient for substantive inquiries.

   The call centre operates from 6 AM to 10 PM. Interestingly, 15% of the calls were received during midnight hours when no staff was scheduled to be available. Based on advice from the Air Canada team, these calls were excluded from the analysis. The distributions of the call duration during working hours and midnight were found to be similar, suggesting the validity of these midnight calls despite their exclusion.

   The distribution of call durations is illustrated in Figure 1.1.

   After these preprocessing steps, 55% of the raw data were retained.

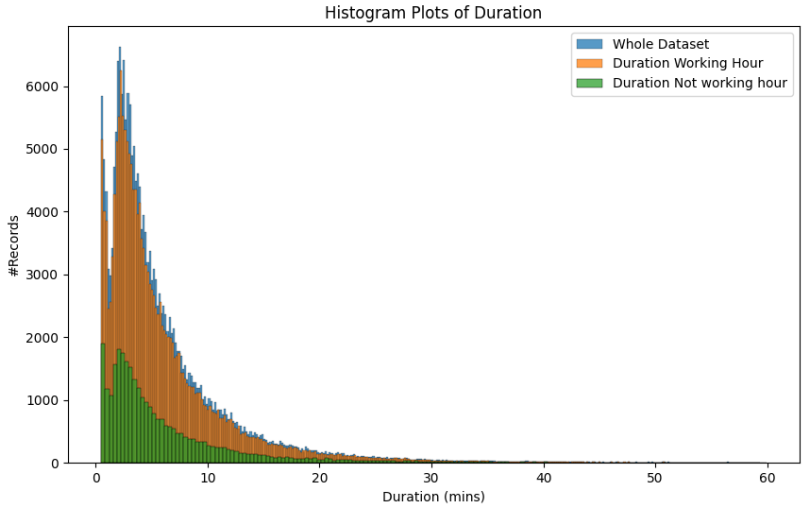


Figure 1.1: Distribution of call durations.

The dataset also contained some calls that did not lead to a reference case number. It was explained to us by the Air Canada team that customer service agents are encouraged to do so in cases where there was only a short enquiry that the agent was able to deal with immediately. For example, a customer might ask for the status of one of their shipments. In those cases, logging a new case could take up more time than answering the query itself, so in order to save time agents do not log a case number.

The language of the calls was also extracted to determine the need for bilingual staff. Further derived features included the hour of the day, calendar day, month of the year, and weekday from the Call Start date/time column.

2. **all_cases.csv:** In this dataset, only closed cases are considered in order to obtain better estimates of their duration and other attributes. Removing the cases that are not closed reduces the dataset by 0.27%. Extra columns such as 'Closed Date FORMATED', 'Status', 'Issue_Type__c', 'Issue__c', and 'IsClosed' are then dropped. Note that only one of the 'Closed Date FORMATED' columns is retained as they are duplicates.

   The 'Origin' field indicates the type of communication. If the origin is 'Transferred' or 'Manual', it is categorized as 'Phone'. If the origin is 'Email', it remains as 'Email'. Cases with the origin 'Proactive' are removed from the dataset.

   As in the case of the previous dataset, the creation and closing dates and times are concatenated into two separate columns. Approximately 94% of the demand is through email, and the remaining 6% is through phone communication.

   In addition some case references from the previous dataset are a subset of cases in the all_cases dataset. After data cleaning, 96% of these references exist in the all_cases dataset.

   The case durations are then calculated in minutes. Interestingly some cases have negative durations, which are displayed in Figure 1.2. A lower bound is set to remove cases with durations less than 30 seconds, which constitute 10% of the dataset. An upper bound is not set as long durations may still be valid due to multiple related emails for a single case.

   Furthermore cases in languages other than English or French (e.g., German) are treated as English after consulting with representatives. This results in approximately 94% of cases being in English and the rest in French. Only data from 2023 is considered to avoid incomplete data cycles for 2024.

	Id	CaseNumber	RecordTypeId	Origin	Language	Time Created	Time Closed	Duration
22745	500Kb00001n3cgYIAQ	868811	NaN	Email	en_US	2024-03-10 03:24:12	2024-03-10 03:18:42	-5.500000
22746	500Kb00001n3cgdIAA	868812	NaN	Phone	en_US	2024-03-10 03:26:03	2024-03-10 02:26:12	-59.850000
22750	500Kb00001n3cgxIAA	868816	NaN	Email	en_US	2024-03-10 03:44:08	2024-03-10 03:12:39	-31.483333
22755	500Kb00001n3chIIAQ	868821	NaN	Email	en_US	2024-03-10 03:55:34	2024-03-10 02:55:37	-59.950000
78612	5001S00001hsOJuQAM	318774	NaN	Email	en_US	2023-03-12 03:19:38	2023-03-12 02:49:18	-30.333333
38623	5001S00001lYKj9QAG	365916	0121S0000001d8zQAA	Email	en_US	2023-05-11 15:39:37	2023-05-11 15:39:37	0.000000

Figure 1.2: Instances of negative case durations.

New columns for the hour of the day, calendar day, month of the year, and weekday are derived from the case creation date/time column. This processed data is then concatenated with the cleaned VoiceCallComplete dataset.

Finally, feature engineering is performed to extract the periodic nature of the time series data using sine and cosine functions applied to features such as HourOfDay, MonthOfYear, and Weekday.

1.3.1 Descriptive analysis

Overall, we had access to 15 months of data, from January 2023 to April 2024. In order to avoid bias stemming from the fact that some months (January – April) are represented twice, we restricted our analysis to the period from January 2023 to 2024 (so as to include a full calendar year) when comparing weekly or monthly data.

Our first question concerned the existence of any seasonal trends. The overall peak appears to occur in autumn, with a secondary local spike in March. It is also interesting to note that requests in French almost completely disappear in the summer months, see Figure 1.3.

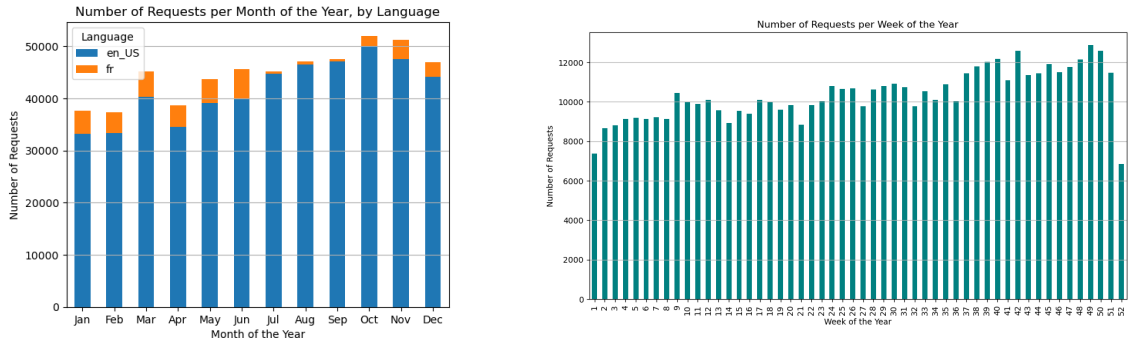


Figure 1.3: Number of requests by month and week.

By looking at the distribution across weekdays (see Figure 1.4), we observed a small number of requests on the weekend. Moreover the greatest variability, but also the highest average number of requests, occurred on Monday, with a slight decline every weekday until Friday.

We also looked at the time of arrival for each request across the day. Times are given in the Eastern Time zone, i.e. the local time of two of the three call centres (YYZ, YUL), and one hour behind the third (YMQ). Nevertheless the business hours at YMQ are the same (6am - 10pm) as the ones in the Eastern Time zone. On the other hand, it is important to note that customers may be sending requests from all parts of Canada, which also includes time zones to the west of Toronto, from Victoria (3 hours behind Eastern Time) to Winnipeg (1 hour behind Eastern Time). This may explain the skew towards the afternoon and evening hours that we have observed, see Figure 1.5.

We were also interested in studying the work time required per request. In order to do so, we analyzed the difference between the time that the request was received (a call was answered or an email

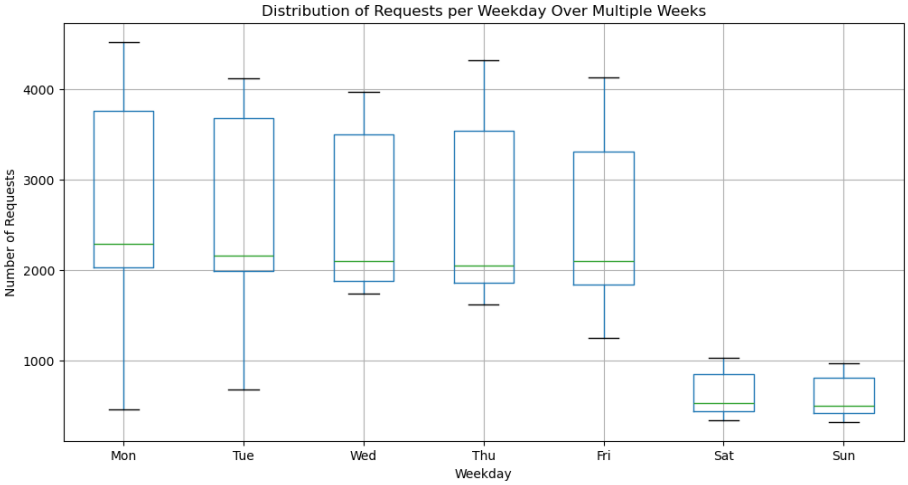


Figure 1.4: Number of requests by weekday.

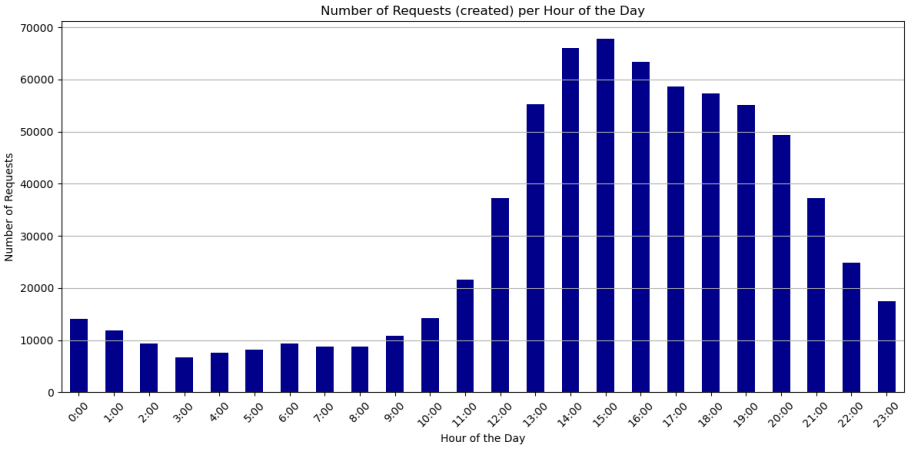


Figure 1.5: Number of requests by hour of the day.

opened by the agent) and the time it was closed (marked as completed) by the agent. The distribution of these times, displayed in Figure 1.6, appeared to follow a bimodal distribution: a positively skewed continuous distribution with a mode around 5 minutes, combined with a second mode close to zero. We concluded from this that there are two types of requests: those that can be completed almost immediately (for example a simple enquiry about the status of a particular shipment) and those which take the agent longer to deal with.

1.4 Methodology

Our methodology follows the classical three-step approach of data analytics: first, a *descriptive* phase allows us to gain an overview of the data set and a general understanding. The highlights of this analysis can be found in the previous section. In the second step, i.e., a *predictive* analysis, we aim to develop a model to predict future call volumes based on past data. Finally the *prescriptive* step consists of developing a decision-making model, using mathematical optimization, that allows us to propose a schedule of how call centre employees should best be deployed in the next year.

The fact that the schedule is only made once a year has had a significant impact on our analysis and the conclusions that we were able to draw. Based on our analysis, a more flexible staffing schedule,

perhaps allowing for extra staff during expected surge periods or taking into account short-term fluctuations, would be beneficial. Given the problem parameters, however, we have not had time to pursue this line of research.

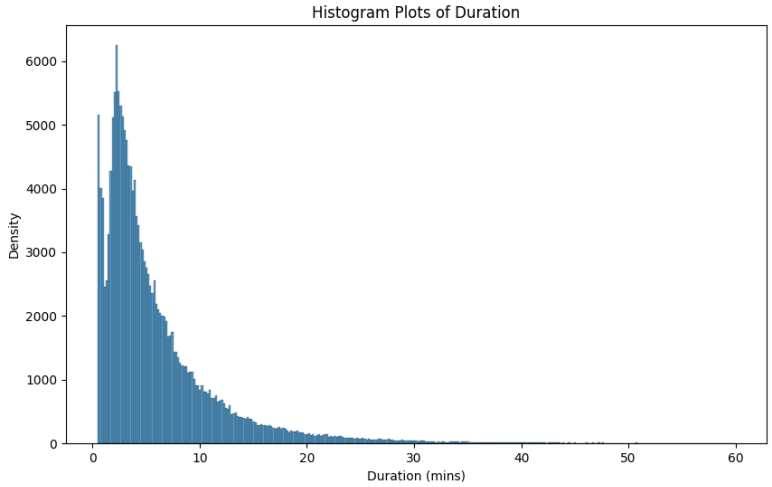


Figure 1.6: Work time required per request.

1.4.1 Forecasting the demand

When working towards a forecasting model for the number of incoming requests, we were faced with two challenges that stemmed from the once-a-year schedule. First of all, we had to predict demand far ahead into the future (more precisely one year ahead).

When predicting time-indexed data, autoregressive models are often used. The term autoregressive means that the model uses previous values in the series to predict future values. A typical example would be predicting tomorrow's stock price: here one would use prices from the past several days to make one's prediction. This is often an effective method because past values have a relationship with future values. Predicting values far into the future with these models is, however, challenging and often unreliable for several reasons.

First of all, the patterns and trends that the model relies on might change over long periods of time, due to seasonal effects but also due to external factors (in the airline industry, extreme examples of this would be 9/11 and more recently the COVID-19 pandemic). Worse still, the errors of the model accumulate over time: while the difference between tomorrow's demand and the demand forecast by the model may be small, the erroneous value is then fed into the prediction for two days into the future, and so on. In this way, prediction errors even for a good model propagate in such a way as to make them essentially useless after a few months.

In order to verify this, we trained a few autocorrelated and recurrent models (such as ARIMA and Xgboost). Our results confirm that, in fact, these models don't work well.

1.4.2 Non-autoregressive models

Instead we adopted a simpler approach that turned out to be more fruitful. We decided to train several supervised machine learning models taking only three predictive variables into account: the calendar month, the day of the week, and the hour of the day. The models were trained to predict the number of requests coming in during a given hour of the day on a given weekday in a given month (see Figure 1.7).

The supervised machine learning algorithms we trained include linear regression, support vector machine, random forest and k-nearest neighbour models. As can be seen in the results section, the random forest model performed the best, predicting the number of requests coming in to a mean absolute error of about 5 minutes.

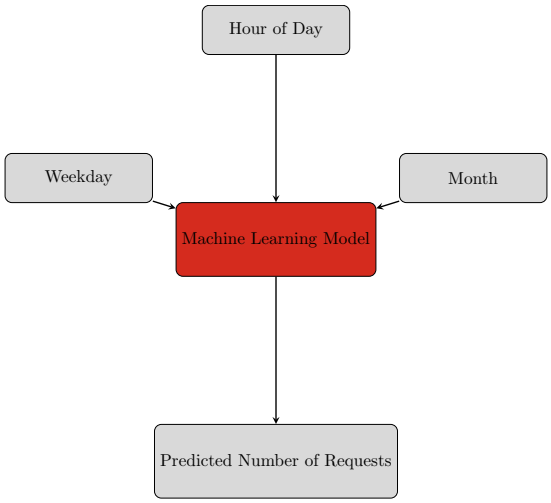


Figure 1.7: Representation of our ML pipeline.

1.4.3 Creating the training and test sets

We encountered a further problem when building training and test sets. In supervised machine learning, these are separated in order to simulate the challenge of predicting unknown data. The *training set* is a subset of historical data used to train the model. This dataset allows the model to learn and understand the underlying patterns, trends, and relationships within the time series data.

The *test set* is a separate subset of historical data that is not used during the training phase. Instead, it is reserved for evaluating the model's performance. After the model has been trained on the training set, it makes predictions on the test set data. These predictions are then compared to the actual values in the test set to assess the model's accuracy and capability to generalize.

When training on temporal data, the train-test split is usually performed by choosing a cut-off date (that constitutes the proxy for the present), with all data points before the cut-off date making up the training set and all data points thereafter being included into the test set. In this way, the procedure of training on past data and evaluating on future data is simulated.

In our context, however, this was not possible, since we only had one full year of data (16 months to be precise). Hence we could not perform a train-test split as explained above while still retaining a full year in the training set. We observed in the descriptive analysis that there are significant seasonal trends and therefore training a forecasting model on a dataset that did not contain at least every month would not work.

In order to address this issue we used the following trick: since the day of the month itself did not appear to have an influence on the demand, we included the data from the 2nd to the 19th of each month into the training set and the data from the 21st to the 29th of each month into the test set. This procedure, illustrated in Figure 1.8, allowed us to create a suitable train-test split while preserving information about each month in each set. We left out a buffer between both sets (i.e., the 1st, 20th, 30th and 31st of each month) in order to minimize information leakage between the training set and the test set.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Figure 1.8: Splitting into training and test sets using the day of the month.

While this procedure of creating training and test sets is sub-optimal, we considered it the best choice given the circumstances. Once more data (spanning several years) is collected, the procedure becomes unnecessary and can be replaced by the approach of taking the first few years as the training set and the last few years as the test set.

1.4.4 Optimizing the schedule

For the prescriptive part of the analysis, we propose a mathematical optimization model that can derive, based on a given request distribution for customer service centre staff, the optimal shift structure. Based on what we were told by Air Canada representatives, we did not model assignment of staff members to shifts, since we understand that staff choose shifts based on seniority within the company. Our model is based on several key assumptions.

1. Each request is processed at the exact moment it is made.
2. Requests are aggregated hourly, and their duration is not considered.
3. On average, it takes 10 minutes to respond to a request, whether it is a call or an email.
4. There is no difference in length of time between responding to English or French calls.
5. The primary focus of our modelling is the minimization of the total cost, rather than minimizing the waiting time, which should ideally be our main objective.
6. Staff members split into *anglophones* who can deal with English-language requests only and *bilinguals* who can deal with either English- or French-language requests.

Due to time constraints during the workshop, we were not able to build a pipeline that feeds the predicted demand (obtained during the predictive step) into the optimization model. Instead the latter takes as input the true number of requests from 2023. In other words, we work on the hypothesis of having perfect information. While this is clearly not realistic, it nevertheless provides a good benchmark for the optimization model. A future step (see also the conclusion) should consist of combining the predictive and prescriptive models into a single pipeline.

A further simplifying hypothesis is that each request (call or email) takes exactly the average time of 10 minutes. In other words we assume that each staff member can handle exactly 6 requests per hour.

For modelling purposes, we calculate two pivot tables: the total labour demand and the English labour demand. The labour demand is calculated as the ceiling of the count of demands (both email and phone) multiplied by the response time and divided by 60. An instance of the pivot tables is displayed in Figure 1.9, which pertains to the total demand. For example, the total labour demand on a Sunday from 10 AM to 11 AM is 2 staff members.

The optimization model (whose precise mathematical formulation can be found in Appendix 1.B) is formulated as follows: the objective function to be minimized is the total wage cost, based on an hourly wage of $20, which yields a cost of $160 for a full-time (FT, 8 hour) and $120 for a part-time (PT, 6 hour) shift. In order to avoid the model calling for only bilingual staff (since these can handle all calls and therefore give more flexibility), we added an additional cost of $40 per FT shift and $20 per PT shift assigned to a bilingual employee. Thus a FT shift of a bilingual employee is assumed to cost $200 and a PT shift of a bilingual employee is assumed to cost $140.

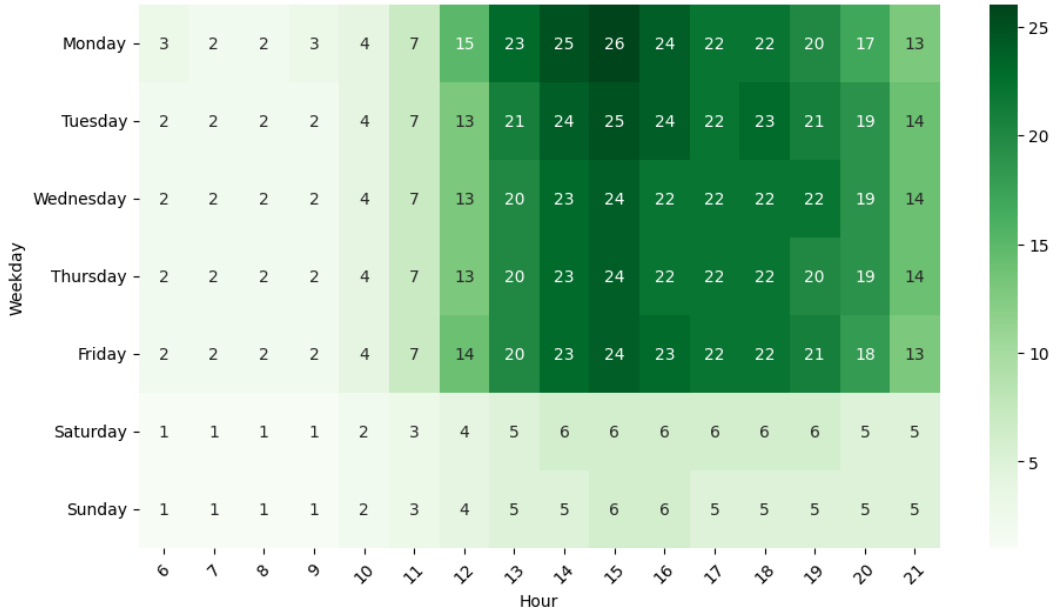


Figure 1.9: Heatmap of total labor demand.

Because of the operating times of 6am to 10pm, PT shifts can start at any time between 6am and 4pm and FT shifts at any time between 6am and 2pm. In this model, we did not account for any breaks.

The decision variables of the optimization model are the number of shifts scheduled at any given hour in a given day of the week (thus yielding a weekly shift) for FT and PT, anglophone and bilingual employees. The goal of the model is to find an optimal configuration of these decision variables, subject to the constraint of satisfying demand: see Appendix 1.B for the mathematical formulation of our prescriptive model.

1.5 Results

1.5.1 Results of the autoregressive time series forecasting

As expected (see the methodology section), autoregressive forecasting models did not produce useful predictions: see Figure 1.10 below for the specific example of an ARIMA forecast.

1.5.2 Results of ML-based forecasting

In contrast to the autoregressive models, a basic supervised machine learning model using the hour, the weekday, and the month (recall Figure 1.7) has performed reasonably well.

In total we trained four models. The mean absolute error (MAE) on training and test sets are displayed in Table 1.1 below. Given that the average number of requests per hour is 27.6, these errors, especially for the best-performing random forest model (MAE of 6.4 on the test set), are quite reasonable. A better performance could potentially be achieved with data spanning more than one year.

Going beyond the mean performance of the predictive model, we have also analyzed the precise error distribution, in order to test whether the model's performance is reliable overall. Figure 1.11 shows that the residuals (differences between predicted and observed values) cluster closely around zero, with few outliers.

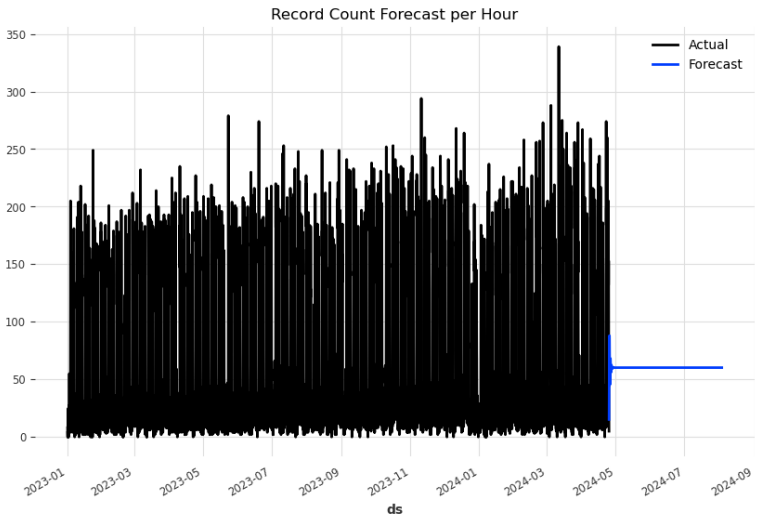


Figure 1.10: ARIMA time-series forecast.

Table 1.1: Performance of the four supervised ML models.

Model	MAE (train)	MAE (test)
Random forest	4.9	6.4
k-nearest neighbour	5.0	7.0
support-vector regression	7.2	7.5
linear regression	14.8	14.8

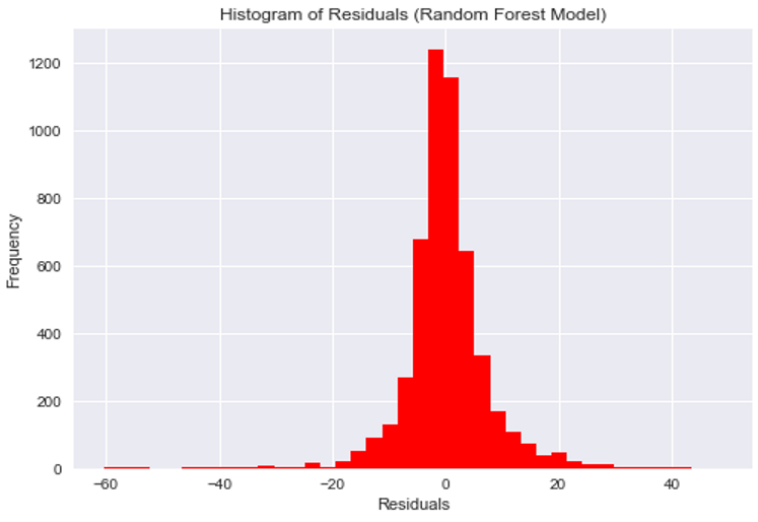


Figure 1.11: Distribution of residuals for the random forest model.

By plotting the predictions on the test set against the actually observed values, we can see that these points lie on a narrow band around the diagonal line, once more confirming the robustness of the model's performance: see Figure 1.12.

In summary we conclude that, even though it is simple, our predictive model achieves a reasonably accurate prediction of future demand and maintains this accuracy consistently. Without another set of new data, we do not believe that changing the model is the most promising avenue for obtaining a higher level of accuracy.

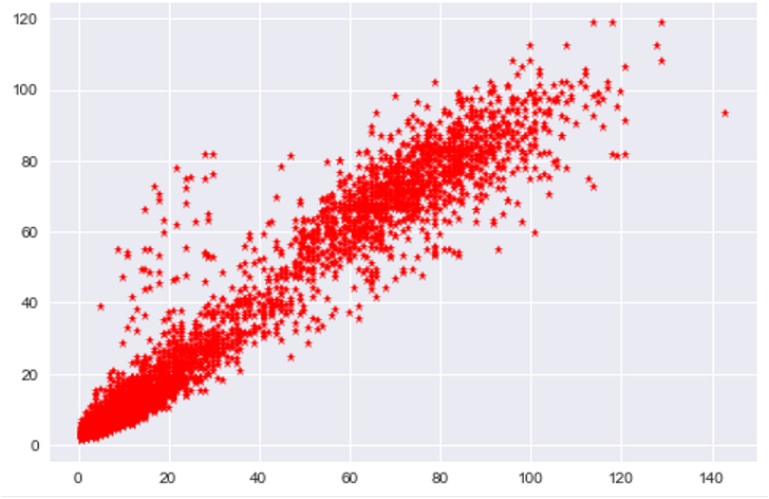


Figure 1.12: Distribution of residuals for the random forest model.

1.5.3 Staff scheduling

We now examine the proposed scheduling results. In the scheduling for Monday (see Figure 1.13), the description of the columns is as follows:

1. **Hour**: The start time of a shift. For example, "hour = 7" means the shift starts at 7 AM. Note that we cannot have any shift starting from 5 PM onwards as the staff would not complete their optimal number of working hours (6 hours for part-time or 8 hours for full-time staff);
2. **Total Demand**: The total labour demand required in the office;
3. **Total in Office**: The total number of people that would be available in the office as a result of scheduling optimization;
4. **English FT Start**: The total number of anglophone FT staff starting their shift at that hour;
5. **English PT Start**: The total number of anglophone PT staff starting their shift at that hour;
6. **Bilingual FT Start**: The total number of bilingual (fluent in both English and French) FT staff starting their shift at that hour;
7. **Bilingual PT Start**: The total number of bilingual PT staff starting their shift at that hour.

Hour	Total Demand	Total in Office	English FT Start	English PT Start	Bilingual FT Start	Bilingual PT Start
6	3	3			3	
7	2	1				
8	2	1				
9	3	1				
10	4	8		1		4
11	7	8				
12	15	15			3	4
13	23	23		1	4	3
14	25	26			6	
15	26	26				
16	24	28				7
17	22	28				
18	22	24				
19	20	20				
20	17	17				
21	13	13				

Figure 1.13: Proposed staff scheduling for Monday.

For weekends, we have a similar scheduling with a significantly lower labour demand, as shown in Figure 1.14.

Hour	Total Demand	Total in Office	English FT Start	English PT Start	Bilingual FT Start	Bilingual PT Start
6	1	1			1	
7	1	1				
8	1	1				
9	1	1				
10	2	1				1
11	3	1				2
12	4	4				
13	5	5			1	
14	6	6			2	
15	6	6				
16	6	8				3
17	6	6				
18	6	6				
19	6	6				
20	5	6				
21	5	5				

Figure 1.14: Proposed staff scheduling for Sunday.

While the result of hiring both full-time and part-time staff might not seem entirely realistic, this can certainly be improved with further exploration and by including more realistic constraints into the model. The current model serves as a foundational approach to optimizing staff scheduling by balancing costs and meeting labour demands.

1.6 Conclusion

Overall we have seen that even a relatively simple model with only three predictors can predict the demand quite well. We have had to make choices and restrict our training due to the availability of only one year of data (given that the schedule must be produced for a full year). Over time Air Canada will have access to more data, which will improve the quality and reliability of the trained model and also allow for more robust testing.

The optimization model we built should be viewed as a work in progress since it was subject to a number of simplifying assumptions. For simplicity, the current optimization model takes the point of view of creating the customer service centre shifts from scratch. That is, we look to minimize staff cost given a fixed objective of responding to each request within the hour. Given that there is an ongoing operation, this should be reversed, i.e., the model should aim to optimize coverage and service levels given staff levels. Doing so will also iron out two issues we encountered, namely the unrealistic ratio of part-time and bilingual employees (compared to full-time and anglophone staff members, respectively). The former are favoured by the model because they lead to more flexibility than the latter.

Overall we believe that our methodology has led to a reasonable prototype that delivers promising results.

Thinking beyond the workshop, we suggest the following next steps. First the prediction and optimization models need to be integrated. In order to do so, a formulation of the optimization model that takes into account the uncertainty of the prediction model is required, for example by considering stochastic, robust, or chance-constrained programming.

Next the overall model should be evaluated on a larger data set. Naturally, this requires waiting for more data to come in. The simplifying assumptions mentioned above should also be removed.

Appendix

1.A Precise description of the data sets

1.A.1 VoiceCallComplete

This dataset contains information about all the calls received by the call centre. The calls might be relevant or irrelevant, including inquiries unrelated to cargo services. The columns of this dataset are described as follows:

1. **Id:** The unique identifier of the call;
2. **Call start date:** The date the call started;
3. **Call start time:** The time the call started;
4. **Call end time:** The time the call ended;
5. **From Phone # (area Code):** The area code from which the customer called;
6. **ToPhoneNumber:** Various phone numbers available in the call centre to answer the call;
7. **CallDisposition:** The disposition of the call;
8. **CallType:** The type of the call, which can be inbound, outbound, callback, or transfer;
9. **Reference_Case_Number__c:** The case number if the call led to a case creation;
10. **Voice_Call_Language__c:** The language used during the call.

1.A.2 all_cases

This dataset is a merge of all CASES datasets from 1 to 6, facilitating easier data loading in Python. The columns are described as follows:

1. **Id:** The unique identifier of the case;
2. **CaseNumber:** The case number for agent follow-up;
3. **RecordTypeId: (Unknown)**;
4. **Status:** The status of the case, whether resolved or not;
5. **Origin:** The method of customer contact;
6. **Language:** The language used for communication between the agent and the customer;
7. **IsClosed:** Whether the case is closed or not;
8. **Created Date FORMATED:** The date the case was created;
9. **Time Created:** The time the case was created;
10. **Closed Date FORMATED:** The date the case was closed;
11. **Time Closed:** The time the case was closed;
12. **Issue_Type__c:** The type of issue that led to the case creation;
13. **Issue__c:** The details of the issue;
14. **Closed Date FORMATED:** The time the case was created. (Note: this column and column 9 refer to the same concept and are exactly the same, except the first occurrence has one extra space.)

1.A.3 AWB

Summary information about all airway bills (cargo bookings) originating in Canada. The columns are:

1. Year;
2. Month;

3. Destination country name;
4. AWB count (number of airway bills);
5. e-AWB count (number of e-airway bills).

1.A.4 AWSConnectData

Monthly aggregate data about each agent's interactions with calls from March 2023 to April 2024. The data set includes features such as the agent's answer rate, average talk time, and online time. We did not use this data set.

1.B Mathematical statement of the optimization model

In this section we give the details of the mathematical optimization model that we used to derive the schedule presented above.

1.B.1 Sets

- H: Set of all hours during the operating day, $H = \{6, 7, \ldots, 21\}$.
- T_{FT}: Set of possible starting times for full-time shifts, $T_{FT} = \{6, 7, \ldots, 14\}$.
- T_{PT}: Set of possible starting times for part-time shifts, $T_{PT} = \{6, 7, \ldots, 16\}$.
- d: Day of the week, here $d = 3$.

1.B.2 Parameters

- D_t: Total demand for agents at hour $t \in H$.
- B_t: Demand for bilingual agents at hour $t \in H$.
- $c_x = 160$: Cost of an English-only full-time (FT) staff member.
- $c_y = 120$: Cost of an English-only part-time (PT) staff member.
- $c_z = 200$: Cost of a bilingual full-time (FT) staff member.
- $c_w = 140$: Cost of a bilingual part-time (PT) staff member.

It is worth mentioning that the costs $160, 120$, etc. are set to achieve a balance in selecting the staff strategically and are completely arbitrary.

1.B.3 Decision variables

- x_t: Integer variable indicating the number of English-only full-time staff members starting their shift at time $t \in T_{FT}$.
- y_t: Integer variable indicating the number of English-only part-time staff members starting their shift at time $t \in T_{PT}$.
- z_t: Integer variable indicating the number of bilingual full-time staff members starting their shift at time $t \in T_{FT}$.
- w_t: Integer variable indicating the number of bilingual part-time staff members starting their shift at time $t \in T_{PT}$.

1.B.4 Objective function

Minimize the total wage cost:

$$\text{Minimize} \sum_{t \in T_{FT}} (160 \cdot x_t + 200 \cdot z_t) + \sum_{t \in T_{PT}} (120 \cdot y_t + 140 \cdot w_t)$$

1.B.5 Constraints

Total Demand Constraints: Ensure that the total number of agents scheduled at each hour meets the total demand.

$$\sum_{i=0}^{7}(x_{t-i} + z_{t-i}) + \sum_{i=0}^{5}(y_{t-i} + w_{t-i}) \geq D_t \quad \forall t \in H$$

Bilingual Demand Constraints: Ensure that the number of bilingual agents scheduled at each hour meets the bilingual demand.

$$\sum_{i=0}^{7} z_{t-i} + \sum_{i=0}^{5} w_{t-i} \geq B_t \quad \forall t \in H$$

Non-negativity Constraints: Ensure that all the decision variables are non-negative.

$$\begin{aligned} x_t &\geq 0 \quad \forall t \in T_{FT} \\ y_t &\geq 0 \quad \forall t \in T_{PT} \\ z_t &\geq 0 \quad \forall t \in T_{FT} \\ w_t &\geq 0 \quad \forall t \in T_{PT} \end{aligned}$$

2 Air Canada: Dynamic spot rate optimization – A simulation model for revenue enhancement in Air Canada cargo pricing

Faramarz Farhangian [a]
Golshid Aflaki [b, c]
Sylvain Perron [b, c]
Ernest Tafolong [b, c]
Georgios Farfaras [d]
Flora Gao [d]
Vasanth Ramkumar [d]
Janik Gagne [d]

[a] *École de Technologie Supérieure, UQAM*
[b] *HEC Montréal*
[c] *GERAD*
[d] *Air Canada*

November 2024
Les Cahiers du GERAD

Abstract: ***Objective:*** *This study aims to optimize air cargo spot rates for a single-leg flight by dynamically adjusting pricing based on variables such as adjustment factors, load factor, and days to departure. The objective is to maximize revenue by identifying the optimal adjustment factor for specific conditions.* ***Methods:*** *Historical booking data from 2018 to 2024 was analyzed, focusing on standard cargo bookings in the North America–Europe market from July 2023 to April 2024. Monte Carlo simulations were conducted, generating 1,000 booking requests per iteration over 500 iterations, with varying adjustment factors. A logistic regression model was trained using 70% of the data and validated on the remaining 30% to predict booking acceptance probabilities. Revenue was compared across different adjustment factors to identify the optimal value.* ***Results and Conclusion:*** *The simulation results indicate that reducing the adjustment factor increases revenue, with the highest average revenue achieved at a 20% reduction. Lower adjustment factors were shown to improve revenue under specific load factor and time-to-departure conditions, highlighting the importance of dynamic pricing strategies tailored to operational constraints.*

2.1 Introduction

Air cargo is a vital pillar of global trade, accounting for just 1% of shipment volume yet contributing around 35% of the overall trade value, highlighting its role in the transport of high-value goods [1]. The industry has experienced consistent growth, fueled by the expansion of the aviation sector and the development of global air networks. This growth is expected to continue at an annual rate of 4.1% over the next two decades, solidifying air cargo's role in enabling efficient, time-sensitive logistics for international markets [2].

Revenue management (RM) has optimized airline revenue, increasing earnings by 4-5% [3]. In the air cargo industry, RM is essential for balancing capacity utilization and customer demand to optimize revenue. Unlike passenger airlines, air cargo operations face unique challenges, such as multi-dimensional capacity (weight, volume), booking uncertainty [4], fluctuating demand, and the perishable nature of cargo space. Traditional RM techniques often fail to address these complexities due to their static nature, highlighting the need for more adaptable solutions. Dynamic pricing and spot rate optimization, facilitated by advanced technologies like machine learning (ML), deep learning, reinforcement learning (RL), and optimization algorithms, allow airlines to adapt prices in real-time based on current market conditions [5]. By adjusting cargo space prices according to demand, competitor pricing, and broader market trends, airlines can reduce prices during low-demand periods to attract customers and increase rates in peak times to maximize revenue. This flexibility not only optimizes revenue but also enhances operational efficiency, competitiveness, and customer satisfaction. This report addresses the gap in air cargo RM by applying dynamic pricing to maximize revenue.

Air Canada Cargo, Canada's largest air cargo provider by capacity, is an award-winning leader in air cargo transportation, connecting over 450 cities across six continents through a network of direct flights and partnerships. With a presence in over 50 countries and strategically located hubs in cities such as Montréal, Toronto, Vancouver, Chicago, London, and Frankfurt, Air Canada Cargo plays a vital role in global logistics.

To enhance customer experience and maintain a competitive edge, Air Canada Cargo has made significant investments in data analytics, APIs, and artificial intelligence. One of the key initiatives in this strategy is the development of an intuitive spot rate recommendation tool. This tool is designed to improve pricing accuracy and transparency by providing automated, explainable, and consistent rate recommendations, addressing the dynamic nature of air cargo pricing where spot rates are influenced by factors such as demand, capacity, and time-to-departure. Traditionally, setting optimal spot rates has relied on a combination of base rates, revenue targets, and a rate change matrix that incorporates key variables like Days to Departure, Current Load Factor, and Forecasted Load Factor. These factors collectively determine the "change factor" applied to the base rate. This optimization process, however,

has been largely manual, which limits its efficiency and scalability. Given the highly competitive and time-sensitive nature of air cargo services, there is a pressing need for a more automated, data-driven approach to rate optimization that can dynamically respond to real-time market conditions.

This report presents the development of a dynamic Spot Rate Change Factor Simulator designed to address this gap by automating the testing of various change factor combinations and assessing their impact on revenue. Leveraging historical booking data, AWB (Air Waybill) data, and market demand data, the simulator aims to streamline the process, enabling Air Canada Cargo to identify near-optimal pricing strategies swiftly. This tool will improve decision-making in rate adjustment and align with Air Canada Cargo's overarching goal of enhancing revenue through a data-informed strategy.

The rest of this paper is organized as follows. In Section 2.2, we review the relevant literature. In Section 2.3, we provide the methodology. Section 2.5 is a report on the numerical experiments. Section 2.6 contains the conclusion and recommendations of this study.

2.2 Literature review

Research on air cargo operations largely addresses areas such as flight scheduling, fleet routing, and revenue management (RM) problems. In this regard, some studies provide a comprehensive review, noting that air cargo scheduling and routing involve complex decisions on route selection, cargo assignments, and crew scheduling [6]. Several studies focus on optimizing operational decisions within these areas: they propose a planning framework that integrates airport choice, fleet routing, and scheduling for profit optimization [7, 8], while some others develope a dynamic programming model for routing air cargo based on real-time data [9].

Given our focus on RM, we specifically review RM-related literature. Though extensively explored in passenger airlines [3, 10, 11], RM applications in air cargo are less studied. Early works contrasted RM practices in passenger versus cargo industries, outlining key distinctions and complexities in air cargo RM [12]. The others discussed cargo RM system components and implementation challenges [13], while some studies detailed KLM's RM implementation for cargo [14]. In 2007, some studies expanded on this, highlighting supply-demand complexities in air cargo RM [15]. Research on RM for cargo often uses static formulations [16, 17, 18] or dynamic models with capacity control.

In 2007, one study formulated cargo spot sales as a multi-dimensional Markov decision process (MDP), introducing penalties to handle overbooking and developing heuristics for volume and weight optimization [4]. Building on their work, one study proposed sampling-based heuristics [19], and Hoffmann (2013) introduced a monotone cost heuristic that reduces computational load through simplified control [20]. Other studies applied bid-price control to manage booking acceptance decisions [5, 21]. In network-based RM, a study proposed a linear programming method for capacity optimization under uncertainty [22], while some others compared heuristics to improve upper bounds on capacity utilization [23]. More recent work developed a stochastic gradient algorithm to manage capacity control with variable availability [24].

Unlike these studies, our work focuses on dynamic pricing rather than capacity control, applying it to spot market bookings with uncertain weight and volume. Dynamic pricing, distinct from capacity control, adjusts prices over time. The article [25] reviewed dynamic pricing in RM, while most research addresses single-product scenarios. Some studies explored single-product dynamic pricing, showing near-optimal results with constant pricing [26]. For multi-product settings, they formulated a deterministic model with near-optimal pricing, while other studies proposed dynamic programming-based heuristics for network RM, yielding high expected revenue [26, 27]. Our study differs in its focus on dynamic pricing for multiple air cargo bookings, incorporating continuous and uncertain weight and volume: the problem we address thus presents unique challenges requiring robust pricing strategies.

2.3 Methodology

Air Canada Cargo faces the challenge of optimizing its adjustment spot rate. Manual optimization of spot rates is inefficient and can lead to suboptimal revenue, requiring an automated solution to enhance pricing strategies. The adjustment spot rate serves as a mechanism to modify the fixed cargo rate established in customer contracts. When a customer submits a cargo request, this adjustment rate is offered based on two key variables: the **current load factor** (the percentage of cargo capacity utilized) and the **days to departure** (the number of days remaining until the scheduled flight). The goal is to optimize the adjustment rate table, which specifies adjustment factors for various load factor ranges and day-to-departure intervals, in order to maximize overall revenue. This involves determining the optimal adjustment factors for each combination of load factor group and day-to-departure group, considering the dynamic nature of demand and the need to balance capacity utilization with competitive pricing. By addressing this problem, Air Canada Cargo aims to implement a data-driven strategy that increases revenue while maintaining operational efficiency and customer satisfaction.

Expected LF: AVERAGE							
Days to Departure / Current LF	0 to 1	1 to 2	2 to 3	3 to 4	4 to 5	5 to 10	10 to 15
0% - 30%	0.7	0.7	0.7	0.7	0.85	1	1
30% - 45%	0.7	0.7	0.7	0.7	0.9	1	1
45% - 60%	0.75	0.75	0.84872	0.87404	1	1	1
60% - 75%	1.1	1.04404	1.04106	1.03808	1.0351	1	1
75% - 90%	1.3	1.2468	1.22744	1.20808	1.18872	1.16936	1.1
90% - 100%	1.5	1.4585	1.4168	1.3751	1.3334	1.2917	1.1
>100%	2.74	2.58	2.41	2.25	2.08	1.92	1.75

Figure 2.1: Adjustment rate table showing the expected load factor (LF) averages for various combinations of days to departure and current LF ranges. This table serves as a guide for dynamically adjusting spot rates to optimize revenue based on capacity utilization and booking timelines.

2.3.1 Overview of proposed method

Optimizing the adjustment factors is a complex task due to the high-dimensional space of uncertainty in demand patterns, customer behaviour, and operational constraints. This involves balancing multiple factors, including the dynamic nature of demand, varying customer reservation prices, and operational constraints like cargo capacity. Traditional analytical methods struggle to capture this complexity effectively, as the problem involves a large number of possible combinations of load factor ranges and days-to-departure intervals, each with its own stochastic behaviour.

There are various approaches to address this complexity. We propose using a **Monte Carlo Simulation** to evaluate and optimize the adjustment factors. Monte Carlo simulation is well-suited for this problem because it allows us to model the inherent randomness in customer behaviour, demand fluctuations, and booking patterns. By simulating a wide range of scenarios, Monte Carlo simulation provides robust estimates of average revenue for each specific pair of load factor ranges and days-to-departure intervals. It enables the evaluation of current adjustment factors as well as alternative factor values, allowing us to identify the combination that maximizes revenue.

This data-driven approach leverages the power of simulation to navigate high-dimensional uncertainty and optimize pricing strategies in a dynamic and competitive environment. This involves evaluating the adjustment factor currently in use as well as a range of plausible adjustment factors to identify the one that maximizes revenue. Figure 2.1 displays the matrix that we want to optimize. To begin, we fix a specific combination of load factor range and days-to-departure interval. For the **Average Expected Load Factor**, there are 49 such pairs to analyze, each representing a unique combination of these two variables. For each pair, we proceed as follows:

1. **Data Generation and Strategy:** Generate a random dataset of n synthetic customers requesting cargo services;

2. **Pre-trained Logistic Regression:** Use a pre-trained logistic regression model to predict the probability of customer offer acceptance for the proposed rate. Identify customers who accept the offer and compute the total revenue for this dataset;
3. **Monte Carlo Sampling:** Repeat the above process m times using Monte Carlo Markov Chain (MCMC) simulation, generating new random datasets each time. Calculate the average revenue over these m samples for the fixed load factor range, days-to-departure interval, and adjustment rate;
4. **Adjustment Factor Optimization:** Iterate through a range of reasonable adjustment factors, repeating the process described above. Compare the average revenues obtained for different adjustment factors and identify the adjustment factor that yields the highest average revenue for the fixed load factor and days-to-departure pair.

By applying this methodology to all 49 load factors and days-to-departure pairs, we can determine the optimal adjustment factor for each pair. These optimal factors will then populate the adjustment rate table, ensuring that revenue is maximized across all scenarios. The study considers four key components: **Load Factor intervals** (lf), which is a vector consisting of 7 predefined ranges representing the percentage of cargo capacity utilization; **Days-to-Departure intervals** (dd), another vector of 7 ranges capturing the number of days remaining until the scheduled flight; **Adjustment Factors** (af), a vector of plausible adjustment rates used to modify the base rate dynamically, with the currently used adjustment factor denoted as af_0; and finally, **Revenue** (R), a single value computed as the total revenue generated for a group of customers requesting cargo services within the specified intervals. These components form the foundation for analyzing and optimizing pricing strategies. From lf and dd, we form pairs (l, d), where $l \in lf$ and $d \in dd$. These pairs represent the 49 unique combinations of load factor intervals and days-to-departure intervals for which we want to find the optimized adjustment factor based on the Algorithm 2.1.

Algorithm 2.1 Adjustment Factor Optimization using Monte Carlo Simulation.

Input: lf, dd, af
Output: Optimal AF $\forall(l, d)\ in(L, D)$
for all $(l, d) \in (L, D)$ **do**
  Initialize $R \leftarrow R_0 = 0$
  Initialize $af \leftarrow af_0$
  for all $a \in af$ **do**
    Generate m synthetic datasets (Monte Carlo Sampling):
    for each dataset $i = 1$ to m **do**
      Generate n synthetic customers with random characteristics
      Compute customer acceptance probabilities using the logistic regression model
      Identify customers who accept the offer at adjustment factor a
      Compute total revenue for dataset i
    end for
    Calculate $\bar{R}$
    if $\bar{R} > R$ **then**
      Update $R \leftarrow \bar{R}$
      Update af
    end if
  end for
  Store af for pair (l, d)
end for
Return: Table of optimal adjustment factors for all pairs (l, d)

2.3.2 Monte Carlo Simulation

Monte Carlo Simulation is a computational method used to model systems with inherent uncertainty by simulating a large number of random scenarios. It relies on the law of large numbers to approximate the expected outcome of a process when an analytical solution is infeasible due to high-dimensional uncertainty or dynamic variables. Mathematically, Monte Carlo Simulation involves defining a system or process represented by a function $f(X_1, X_2, \ldots, X_n)$, where $X_1, X_2, \ldots, X_n$ are random variables

with known probability distributions. Random samples $X_1^{(i)}, X_2^{(i)}, \ldots, X_n^{(i)}$ are generated independently for $i = 1, 2, \ldots, N$, where N is the number of simulations. For each simulation, the outcome $Y^{(i)}$ is calculated as

$$Y^{(i)} = f\left(X_1^{(i)}, X_2^{(i)}, \ldots, X_n^{(i)}\right).$$

The expected value of the outcome Y is estimated as the average of all simulated outcomes, i.e., the following holds:

$$\hat{E}[Y] = \frac{1}{N}\sum_{i=1}^{N} Y^{(i)}.$$

As $N \to \infty$, the estimated expected value $\hat{E}[Y]$ converges to the true expected value $E[Y]$. Variance or other statistical metrics can also be computed to evaluate the variability of the outcomes. Monte Carlo Simulation is particularly useful for problems where the complexity of random interactions and high-dimensional spaces makes traditional analytical approaches impractical. By simulating a wide range of scenarios, it provides robust estimates of expected outcomes and insights into system behaviour under uncertainty, making it a powerful tool for optimization and decision-making.

2.3.3 Logistic regression

Logistic regression is a generalized linear model used for both inference and prediction of binary outcomes. In our Monte Carlo simulation, after generating each synthetic dataset, we aim to predict the quote status of each customer (in each row of the dataset). This means determining whether they accept ($QuoteStat = 1$) or reject the offer they received from the contact center. The logistic regression model is based on the available information, including the load factor at the time of the request (LF), the number of days remaining until the departure time ($Days$), the adjustment factor offered on top of their fixed rate (AF), and the package weight ($Weight$). We can formulate the model as follows.

$$log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1.LF + \beta_2.Days + \beta_3.Weight + \beta_4.AF \tag{2.1}$$

Here, $p = P(QuoteStat = 1)$ represents the probability that a customer will accept the offer. By using a preferred threshold, we can assign each customer to either the accepted or rejected group. The choice of the threshold will be influenced by the balance of the quote status variable, as a skewed distribution may require adjusting the threshold to achieve meaningful classifications. The logistic regression in Equation 2.1 is trained on the training set, which constitutes 70% of the original dataset received from Air Canada Cargo, and tested on the rest 30%. The coefficients are estimated using Ordinary Least Squares (OLS), and the regression model is then applied to predict unseen data — specifically, the synthetic data generated in each iteration of the Monte Carlo simulation.

2.4 Dataset

The dataset used in this study is derived from Air Canada Cargo's internal booking system, as illustrated in Figure 2.2. According to Figure 2.2, in the first phase, the system collects data in real-time, capturing essential information about booking requests. This information includes:

Routing Details: These specify the time of departure, origin, and destination;

Shipment Details: This includes the type of cargo being shipped and its specific characteristics;

Customer Information: This refers to the customer account, which contains details about the contract and additional information related to the client.

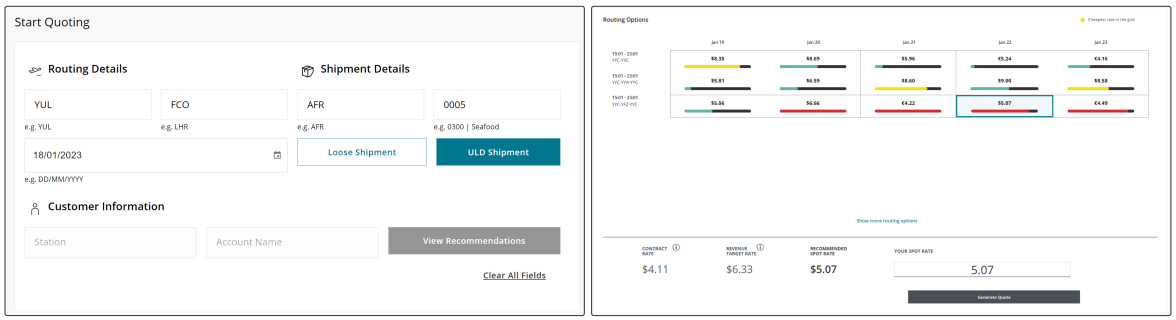


Figure 2.2: Illustration of the internal booking system: Phase 1 (left) collects and processes booking request information, while Phase 2 (right) generates dynamic pricing recommendations based on variables such as load factor (indicated by colors: green for low, yellow for moderate, and red for high), time to departure, routing options, and revenue targets.

After processing the request and collecting all relevant information, the system transitions to Phase 2, where it recommends various options tailored to the booking. The system applies dynamic pricing rules to calculate base rates and adjusted spot rates based on several factors, including time to departure, available routing options, the base contract rate associated with the client, the company's target revenue rates, and, most importantly, the current load factor, which indicates how full the flight is at the time of the request. By integrating these variables, the system ensures that pricing recommendations are both competitive for the client and aligned with the company's revenue optimization objectives. Customer requests are logged with details, and each request's outcome (Accept or Reject) is recorded to track customer behaviour and improve pricing strategies.

2.4.1 Data description

In this study we used historical booking data comprising over 150,000 records collected between 2018 and 2024. The dataset is structured around several key variables, each of which plays a vital role in modelling customer behaviour and optimizing pricing strategies. A detailed description of these variables is provided below.

Days to Departure: The number of days remaining until the flight's scheduled departure at the time of booking. This variable is essential for analyzing how timing impacts pricing and acceptance behaviour.

Routes: Data specifying the origin and destination airports, representing the start and end points of the shipment.

Type of Cargo: The category of the shipment includes classifications such as post, standard, animal, secure, PharmaCare, and others. This variable helps identify the nature of the cargo and its specific handling or pricing requirements.

Chargeable Weight: The weight of the cargo for which the customer is charged. This directly impacts revenue calculations.

Density Change Factor: A factor used to adjust the rate based on the density of the cargo, ensuring optimal space utilization.

Adjustment Factor (Adj_factor): A composite factor reflecting dynamic adjustments to the base rate, influenced by variables such as load factor and days to departure.

Current Load Factor: The percentage of available cargo space already booked at the time of the request. Higher load factors typically result in higher spot rates due to limited availability.

Base Rate: The initial, unadjusted rate for the cargo shipment before any adjustments.

Revenue Target Rate: A target rate set by the company to ensure profitability for a given route and time frame.

Spot Rate: The final adjusted rate offered to the customer, also referred to as the quoted spot rate, is calculated by applying dynamic pricing adjustments to the base rate. These adjustments account for factors such as load factor, which reflects how full the flight is, and cargo density, optimizing pricing based on space utilization. After these adjustments, additional costs, such as handling fees, are incorporated. The final spot rate is determined by selecting the higher value between the revenue target and the adjusted base rate, ensuring alignment with both operational goals and profitability targets.

Quote Status: A binary outcome indicating whether the booking request was accepted or rejected.

2.5 Experiments and results

To simplify the problem and manage complexity, we make the following assumptions and methodological choices. We consider a single-leg flight with fixed weight and volume capacities. Since the behaviour of area markets varies, we restrict the booking data to the North America–Europe market. Additionally, due to policy changes and the atypical booking patterns during the COVID-19 pandemic, we limit the dataset to bookings from July 2023 to April 2024. Furthermore, while there are various cargo types (e.g., standard, secure, PharmaCare), this case study focuses exclusively on standard cargo. Given the computational intensity of optimizing the entire rate matrix, we simplify the problem by focusing on a single cell: bookings with load factors of 0%–30% and 0 to 1 day to departure. For this cell, we simulate 500 iterations, each generating 1,000 booking requests with key variables, including Load Factor (sampled uniformly within 0–30%), Days-to-Departure (binary values of 0 or 1 as a Bernoulli random variable), Chargeable Weight (sampled from a Gamma distribution to reflect right-skewness), and Base Rate (generated using a Normal distribution based on observed data characteristics), using the average expected load factor. The distributions of chargeable weight and base rate are displayed respectively in the density plots in Figure 2.3 and Figure 2.4.

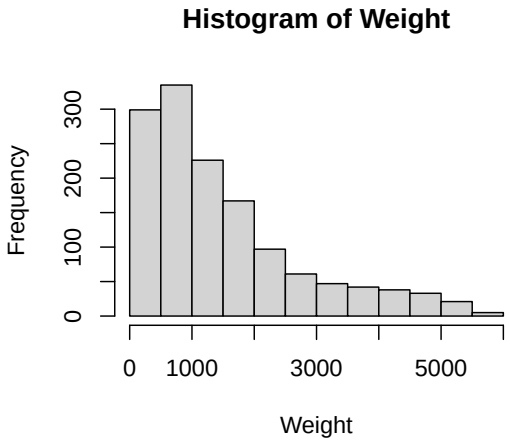


Figure 2.3: The distribution of chargeable weight.

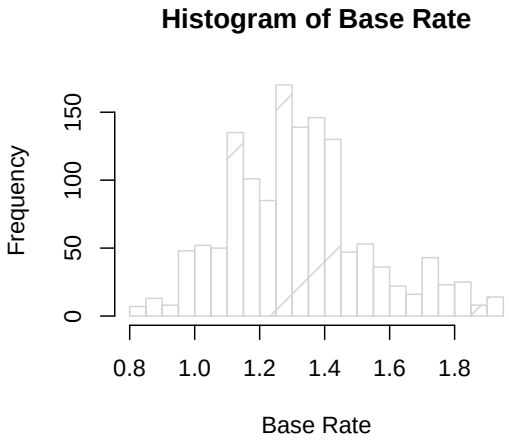


Figure 2.4: The distribution of base rate.

Spot rates are optimized by comparing the company's base value with the ±10% and ±20% ranges to identify the most profitable option. Logistic regression is trained on a 70% training set and a 30% validation set to predict acceptance rates. To address data imbalance, we adjust the acceptance threshold, and accuracy is reported as the primary evaluation metric. Then, by using the pre-trained logistic regression model, we predict the offer acceptance probability ($P(QuoteStatus = 1)$) for each simulated booking and classify requests as "accepted" or "rejected" based on a threshold of 0.3. This approach ensures realistic data simulation and accurate revenue predictions within the Monte Carlo framework. By making these assumptions and narrowing the scope, this study establishes a manageable

framework for developing and evaluating a dynamic pricing model while addressing the complexities of air cargo operations.

2.5.1 Pre-trained logistic regression results

The logistic regression model was developed to predict the binary outcome of customer offer acceptance ($QuoteStatus = 1$) using key variables, including the number of days left until departure ($days\_out$), the base rate for the shipment ($Base\_rate$), the load factor at the time of the quote (LF_at_quote), the chargeable weight of the cargo ($chargeable\_wgt$), and the adjustment factor applied to the base rate (AF). These variables were chosen for their significant influence on customer decisions and pricing dynamics. Here is the logistic regression equation for the model.

$$\log\left(\frac{p}{1-p}\right) = 0.416 - 0.034 \times \text{DD} - 0.579 \times \text{BR} - 0.002 \times \text{LF} - 0.000148 \times \text{CW} - 0.564 \times \text{AF}$$

In the above equation p represents the probability of offer acceptance, and the coefficients indicate the direction and magnitude of the effect of each predictor. The regression coefficients and their significance levels are given in the following table.

Variable	Estimate	Std. Error	z-value	p-value
Intercept	0.416	0.659	0.631	0.528
DD	−0.034	0.047	−0.726	0.468
BR	−0.579	0.371	−1.562	0.118
LF	−0.002	0.004	−0.587	0.557
CW	−0.000148	0.000071	−2.099	0.036*
AF	−0.564	0.446	−1.267	0.205

The logistic regression results indicate that most variables, including DD, BR, and AF, are statistically insignificant ($p > 0.05$), while CW is significant ($p = 0.036$). The negative coefficients suggest that higher rates and adjustments decrease acceptance probability, highlighting the importance of balancing chargeable weight and rate adjustments in order to optimize pricing strategies.

Then the revenue is computed only for customers predicted to accept the offer, using the formula below.

$$Revenue = \text{Base Rate} \times \text{Adjustment Factor} \times \text{Weight}$$

2.5.2 Simulation results

This process is repeated 500 times, generating new datasets for each iteration. The average revenue over these simulations represents the expected revenue for the specified load factor, days-to-departure interval, and adjustment factor. The process is conducted five times with the following adjustment factor values: $AF = 0.7$ (current adjustment factor), $AF = 0.7 + 0.1 \times 0.7 = 0.77$ (10% higher), $AF = 0.7 - 0.1 \times 0.7 = 0.63$ (10% lower), $AF = 0.7 + 0.2 \times 0.7 = 0.84$ (20% higher), and $AF = 0.7 - 0.2 \times 0.7 = 0.56$ (20% lower). By comparing the average revenues computed for each adjustment factor, we identify the optimal adjustment factor that maximizes revenue. For this specific case, the maximum revenue occurs when the adjustment factor is reduced by 20% ($AF = 0.56$), as the base rates are mostly below 1 (see Table 2.1).

The simulation results in Table 2.1 show that reducing the adjustment factor increases total revenue, with the highest average revenue (35, 817.49) achieved when the adjustment factor is reduced by 20%. A 10% reduction also yields a higher revenue (35, 791.96) compared to the current adjustment factor (35, 764.79). Conversely, increasing the adjustment factor by 10% and 20% results in progressively lower revenues (35, 732.87 and 35, 723.04, respectively). These results suggest that lower adjustment factors are more effective in optimizing revenues, especially under the combination of a load factor in the interval 0–30% and a number of days to departure in the interval 0–1.

Table 2.1: Average Revenue using different adjustment factors for the combination of a Load Factor between $0 - 30\%$ and a number of days to departure between $0 - 1$.

Adjustment Factor	Total Rev
Current AF - 20% Current AF	**35817.49**
Current AF - 10% Current AF	35791.96
Current AF	35764.79
Current AF + 10% Current AF	35732,87
Current AF + 20% Current AF	35723.04

2.6 Conclusion and discussion

Dynamic pricing models are essential for air cargo carriers to stay competitive and maximize revenue in today's fast-paced market. By leveraging advanced technologies such as machine learning, deep learning, reinforcement learning, and optimization algorithms, airlines can develop sophisticated pricing strategies that adapt to real-time market conditions. Despite challenges related to data quality, model explainability, and regulatory compliance, the benefits of dynamic pricing in terms of increased revenue, competitiveness, customer satisfaction, and optimized capacity utilization make it a worthwhile investment for air cargo carriers.

This study explored the optimization of air cargo spot rates using historical booking data and simulation techniques. By focusing on a specific case with a load factor between 0–30% and 0–1 days to departure, we identified how adjustment factors influence revenue generation. The results demonstrate that reducing the adjustment factor significantly improves revenue, with a 20% reduction yielding the highest average revenue. Conversely, increasing the adjustment factor results in progressively lower revenues, emphasizing the sensitivity of revenue optimization to precise rate adjustments. Despite these limitations, the findings offer actionable insights for revenue management in the air cargo industry. Lower adjustment factors appear to align better with customer price sensitivity, particularly in scenarios with low load factors and short booking windows. These results suggest that dynamic pricing strategies should emphasize optimizing adjustment factors to balance profitability and customer acceptance rates.

In conclusion, this study provides a foundation for developing dynamic pricing models tailored to air cargo operations. By leveraging predictive analytics and simulation, we demonstrate a data-driven approach to revenue optimization that can be adapted to other market conditions and operational settings. Future work could explore advanced reinforcement learning (RL) techniques to further enhance dynamic pricing strategies. Proximal Policy Optimization (PPO) could be extended with Deep Q-Networks (DQN) or Soft Actor-Critic (SAC) to manage high-dimensional state-action spaces and improve pricing precision. Multi-agent reinforcement learning offers another promising avenue, allowing the simulation of competitive pricing strategies where multiple airlines interact within a shared market environment. Additionally, integrating explainable RL methods can provide transparency in decision-making, ensuring stakeholder trust and regulatory compliance. These advancements in RL have the potential to optimize dynamic pricing strategies, maximize revenue, and improve capacity utilization in air cargo operations.

2.7 Acknowledgments

We wish to thank Air Canada for submitting this very interesting problem.

Bibliography

[1] IATA, The value of air cargo: Air cargo makes it happen (2015).

[2] Boeing, World air cargo forecast. Technical report, Boeing company (2022).

[3] K. T. Talluri, G. J. Van Ryzin, The theory and practice of revenue management, Vol. 68, Springer Science & Business Media, 2006.

[4] K. Amaruchkul, W. L. Cooper, D. Gupta, Single-leg air-cargo revenue management, Transportation Science 41 (4) (2007) 457–469.

[5] K. Pak, R. Dekker, Cargo revenue management: Bid-prices for a 0-1 multi knapsack problem, Available at SSRN 594991 (2004).

[6] B. Feng, Y. Li, Z.-J. M. Shen, Air cargo operations: Literature review and comparison with practices, Transportation Research Part C: Emerging Technologies 56 (2015) 263–280.

[7] S. Yan, S.-C. Chen, C.-H. Chen, Air cargo fleet routing and timetable setting with multiple on-time demands, Transportation Research Part E: Logistics and Transportation Review 42 (5) (2006) 409–430.

[8] F. Xiao, S. Guo, L. Huang, L. Huang, Z. Liang, Integrated aircraft tail assignment and cargo routing problem with through cargo consideration, Transportation Research Part B: Methodological 162 (2022) 328–351.

[9] F. Azadian, A. E. Murat, R. B. Chinnam, Dynamic routing of time-sensitive air cargo using real-time information, Transportation Research Part E: Logistics and Transportation Review 48 (1) (2012) 355–372.

[10] P. P. Belobaba, Survey paper: Airline yield management, An overview of seat inventory control, Transportation Science 21 (2) (1987) 63–73.

[11] J. I. McGill, G. J. Van Ryzin, Revenue management: Research overview and prospects, Transportation science 33 (2) (1999) 233–256.

[12] R. G. Kasilingam, Air cargo revenue management: Characteristics and complexities, European Journal of Operational Research 96 (1) (1997) 36–44.

[13] J. S. Billings, A. G. Diener, B. B. Yuen, Cargo revenue optimisation, Journal of Revenue and Pricing Management 2 (2003) 69–79.

[14] B. Slager, L. Kapteijns, Implementation of cargo revenue management at KLM, Journal of Revenue and Pricing Management 3 (2004) 80–90.

[15] B. Becker, N. Dill, Managing the complexity of air cargo revenue management, Journal of Revenue and Pricing Management 6 (2007) 175–187.

[16] I. Z. Karaesmen, Three essays on revenue management, Columbia University, 2001.

[17] R. G. Kasilingam, An economic model for air cargo overbooking under stochastic capacity, Computers & industrial engineering 32 (1) (1997) 221–226.

[18] S. Luo, M. Çakanyıldırım, R. G. Kasilingam, Two-dimensional cargo overbooking models, European Journal of Operational Research 197 (3) (2009) 862–883.

[19] K. Huang, K.-c. Chang, An approximate algorithm for the two-dimensional air cargo revenue management problem, Transportation Research Part E: Logistics and Transportation Review 46 (3) (2010) 426–435.

[20] R. Hoffmann, Dynamic capacity control in cargo revenue management–a new heuristic for solving the single-leg problem efficiently, Journal of Revenue and Pricing Management 12 (2013) 46–59.

[21] D. L. Han, L. C. Tang, H. C. Huang, A Markov model for single-leg air cargo revenue management under a bid-price policy, European Journal of Operational Research 200 (3) (2010) 800–811.

[22] T. Levina, Y. Levin, J. McGill, M. Nediak, Network cargo capacity management, Operations Research 59 (4) (2011) 1008–1023.

[23] C. Barz, D. Gartner, Air cargo network revenue management, Transportation Science 50 (4) (2016) 1206–1222.

[24] F. Previgliano, G. Vulcano, Managing uncertain capacities for network revenue optimization, Manufacturing & Service Operations Management 24 (2) (2022) 1202–1219.

[25] M. Chen, Z.-L. Chen, Recent developments in dynamic pricing research: multiple products, competition, and limited demand information, Production and Operations Management 24 (5) (2015) 704–731.

[26] G. Gallego, G. Van Ryzin, Optimal dynamic pricing of inventories with stochastic demand over finite horizons, Management Science 40 (8) (1994) 999–1020.

[27] A. Erdelyi, H. Topaloglu, Using decomposition methods to solve pricing problems in network revenue management, Journal of Revenue and Pricing Management 10 (2011) 325–343.

3 AMF: Transaction tracing on the Ethereum platform

Philippe Béliveau [a]
Helen Samara Dos Santos [b]
Frédéric Dupont-Marillia [c]
Jacky Jang [d]
Kiyan Karimi Nemch [e]
Odile Marcotte [f,g]
Jean-François Plante [a,g]
Lingyi Yang [h]
Rami Younes [d]

[a] *HEC Montréal*
[b] *Memorial University of Newfoundland*
[c] *Autorité des marchés financiers*
[d] *Université de Montréal*
[e] *Concordia University*
[f] *UQAM*
[g] *GERAD*
[h] *University of Oxford*

November 2024
Les Cahiers du GERAD

Abstract: *In this report, we describe strategies to improve transaction tracing on the Ethereum platform, i.e., determine the value of a specific transaction. Indeed, there are many types of cryptocurrency on this platform, and the challenge is to find out the value, expressed in the WETH/ETH currency, of the total monetary amount transferred to various "sinks" by the instigator of a given complex transaction. We show that an approach based on graph theory complements the approach currently used by the Autorité des marchés financiers and actually makes it more robust.*

3.1 Introduction and statement of the problem

The rapid growth of cryptocurrencies and Decentralized Finance (DeFi) has introduced significant complexities in financial transactions. Unlike traditional financial systems, DeFi transactions are intricate, involving multiple layers of internal and external addresses, routers, smart contracts, and various operations. This complexity presents considerable challenges for understanding and analyzing these transactions, particularly for ensuring the accuracy and reliability of traced data. The *Autorité des marchés financiers* (AMF) has proposed an investigation of transaction tracing methods to address this issue. The current report examines this problem and proposes a method for transaction tracing on the Ethereum platform.

DeFi transactions are essential because they enable financial services such as transfers, swaps, and liquidity provision to be conducted directly between users without the need for traditional financial intermediaries. This increases accessibility to financial services, enhances transparency, and reduces costs [3].

Blockchain is a distributed ledger that records transactions across multiple computers. This decentralized network maintains a synchronized and consistent log: each computer (node) participating in the blockchain network has a copy of the entire blockchain, meaning all nodes have the same data. To validate and add a new transaction to the blockchain, nodes use consensus mechanisms such as Proof of Work (PoW) or Proof of Stake (PoS) (see [2]). These mechanisms ensure that all copies of the blockchain are synchronized and agree on the current state of the ledger. Once a transaction is recorded in a block and added to the blockchain, it is extremely difficult to alter or delete it because this would require changing the data on all nodes simultaneously. This provides security and integrity to the data.

The Ethereum platform is built on blockchain technology. It uses this decentralized ledger to securely record transactions and execute smart contracts. Smart contracts are self-executing contracts with the terms directly written into the code. These contracts enable the automatic enforcement and execution of agreements, eliminating the need for intermediaries. This blockchain-based framework allows Ethereum to support decentralized applications (dApps) and complex financial transactions without relying on central authorities.

Tokens on Ethereum are digital assets that can represent various items, from currencies to property. They come in different types: for example, ERC-20 tokens are fungible, meaning each token is identical, while ERC-721 tokens are non-fungible, meaning each token is unique.

Liquidity Pools provide liquidity to decentralized exchanges (DEXs) by allowing users to deposit tokens into a pool. This liquidity enables other users to swap tokens directly from the pool. Liquidity pools use Automated Market Makers (AMMs) to set prices based on supply and demand, and they reward liquidity providers with a share of the transaction fees generated by the pool.

Routers in DeFi find the best path for a transaction, ensuring users get the most efficient and cost-effective route for their trades. Routers can work with other systems (such as specialized cross-chain bridges and wrapping protocols), for example, to convert non-ERC20 tokens into ERC20 tokens, making them compatible with the Ethereum platform.

These components interact seamlessly on the Ethereum platform to create an elaborate and robust environment for DeFi. Smart contracts automate complex financial operations, tokens represent assets, liquidity pools ensure there is enough liquidity for trades, and routers optimize transaction paths. Together, they allow a wide range of financial activities to be conducted in a decentralized manner, improving the overall efficiency and security of the financial system.

`Ethereum.org` is the official website for the Ethereum project, and `Etherscan.io` is primarily a blockchain explorer for the Ethereum platform. It allows users to view detailed information about transactions, addresses, tokens, smart contracts, etc. It provides real-time data and analytics, including gas prices, block times, and historical data.

Ether (ETH) is the native cryptocurrency of the Ethereum blockchain. According to `Ethereum.org` [1], "There are many cryptocurrencies and lots of other tokens on Ethereum, but there are some things that only ETH can do." It is used to pay for transaction fees (gas) and as a form of value transfer within the network. Wrapped Ether (WETH) is ETH wrapped in an ERC-20 compatible token.

The primary goal of this project is to improve the accuracy of transaction tracing and explain these transactions, specifically by determining the total asset value expressed in the WETH currency. The approach currently employed by the AMF is efficient for simple transactions but becomes computationally expensive for complex transactions, i.e., transactions consisting of several asset transfers. The AMF approach involves extracting main, internal, and ERC20 transactions, tracing each one, extrapolating the predicted rate, and looking for close ETH values in ERC transactions, then replacing their values and completing the process.

The following five sections of this report describe, respectively, the data sets, a supervised learning approach, the modelization of complex transactions through graph theory, the proposed algorithms, and our results. We conclude the report with a section on future work.

3.2 Description of the data sets

The original data used in this project (and by the AMF) consists of detailed transaction records found on the Ethereum project website. From `Etherscan.io`, we have the following examples of transactions.

Figure 3.1 displays a transfer of 1,695 Tether USD (USDT). Figure 3.2 is an example of a complex transaction, i.e, a transaction with many sub-transactions.

The AMF put the original data in tabular format and provided us with JSON and Excel spreadsheets. We had access to data from the 20th, 24th, and 28th of March 2024, as well as the 5th and 28th of April 2024.

Within the provided spreadsheets some of the columns are labeled as follows.

tokenName/tokenSymbol: The name of the token involved in the transaction (e.g., "Toshi", "Brett").

value: The amount of the token transferred in the transaction.

blockNumber: The block number in which the transaction was included.

timeStamp: The timestamp when the block was created.

date: The date and time when the transaction occurred.

from: The address initiating the transaction.

to: The address receiving the transaction.

tx_type: The type of transaction (e.g., ERC20, call).

type: The nature of the transaction, if known (e.g., transfer, swap).

hash: A unique identifier for the transaction.

transactionFee: The fee paid for the transaction.
tx_source: Indicates the source of the transaction (e.g., main, internal).
blockHash: The hash of the block containing the transaction.
tokenDecimal: The number of decimal places used by the token.
nonce: The number of transactions sent from the sender's address.
ETH: The estimated transaction value expressed in ETH.
ETH_adj: The adjusted estimated transaction value expressed in ETH.
gas: The maximum amount of gas units that the transaction is allowed to use.
gasPrice: The price per unit of gas.
gasUsed: The actual amount of gas used by the transaction.

The initial phase of the project consisted of data cleansing and the creation of a new data frame for our purposes, with the following column headings: hash, source (tx_source), from, to, value, symbol (tokenName/tokenSymbol), sub_type (tx_type), ETH, type, and ETH_adj.

3.3 A supervised learning approach

The AMF is currently using a supervised learning approach to evaluate transactions, the details of which we do not know. In this section, we present the approach proposed by a subgroup of our team. (The other subgroup investigated the graph theory approach, described later in this report.) The first subgroup tested the capability of Large Language Models to interpret logs of sub-transactions. Their approach involved the following steps.

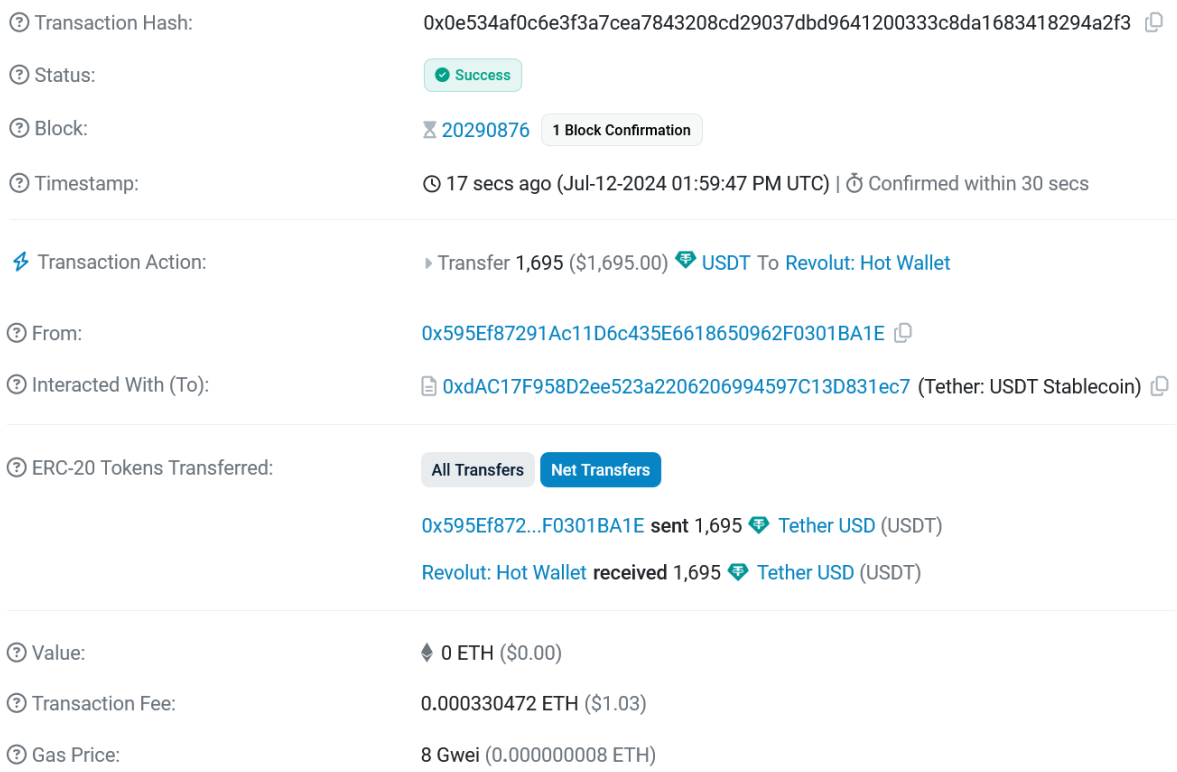


Figure 3.1: Example of a simple transaction.

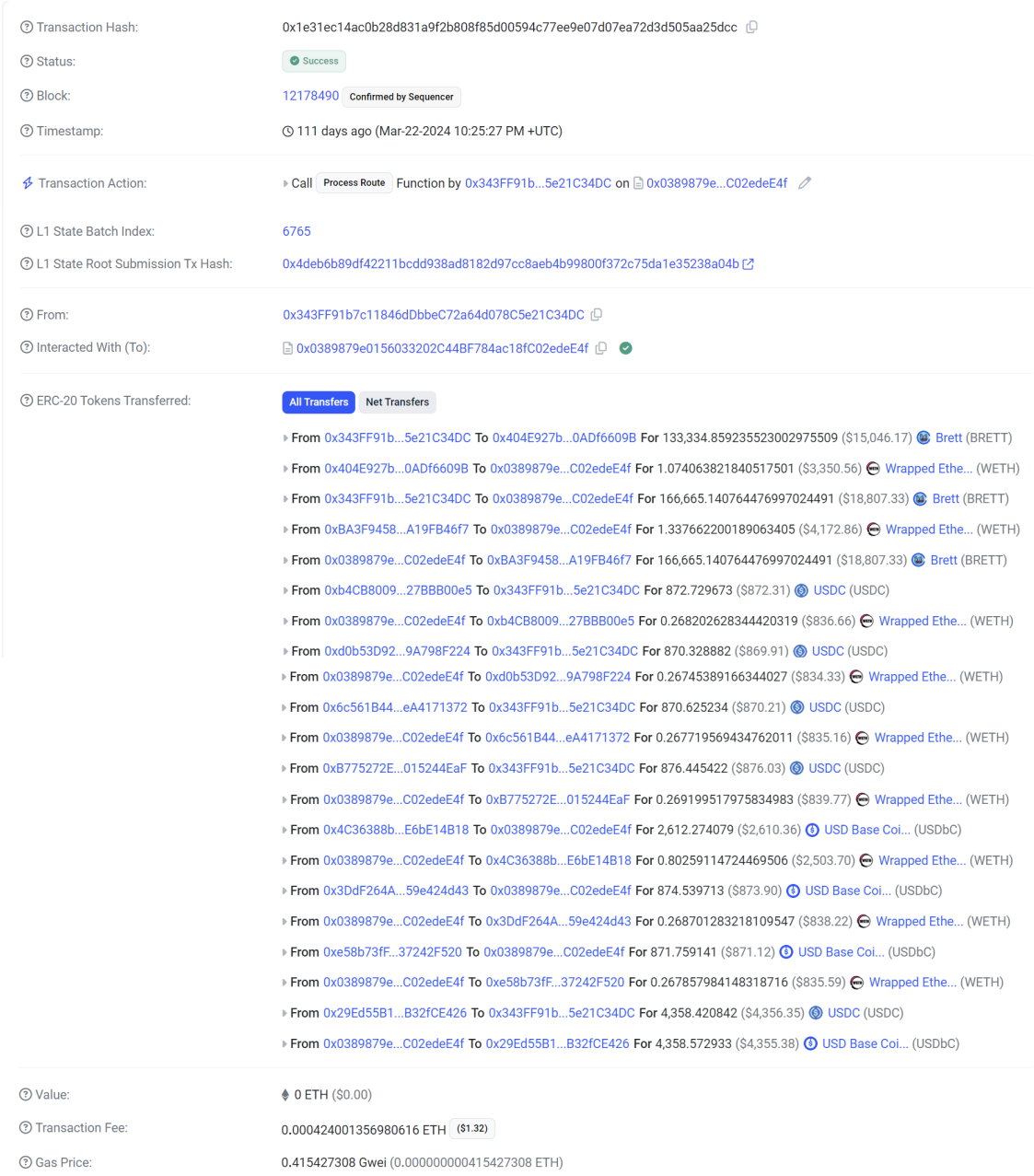


Figure 3.2: Example of a complex transaction.

1. **Feature Extraction**: Transactions are characterized by various features. We can enhance these features by extracting additional network-based metrics, such as:
   - Number of nodes in the network;
   - Node with the highest number of edges;
   - Longest path within the network;
   - Number of sinks (nodes with no outgoing edges), among others.
2. **Manual Labelling**: Certain characteristics of transactions that require investigation, such as the presence of a pool or other specific attributes, can be manually identified from a finite set of examples.

3. **Model Training**: A supervised learning model is then trained using these manually labelled examples. This model learns to recognize patterns associated with these labels.
4. **Prediction**: Once trained, the model can be applied to classify a large volume of transactions based on the features extracted.

For an illustration of potential use, consider a long list of transactions where:

- A network algorithm extracts the value in WETH/ETH;
- Additional information about the sub-transactions network shape is gathered;
- The machine learning algorithm classifies the transactions based on this information.

The result is a summary of all transaction types, allowing for more efficient analysis and investigation.

Figure 3.3: Example of model's input.

Figure 3.4: Example of model's output.

3.4 Modelization through graph theory

A graph is a mathematical structure used to model pairwise relationships between objects. In this model, the objects are represented as nodes (or vertices), and the relationships between them are represented as edges (or arcs). In an undirected graph, an edge is a pair of nodes, and in a directed graph (or digraph), an arc is a couple of objects denoted by (u, v), where u and v are two (not necessarily distinct) objects. Thus, an arc has a direction and may represent a "transfer" of some quantity from u to v. To allow for the possibility that there are several arcs from u to v, we introduce the notion of multidigraph. An arc may also have a label or several labels. For our purposes, we define a *labeled multidigraph* as a couple (N, A), where N is a finite set of nodes and A a finite multiset of quadruples of the form (u, v, c, w) with u and v being nodes, c representing a colour or token type, and w being a weight (i.e., a real number). We refer the reader to *Graph Theory With Applications*, by Bondy and Murty, for the standard graph-theoretical phrases used in the sequel.

We propose modelling a main transaction as a labeled multidigraph ("graph"), where the nodes represent the addresses involved in the sub-transactions and an arc represents the flow from one address to another. Each arc carries information about the token symbol (cryptocurrency) and its value in the corresponding sub-transaction.

A graph is an abstract discrete mathematical structure, but it can be embedded into the plane and thus visualized. For example, after the anonymization of the involved tokens (denoted by $T1$, $T2$, and $T3$), the transaction shown in Figure 3.2 can be visualized as the graph model embedding displayed in Figure 3.5.

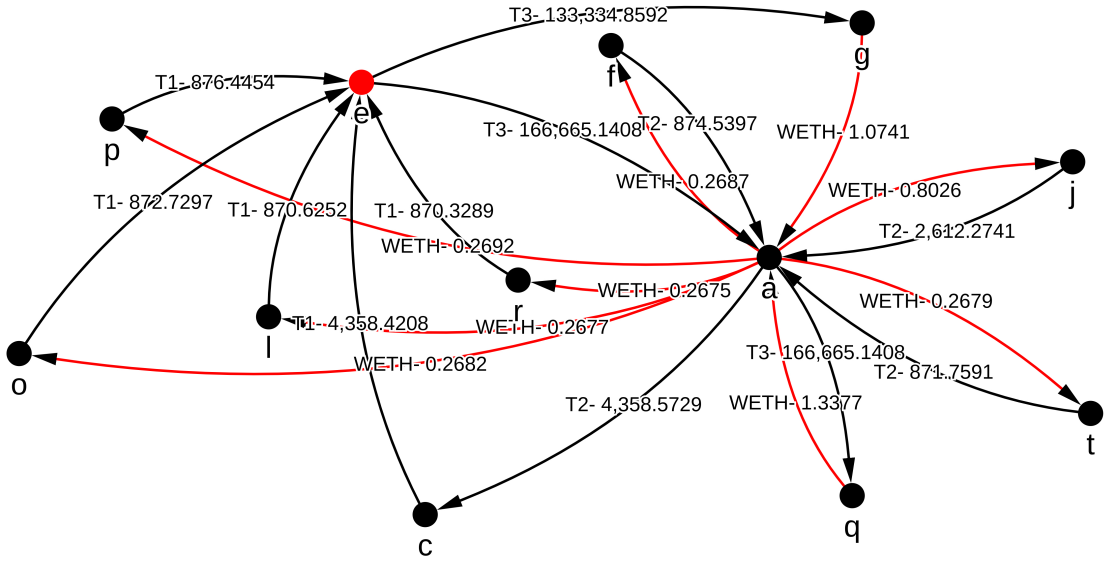


Figure 3.5: Graph illustration representing the transaction described in Figure 3.2.

For simplicity, we will refer to the graph embedding simply as a "graph." The instigator of the transaction is always known and is represented by a red node in the graph. Moreover, any red arc represents a quadruple (u, v, c, w) such that the label c is WETH/ETH.

Given a labeled multidigraph $G = (N, A)$, an *arc cut* $C \subset A$ is a non-empty multiset of arcs such that the graph $G' = (N, A - C)$ is disconnected, i.e., there exists a partition (N_1, N_2) of the node set of G' such that there is no quadruple (u, v, c, w) in G' with $u \in N_1$ and $v \in N_2$. Naturally, this is equivalent to saying that there exists a partition (N_1, N_2) of the node set of G such that C includes every quadruple (u, v, c, w) with $u \in N_1$ and $v \in N_2$.

Let C be an arc cut whose removal destroys all the paths from the instigator of the main transaction to the recipients of the assets. Intuitively, one realizes that any asset or money dispatched by the instigator must "go through" one of the arcs of C. Therefore, since WETH is the reference currency (similar to the gold standard or the US dollar), one should look for an arc cut that only consists of WETH/ETH arcs.

We say that the node u is a *source* if it has only outgoing arcs, i.e, it only appears in quadruples of the form (u, v, c, w). We say that the node u is a *sink* if it has only incoming arcs, i.e., it only appears in quadruples of the form (v, u, c, w).

To traverse the graph in search of an arc cut, we need a source. The instigator of the transaction is always known, so it is a natural choice for a source. There might be cases, however, where the instigator also receives tokens, i.e., it appears in incoming arcs. To remedy this, if the instigator is not a source, we introduce into the graph a new node called *Dummy* and redirect the instigator's incoming edges to this *Dummy* node. For example, after adding a *Dummy* node, the graph in Figure 3.5 is transformed into the graph displayed in Figure 3.6.

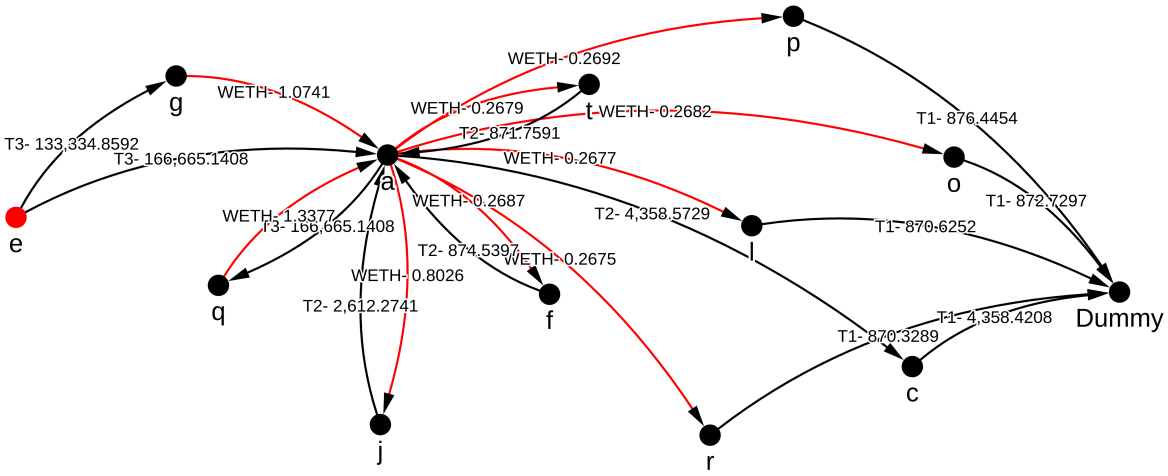


Figure 3.6: Updated graph representation of the transaction in Figure 3.2.

In our initial algorithm, if a path existed from the instigator (now a source) to a sink where none of the arcs along the path had the WETH/ETH label, the transaction was flagged as untraceable. We can reduce, however, the number of transactions flagged as untraceable by addressing specific graph patterns, as detailed below.

We say that a token T *flows through* a node n if n has at least one incoming or outgoing arc labelled T. A node n is said to be a *flow-conservation* node if, for any token T that flows through n, the sum of the values of all incoming arcs of label T equals the sum of the values of all outgoing arcs of label T.

We split a flow-conservation node, i.e., for any token T that flows through the node n, we create a new node n_T and redirect all incoming and outgoing arcs of label T to n_T. Finally, we delete the node n from the graph. For an example, we refer the reader to Figure 3.9 in the Section 3.5.1.

A path from a source to a sink is called a *monochromatic path* if all arcs along the path have the same label. In graph theory, it is a standard notation to refer to an arc label (in this context, the asset type being dispatched) as a colour.

We handle a monochromatic path as follows: we identify the minimum arc value on the path and denote it by *valuemin*. We then update each arc value along the path by subtracting *valuemin* from it. We carry out this operation as many times as possible, and after that, we look for a WETH/ETH arc cut by carrying out a breadth-first search (BFS) starting at the instigator node. With this approach, we could identify arc cuts for all the reconstructed transactions. If the arc cut does not include any WETH/ETH arc, the transaction is assigned a value of 0.

3.5 Proposed algorithms

After processing the data and modelling the transaction as a graph, we modify the graph if necessary by including a dummy into it, splitting every flow-conservation node, detecting as many monochromatic paths as possible, and updating the arc values of these paths.

Then, starting from the instigator (now a source), we traverse the modified graph using a Breadth-First Search (BFS) algorithm, looking for an arc cut that disconnects the graph into two components.

This deterministic approach provides us with an estimate of the value of the transaction: the sum of the values of the arcs in the arc cut that are of symbol WETH/ETH. We will denote this value by "Transaction_Value".

When there is a path from the instigator (source) to a sink in which none of the edges are of symbol WETH/ETH and the arc cut does not include any arc of symbol WETH/ETH, the transaction is flagged as having a zero value.

3.5.1 Illustration of the Algorithm's process

In this section, we will demonstrate how the algorithm operates using a specific transaction (of hash "0xe...79c") as an example. This transaction is categorized as a "swap", has an "ETH_adj" of 1.577414, and its "Transaction_Value" is 1.612981.

For simplicity, we will display only the arcs with non-zero values. The graph representation of this transaction consists of 16 nodes and 27 arcs, as illustrated in Figure 3.7.

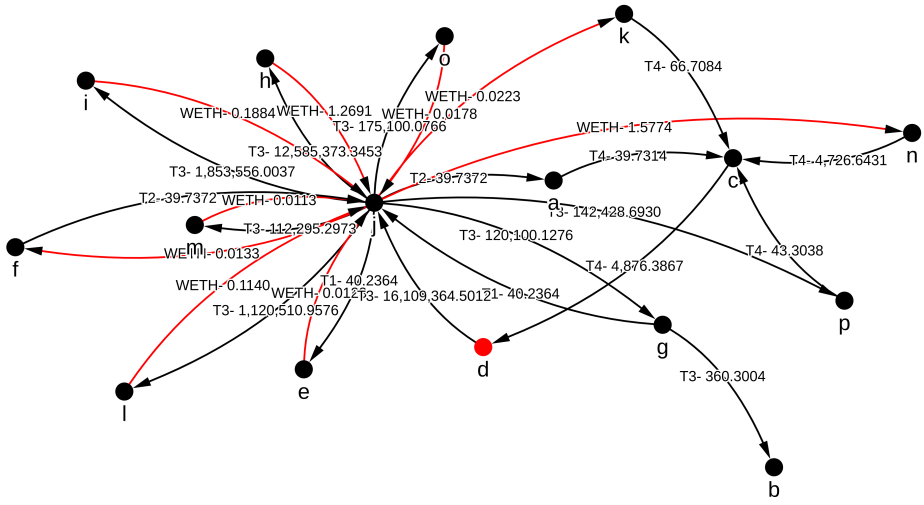


Figure 3.7: Graph representation of the transaction.

In Figure 3.7 the instigator is not a source, so we add a "Dummy" node to the graph to make the instigator a source and obtain the graph displayed in Figure 3.8.

The graph in Figure 3.8 has two flow-conservation nodes: "c" and "j". Only "T4" flows through "c", so the node "c" is renamed "c_T4". Four tokens flow through the node "j", so it will be split into four different nodes: "j_WETH", "j_T1", "j_T2", and "j_T3". The updated graph has 20 nodes and 27 arcs, as illustrated in Figure 3.9.

There is a monochromatic path consisting of three arcs: from "d" to "j_T3", from "j_T3" to "g", and finally from "g" to the sink "b". The "valuemin" of this path is the value of the arc ("g", "b"), which is 360.30038 T3. After updating the monochromatic path, the final graph is shown in Figure 3.10.

Finally, the arcs connecting the set of marked nodes to the set of unmarked nodes define the arc cut. We have the following results.

- Marked Nodes, in the order in which they are marked: "d", "j_T3", "l", "m", "h", "p", "i", "o", "g", "c_T4", "j_T1", "Dummy" and "e".
- Unmarked nodes: "j_WETH", "k", "b", "n", "j_T2", "a", and "f".

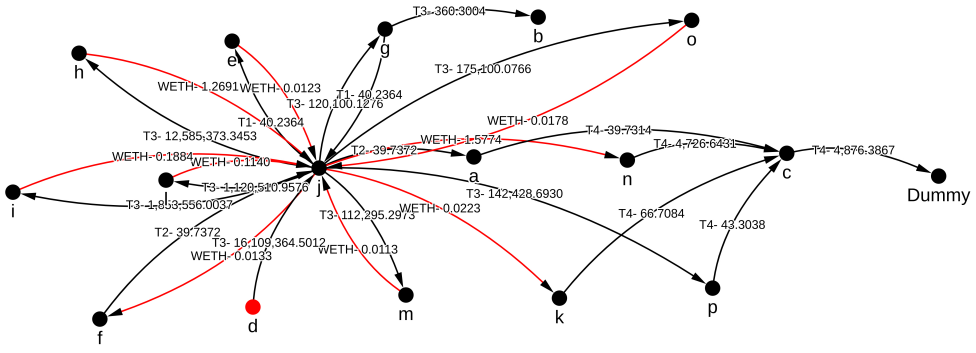


Figure 3.8: Graph with the node "Dummy" added.

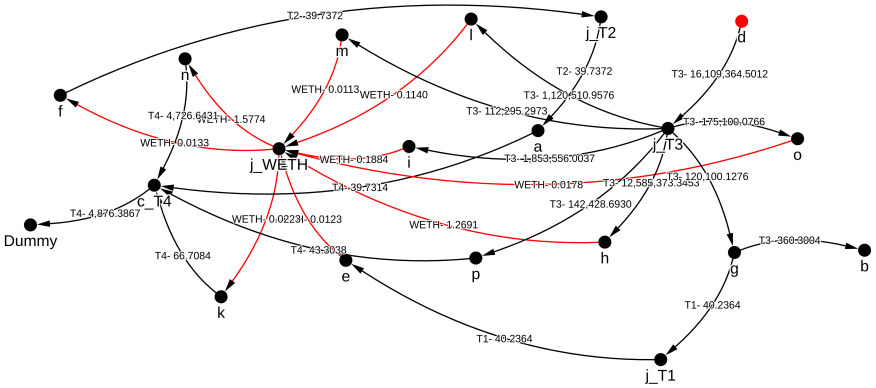


Figure 3.9: Graph with the flow-conservation nodes split.

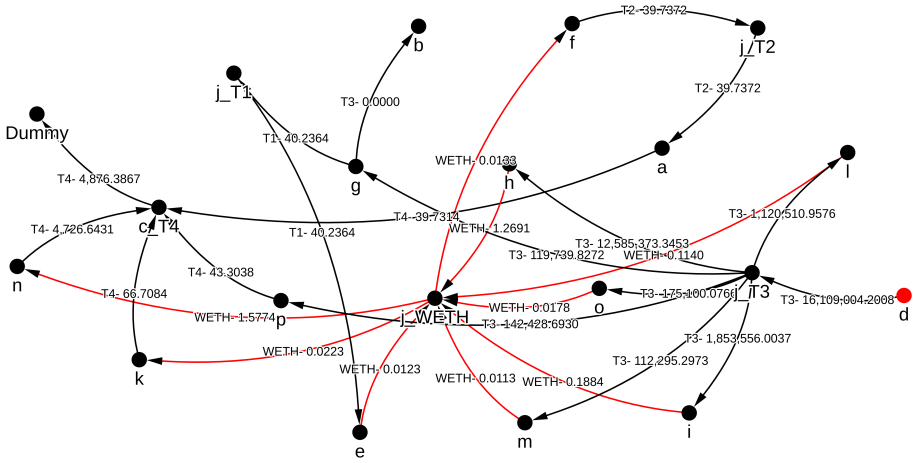


Figure 3.10: Final processed graph.

- Arc cut: { (“h”, “j_WETH”, WETH, 1.2691), (“l”, “j_WETH”, WETH, 0.1140), (“o”, “j_WETH”, WETH, 0.0178), (“i”, “j_WETH”, WETH, 0.1884), (“m”, “j_WETH”, WETH, 0.0113),(“e”, “j_WETH”, WETH, 0.0123) }

The Transaction_Value (in WETH) is 1.6129812504136791.

3.5.2 Pseudocodes

To find a monochromatic path, we use breadth-first search (BFS), a standard algorithm for exploring the nodes of a graph. Recall that BFS constructs a rooted tree, so that the predecessor of a node (its “father”) is uniquely defined. Let c be the colour (token type) of an arc of the form (s, u), where s is the source (the instigator). The following algorithm finds monochromatic paths of colour c from the source to a sink. The process is then repeated for every other colour incident to the source. Note that for the results presented in later sections, we only computed monochromatic paths appearing in the rooted tree initially constructed by the BFS algorithm: in some cases, it could be useful to run the algorithm a second time, for instance, when the collection of monochromatic paths forms a directed acyclic graph that is not a rooted tree.

In what follows a queue is a FIFO (first in, first out) list.

Algorithm 3.1 Algorithm for finding a monochromatic path (of colour c).

begin
Initialize Q to the value `EmptyQueue`
Add the source s to Q and mark s
while Q is not empty **do**
  Let u be the first element of Q
  Remove u from Q
  for every v such that (u, v) is an arc of the graph with nonzero value **do**
    if (u, v) is of colour c and v is not marked **then**
      Add v to Q (at the end of Q) and mark v
      Assign the value u to `predecessor`(v)
    end if
  end for
end while
if a sink t is marked **then**
  `valuemin` $\leftarrow$ `infinity`
  $v \leftarrow t$
  while $v \neq s$ **do**
    `valuemin` $\leftarrow$ `min`(`valuemin`, value of the arc (`predecessor`(v), v))
    $v \leftarrow$ `predecessor`(v)
  end while
  $v \leftarrow t$
  while $v \neq s$ **do**
    Decrease the value of the arc (`predecessor`$(v), v)$ by `valuemin`
    $v \leftarrow$ `predecessor`(v)
  end while
end if
end

When no more monochromatic path from the source to a sink can be found, we use BFS again to find an arc cut constructed by starting BFS at the instigator node. The following algorithm returns a set of marked nodes S such that S includes the source. The arc cut includes any arc from a node in S to a node in T, where T denotes the complement of S. If every arc in this cut is of the WETH/ETH type, we have found the value of the transaction, defined as the sum of the values of the arcs in the cut that are of the WETH/ETH type.

Algorithm 3.2 BFS to find the desired arc cut.

begin
Initialize Q to the value `EmptyQueue`
Add the instigator node (source) w to Q and mark w
while Q is not empty **do**
  Let u be the first element of Q
  Remove u from Q
  for every v such that (u, v) is an arc of the graph with nonzero value **do**
    if (u, v) is not a WETH/ETH arc and v is not marked **then**
      Add v to Q (at the end of Q) and mark v
    end if
  end for
end while
end

3.6 Results

We gathered all available information for (sub)transactions from various Excel files and grouped them by hash to reconstruct the transactions.

From the data provided, we reconstructed 83,486 transactions of different types, and overall, we were able to trace 92.728% of them to a non-zero value. From the transactions whose type is "swap", we traced 99.986% of them to a non-zero value.

We focused on transaction types that do not begin with "0x" and have a minimum of ten samples. Investigating additional transaction types is included in the scope of future work. A summary of the results is presented in Table 3.1.

Table 3.1: Percentage of transactions (per type) that were traced to a non-zero value.

Type	Counts	Traced
swap	74011	99.99%
remove liquidity: multicall	581	0.00%
add liquidity: multicall	383	100.00%
transfer	217	19.35%
rebalance	100	0.00%
exactOutputSingle	88	100.00%
remove liquidity: collect	86	0.00%
swapExactTokensForTokens	69	100.00%
execute	65	93.85%
reinvest	52	0.00%
safeTransferFrom	48	0.00%
add liquidity: mint	40	0.00%
settleOrders	39	71.79%
non processed	29	72.41%
remove liquidity: safeTransferFrom	29	0.00%
swapExactTokensForETHSupportingFeeOnTransferTokens	24	100.00%
add liquidity: addLiquidityETH	17	100.00%
transfer token other than the pool	17	29.41%
bridge	12	100.00%
withdrawTokens	11	0.00%

Comparing with the data provided, out of the twenty transaction types analyzed, we successfully matched or exceeded the number of (non-zero) traced transactions in fourteen cases (see Figure 3.11).

Table 3.2 contains a summary of the percentage of transactions for which the value of "ETH_adj" is matched, i.e., the absolute value of the difference between the "Transaction_Value" and "ETH_adj" is less than or equal to 0.0001, based on the traced transactions.

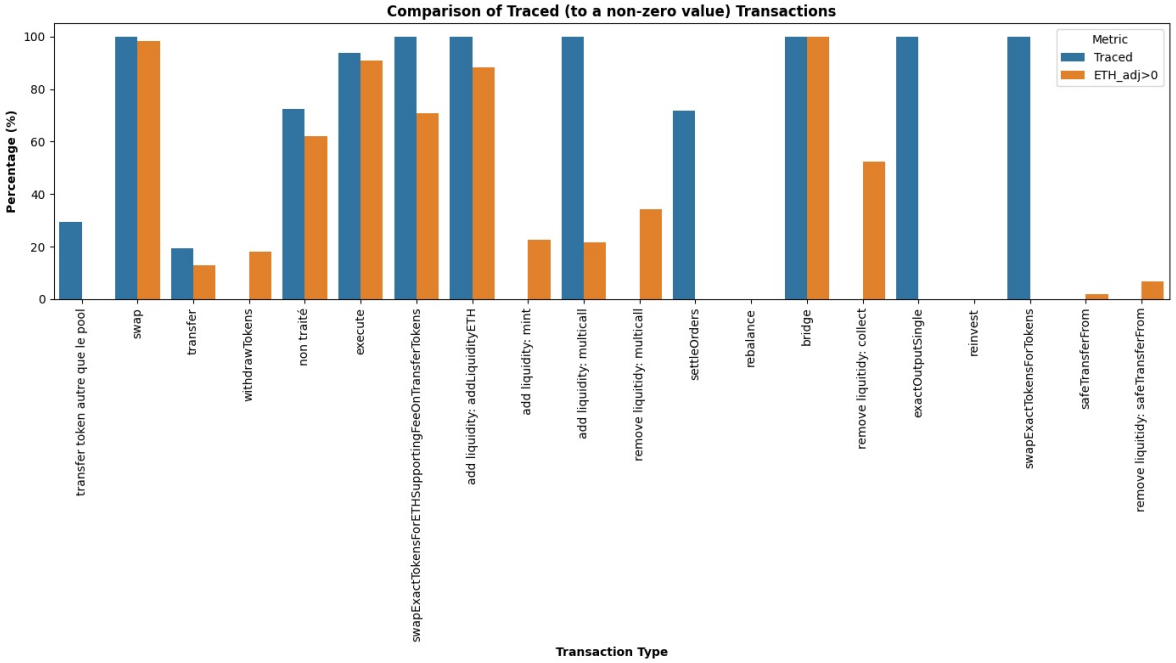


Figure 3.11: Comparison of the percentage of traced (to a non-zero value) transactions.

Table 3.2: Transaction types and their percentage of matches.

Type	Count	Match Percentage
swap	74011	89.83%
remove liquidity: multicall	581	0.00%
add liquidity: multicall	383	20.89%
transfer	217	0.00%
rebalance	100	0.00%
exactOutputSingle	88	0.00%
remove liquidity: collect	86	0.00%
swapExactTokensForTokens	69	0.00%
execute	65	81.97%
reinvest	52	0.00%
safeTransferFrom	48	0.00%
add liquidity: mint	40	0.00%
settleOrders	39	0.00%
remove liquidity: safeTransferFrom	29	0.00%
non processed	29	66.67%
swapExactTokensForETHSupportingFeeOnTransferTokens	24	50.00%
add liquidity: addLiquidityETH	17	88.24%
transfer token other than the pool	17	0.00%
bridge	12	91.67%
withdrawTokens	11	0.00%

Overall this approach allows us to trace a larger number of transactions, and once traced, the deterministic method provides a high level of confidence in the results. Moreover the algorithms used are of linear complexity.

3.6.1 Examples

We present here examples of transactions where the “Transaction_Value” does not match “ETH_adj”. All tokens, except for WETH/ETH, have been anonymized and are denoted by Ti, where i is a natural number.

Transactions of type “swap”

Out of the 74,011 transactions of type “swap” that we reconstructed, we traced 74,000 to a non-zero value. Out of the traced transactions, 1,235 had no value in the “ETH_adj” column, while in 6,291 cases, the value we identified differed from the “ETH_adj” value by more than 0.0001.

The graph in Figure 3.12 represents the transaction with hash “0x4...764”. This transaction is of type “swap”, has an “ETH_adj” value of NaN, and the “Transaction_Value” is 1.511991 WETH/ETH.

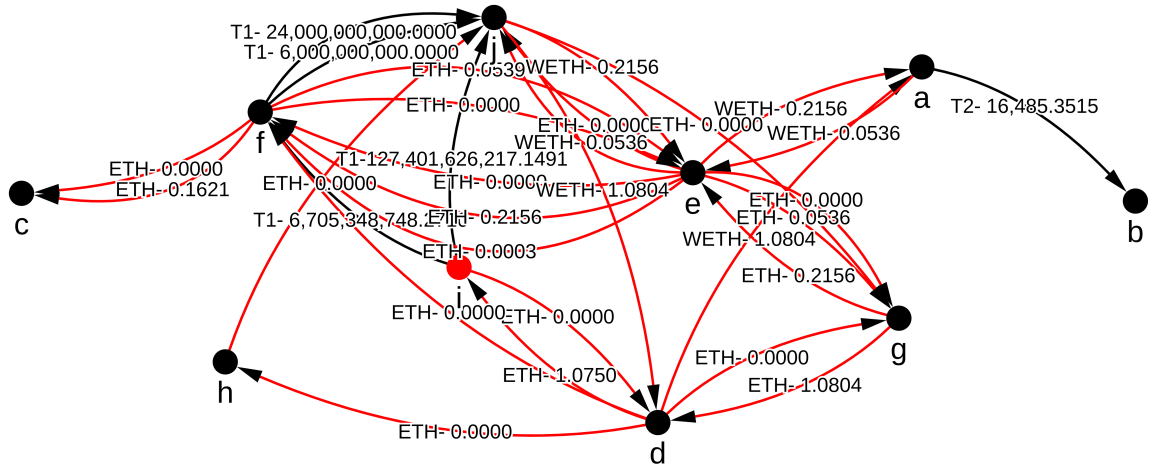


Figure 3.12: Traced transaction of type “swap” that had no “ETH_adj” value.

Figure 3.13 displays the same transaction with sub-transactions of zero value filtered out for better readability.

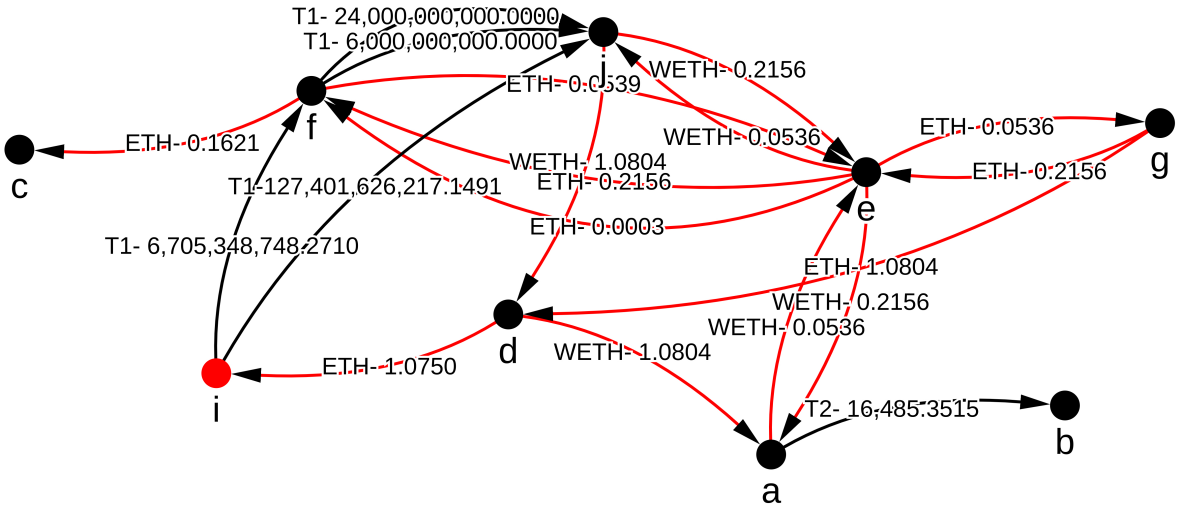


Figure 3.13: The same transaction as in Figure 3.12 with zero-value sub-transactions removed.

The marked nodes are “f”, “j”, and “i”. The non-zero arcs in the arc cut are (“f”, “e”, ETH, 0.0539), (“f”, “c”, ETH, 0.1621), (“j”, “e”, WETH, 0.2156), and (“j”, “d”, WETH, 1.0804).

Figure 3.14 is another example where the transaction value does not match the “ETH_adj” value. This transaction of hash “0xd...779” is of type “swap”, has an “ETH_adj” value of 0.103824 and a “transaction_fee” value of 0.010074, and the “Transaction_Value” is 0.193881 WETH/ETH.

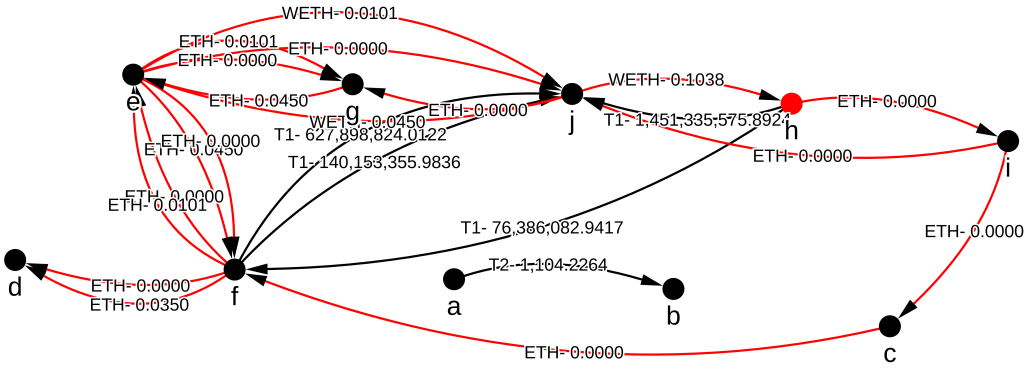


Figure 3.14: Traced transaction of type “swap” where the “Transaction_Value” does not match the “ETH_adj” value.

The graph in Figure 3.15 shows the same transaction with sub-transactions of zero value filtered out.

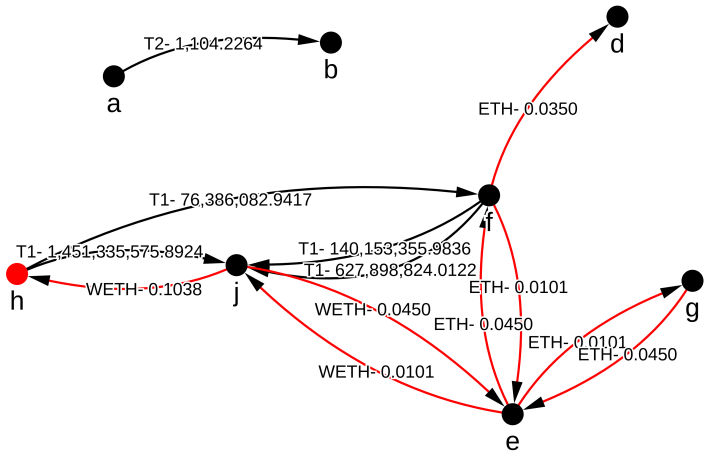


Figure 3.15: The same transaction as in Figure 3.14 with zero-value sub-transactions removed.

The marked nodes are “j”, “h”, and “f”. The non-zero arcs in the arc cut are (“j”, “e”, WETH, 0.0450), (“j”, “Dummy”, WETH, 0.1038), (“f”, “d”, ETH, 0.0350), and (“f”, “j”, ETH, 0.0101).

Transactions of type “non processed”

We reconstructed 29 transactions of type “non processed”. In the “ETH_adj” column, 18 transactions had a non-zero value, while 11 showed NaN.

We traced 21 transactions to a non-zero value and found a match with the “ETH_adj” value in 14 cases.

Of the 8 transactions flagged as having zero value, 6 had NaN in the “ETH_adj” column.

From the 11 transactions with NaN in the “ETH_adj” column, we traced 5 to a non-zero value. To illustrate, Figure 3.16 represents the transaction with hash “0xd...927”. This transaction is of type “non processed”, has NaN in the “ETH_adj” column, and the “Transaction_Value” is 0.000209 WETH.

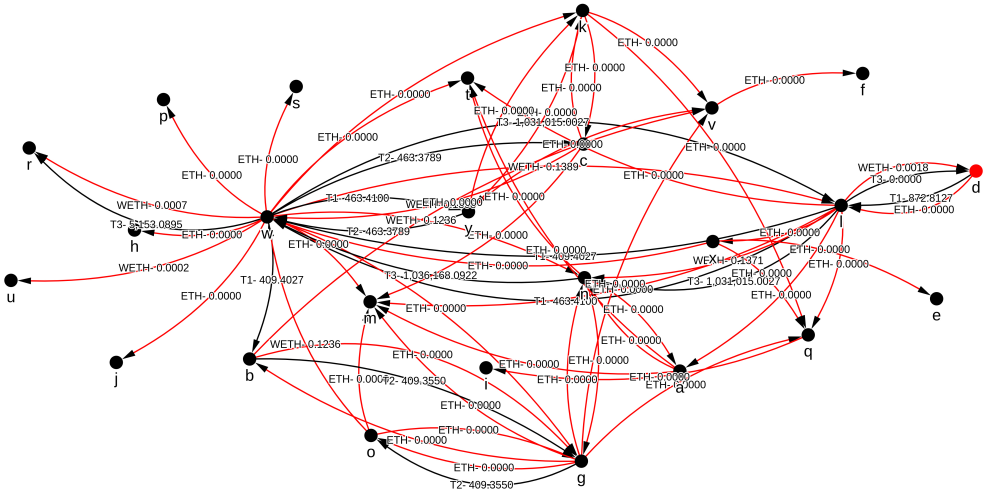


Figure 3.16: Traced transaction of type “non processed” that had no “ETH_adj” value.

Figure 3.17 shows the same transaction with zero-value sub-transactions filtered out.

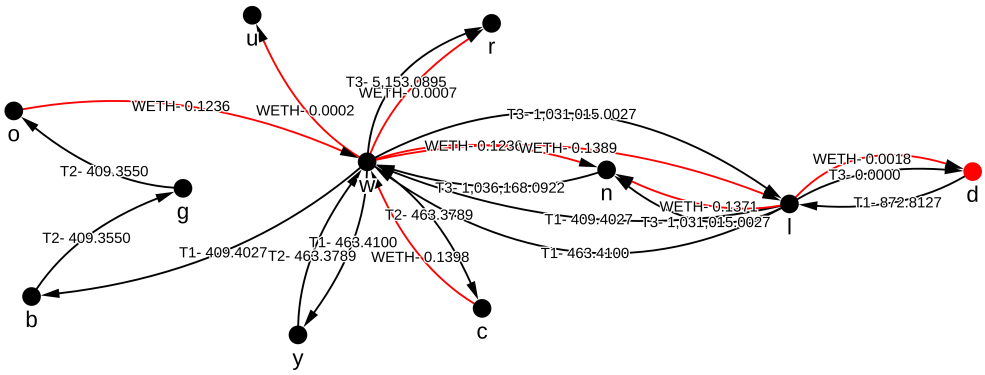


Figure 3.17: The same transaction as in Figure 3.16 with zero-value sub-transactions removed.

The marked nodes are “b”, “c”, “Dummy”, “y”, “g”, “l”, “r”, “d”, “n”, “w”, and “o”. The non-zero arc in the arc cut is (“w”, “u”, WETH, 0.0002).

Figure 3.18 represents the transaction with hash “0x7...aa3”. This transaction is of type “non processed”, has an “ETH_adj” value of 0.047646, and the “Transaction_Value” is 0.12924 WETH.

Figure 3.19 shows the same transaction with zero-value sub-transactions filtered out.

The marked nodes are “b”, “h”, and “d”. The non-zero arc in the arc cut is (“h”, “i”, WETH, 0.1292).

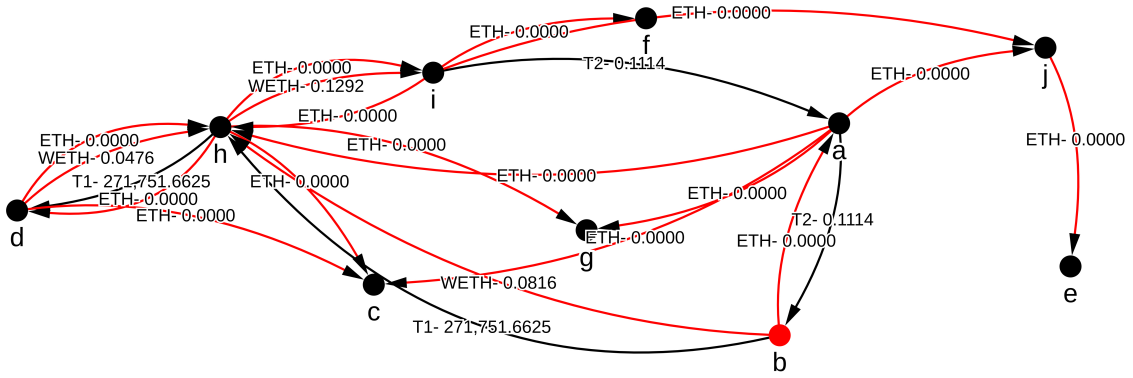


Figure 3.18: Traced transaction of type "non processed" where the "Transaction_Value" does not match the "ETH_adj" value.

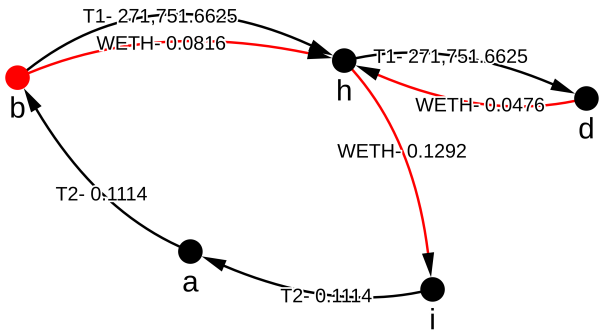


Figure 3.19: The same transaction as in Figure 3.18 with zero-value sub-transactions removed.

3.7 Future work

3.7.1 Data processing

Reconstructing the transactions is an essential part of the work, as our approach depends on the quality of the graphs. Thus it is important to test different ways to reconstruct transactions and evaluate which one performs best in terms of accuracy and complexity.

As an example let us consider the transaction of hash "0x4...253", whose type is "add liquidity: mint" and "ETH_adj" is "NaN".

Given the available data we have for this hash, excluding only transactions of the *sub_type* "staticcall", this transaction can be represented by the graph in Figure 3.20.

By standardizing all the node addresses, we obtain the following graph (Figure 3.21) representation of the transaction.

If we filter the data excluding all non-zero transactions, we obtain the graph in Figure 3.22.

For this transaction, using the graphs in Figures 3.20 or 3.22, our approach estimates the transaction value at 7.7495 WETH. Using the graph 3.21, we estimate the transaction value at 0 WETH. This difference occurs because the node g from Figure 3.20 is a sink, so the arc from f to g will take the

value zero after we update all monochromatic paths in the graph; hence, the only marked node, after we run the algorithm to find an arc cut, is the node f. Analogously, for the graph in Figure 3.22, c is a sink, and the only marked node is b.

Comparing some examples against the *Etherscan* platform, standardizing all the graphs as in Figure 3.21 seemed to be the most accurate method. The results presented in our report are for this kind of data processing.

For the results with the data processing as in Figure 3.22, we refer the reader to the Appendix 3.A.

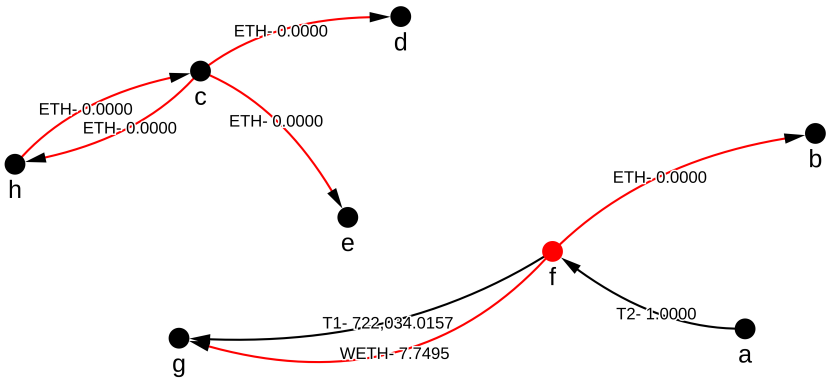


Figure 3.20: Graph representation of the available data for the given hash.

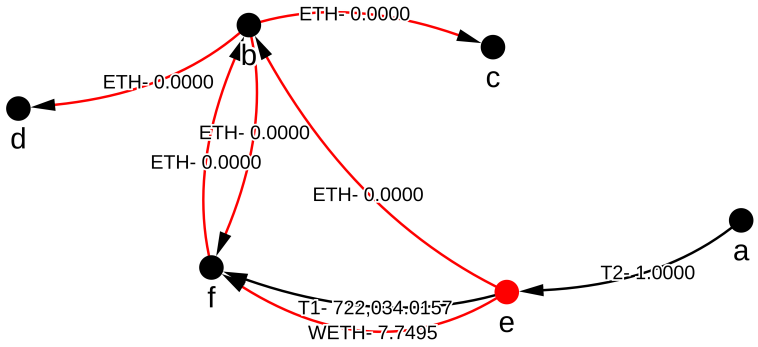


Figure 3.21: Graph representation of the cleaned data for the given hash.

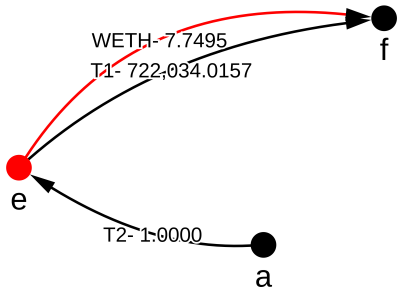


Figure 3.22: Graph representation of the non-zero arcs for the given hash.

3.7.2 Refining the model

The model can be refined to trace accurately more transactions.

The next step should be to differentiate between transactions with zero value and those where the zero value indicates a pattern that requires further investigation for accurate tracing, as illustrated by the transaction shown in Figure 3.20.

For certain transactions types, such as “withdrawTokens”, “add liquidity: mint”, “remove liquidity: multicall”, “remove liquidity: collect”, “safeTransferFrom”, and “remove liquidity: safeTransferFrom”, additional investigation is needed.

Incorporating fee values is also one of the goals of our future work.

Appendix

3.A Results for the finer data processing

We reconstructed 66,591 transactions of different types, and overall, we were able to trace 95.137% of them to a non-zero value. Out of the transactions whose type is “swap”, we traced 99.917% to a non-zero value.

Focusing on transaction types that do not start with “0x” and have at least ten samples, we obtain the results in Table A1.

Table A1: Percentage of transactions (per type) that were traced to a non-zero value.

Type	Counts	Traced
swap	62360	99.92%
remove liquidity: multicall	440	0.00%
add liquidity: multicall	293	100.00%
transfer	203	20.20%
exactOutputSingle	88	100.00%
rebalance	83	0.00%
remove liquidity: collect	69	0.00%
swapExactTokensForTokens	69	100.00%
execute	52	92.31%
safeTransferFrom	41	0.00%
settleOrders	34	0.00%
add liquidity: mint	34	100.00%
remove liquidity: safeTransferFrom	26	0.00%
swapExactTokensForETHSupportingFeeOnTransferTokens	24	100.00%
reinvest	23	0.00%
add liquidity: addLiquidityETH	17	100.00%
non traité	14	92.86%
transfer token autre que le pool	13	38.46%
withdrawTokens	11	0.00%

Comparing with the data provided, out of the nineteen transaction types analyzed, we successfully matched or exceeded the number of (non-zero) traced transactions in thirteen cases.

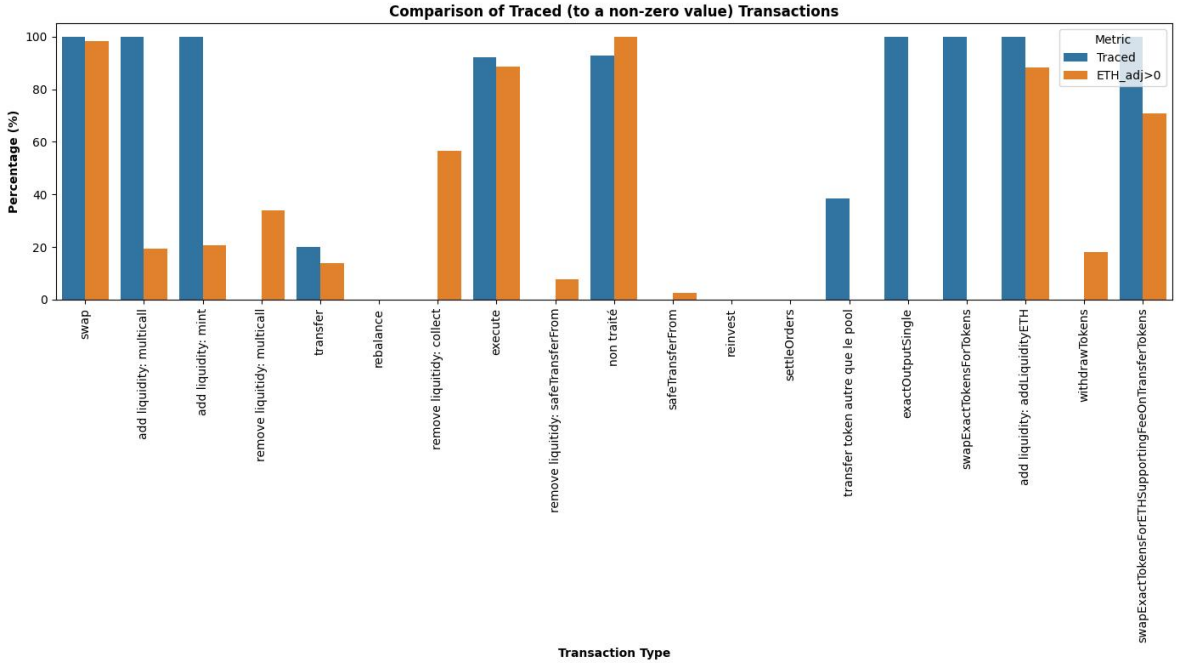


Figure A1: Comparison of the percentages of traced (to a non-zero value) transactions.

Bibliography

[1] Ethereum Foundation. Ethereum. https://ethereum.org/en/eth/, 2023. Accessed: 2024-07-10.

[2] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and Cryptocurrency Technologies. Princeton University Press, Princeton, NJ, 2016.

[3] Fabian Schär. Decentralized finance: On blockchain- and smart contract-based financial markets. Federal Reserve Bank of St. Louis Review, 103(2):153–174, 2021.

4 ECCC: Water level extremes at ungauged locations along the St. Lawrence river and fluvial estuary

Oluwatosin Babasola [a]
Léo Belzile [b]
Thomas Binet [c]
Criscent Birungi [d]
Guillaume Cantin [e]
Janos C. R. Füting [b]
Iheb Hajji [b]
Kristen Hallas [f]
Maria Camila Mejia Garcia [f]
Kayode Oshinubi [g]
Silvia Innocenti [h]
Marianne Fortier [h]
Samuel Perreault [i]

[a] *University of Georgia, Athens*
[b] *HEC Montréal*
[c] *LAGA and Safran Aircraft Engines*
[d] *Concordia University*
[e] *Université de Nantes*
[f] *University of Texas Rio Grande Valley*
[g] *Northern Arizona University*
[h] *Environment and Climate Change Canada*
[i] *University of Toronto*

November 2024
Les Cahiers du GERAD

4.1 Context and problem definition

Under the governmental Flood Hazard Identification and Mapping Program (FHIMP), Environment and Climate Change Canada (ECCC) has been mandated to provide 2D simulations of extreme water levels in the St. Lawrence fluvial estuary under historical and future conditions. The elevations of water levels in this system are triggered by the complex interaction of hydrological, meteorological, and tidal processes that must be considered to simulate river dynamics and flood events. Constraints on the computational resources and time requirements and the necessity for background geophysical fields currently limit the feasibility of producing fine-scale 2D hydrodynamic simulations to a limited set of relatively short extreme events (approximately 400 events with duration ranging from one hour to several weeks). Hence several complementary modelling tools have been explored to study the temporal evolution of water level extreme properties. Among them, some multivariate statistical models and machine learning tools have proven effective in reconstructing continuous water level series over long historical periods, which is essential to assess the extreme probability distributions.

While these methodologies have shown promising performances at gauged stations, some challenges remain in extending their applicability to describe the extreme event characteristics in the river and fluvial estuary sections where few or no observations are available.

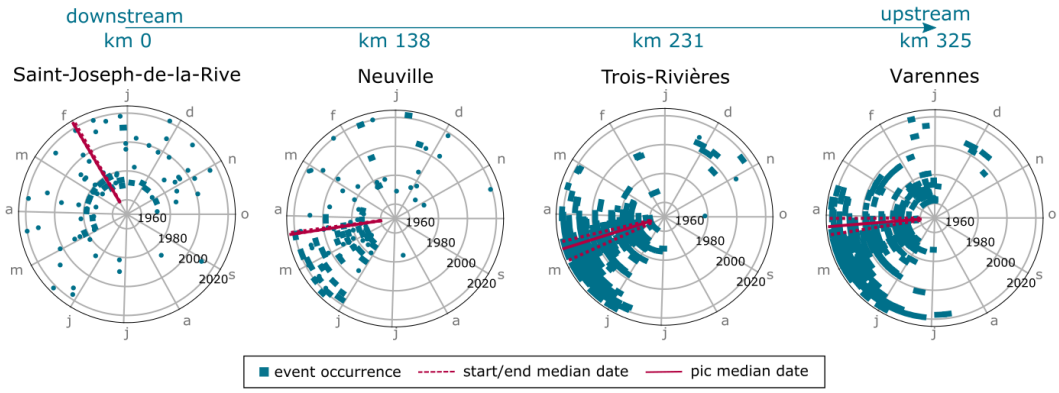


Figure 4.1: High water level occurrences at four water level stations ordered from the most downstream to upstream: the typology of the identified events changes along the river.

We observe short-lived hydrological events in the river downstream sections that occur intermittently between November and June. In contrast, moving upstream, these events become more prolonged and are primarily concentrated during the spring freshet. Additionally, the number of stations involved in the extremes and the event spatial extents vary significantly between upstream and downstream sections. This leads to the first fundamental research question: how do the specific event characteristics (e.g., return period, duration, and seasonality) change locally over the study domain? In this context, during the workshop, we aimed to *determine whether it is possible to summarize local extreme event characteristics into few comprehensive measures*, to evaluate the dominant typology of the events at each location based on these measures, and finally to assess the spatial distribution of these metrics along the river continuum to extrapolate the event typology between monitoring stations.

The project's second phase involves conducting two-dimensional hydrodynamic simulations and statistical reconstructions to reproduce the water level series. The first step is to determine how well these simulations and reconstructions reproduce all relevant features of some specific events that occurred in the past period over the whole study domain. Figure 4.2 provides an example of the

practical issues involved in this evaluation by showing the hydrodynamic simulation outputs from two nearby stations in the central part of the domain during an extreme event.

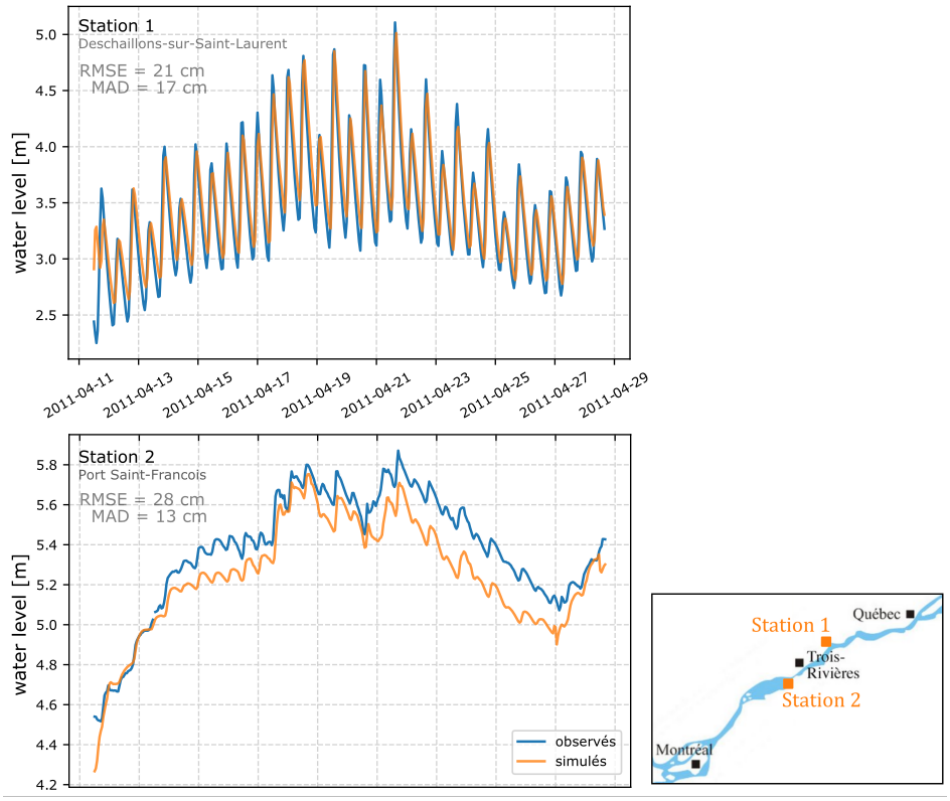


Figure 4.2: Observed water level series and corresponding numerical simulation [H2D2 hydrodynamic model].

The figure shows that the same model leads to different error types at these stations: a clear underestimation of the signal variability and amplitude at the first station, while a constant bias in the mean water level is seen at the second station. This discrepancy raises the following question: at which point in space the model errors change from variability-related to bias-related errors, and how does this transition affect the overall estimation of extreme event characteristics? Accordingly, during the workshop, we tackled the following second objective: *define one or several summary statistic(s) that evaluate the model performance in terms of the ability to reproduce the extreme event characteristics and that can be applied at each spatial point over the study domain.*

4.1.1 Data

The project domain goes from Montréal to Saint-Joseph-de-la-Rive. This spans 450 km and includes two fluvial lakes (Lac Saint-Louis and Lac Saint-Pierre). The domain can be schematically divided into three sections.

1. The Montréal region where the Ontario Lake outflow and Ottawa River streamflow control the water level fluctuations at the daily to decadal scales.
2. The fluvial section from Sorel to Trois-Rivières, including a large fluvial lake, Lac Saint-Pierre, as well as some important tributaries (the Richelieu, Yamaska, Saint-François, and Saint-Maurice rivers). In this section water levels are mainly influenced by long-term hydrological trends and the annual hydrological cycles, while presenting tidal oscillations at diurnal to fortnightly scales in periods with relatively low river discharge.

3. The fluvial estuary, spanning from Trois-Rivières to Saint-Joseph-de-la-Rive: here water level variability is driven by tides at semi-diurnal and lower frequencies as well as by the seasonal variability of the river streamflow. Several tributaries are also present, mostly on the North Shore.

Tide and maritime processes such as storm surge are the main contributors to the water level variability in the proper (maritime) estuary downstream of Saint-Joseph-de-la-Rive.

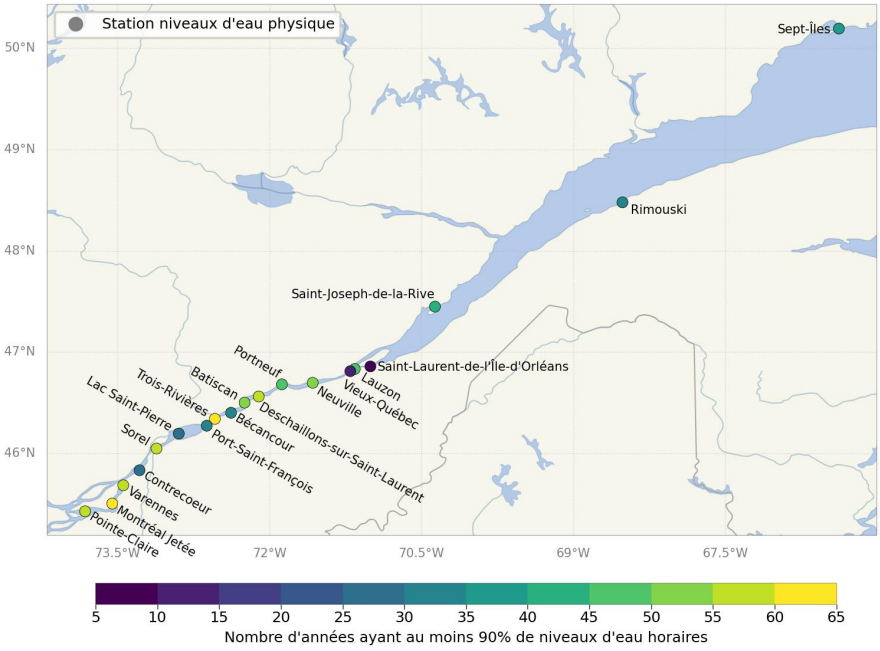


Figure 4.3: Location of the 19 water level stations available.

ECCC originally provided data suitable for a benchmark analysis, including the following datasets.

- Hourly water level records observed at 19 stations over the 1970-2022 period; the station locations are displayed in Figure 4.3.
- Three examples of 2D hydrodynamic simulations obtained with the H2D2 hydrodynamic model for the following high water level events: April 2011, November 2016, and December 2022.
- Hourly water levels reconstructed using three statistical models (non-stationary harmonic regression, recurrent neural network, and basic multiple regression) over the whole study period (1960-2022) at some specific stations and the corresponding extreme events.

Over the course of the week, ECCC provided additional data, including some covariates useful for classifying the event drivers: wind speed series and the water level decomposition in long-term trends and other variability components at one upstream and one downstream station.

In the context of the FHIMP project, ECCC applies a Peak Over Threshold (POT) approach with temporal declustering to identify high-water-level events (see Figure **??**).

To avoid biases in the event selection due to 18.6-yr nodal cycles in tides and possibly long-term climatic trends and cycles, the series were firstly filtered by removing the water level running mean computed on a 20-yr centred moving window. The POT threshold was set for each water level station to sample approximately 2 extremes per year. Finally, the temporal declustering was applied by imposing a minimum spacing between events and decreasing water level from each peak.

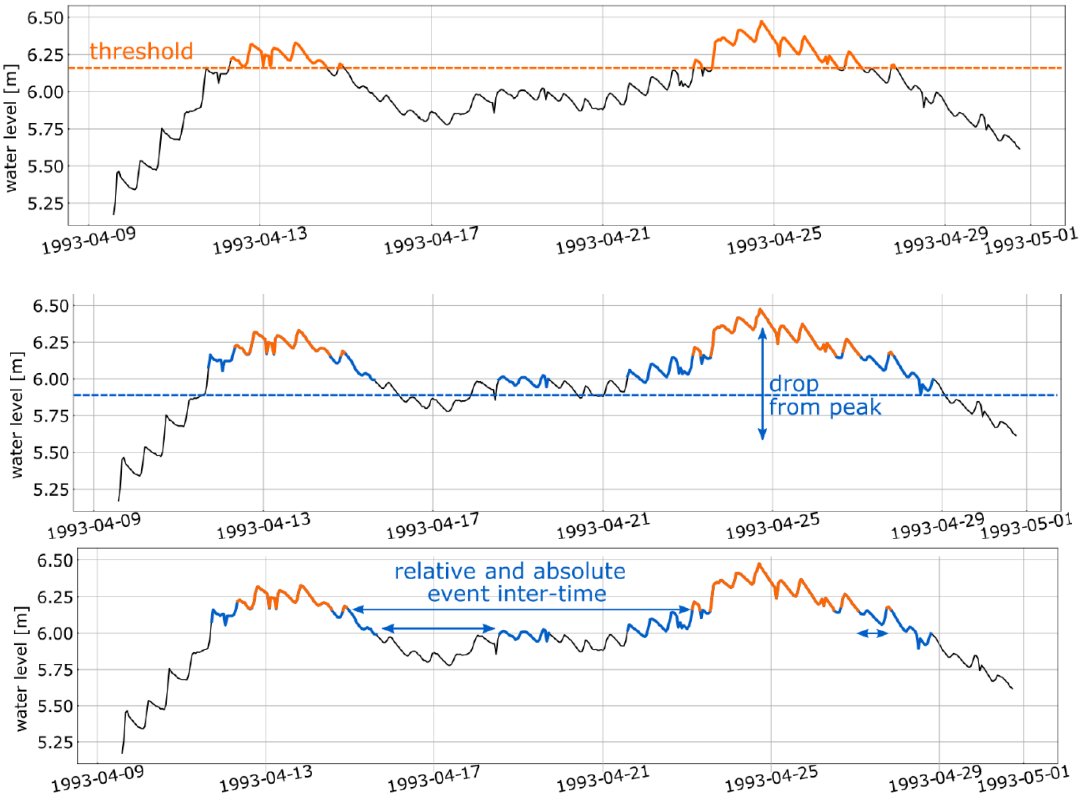


Figure 4.4: Extreme event definition: POT and temporal declustering

Depending on the process generating the high water levels, in many cases, temporal overlaps can be expected between local events sampled at two or more stations, with a time lag of the peaks depending on the celerity of the flow waves. To discern and categorize extreme events on a regional basis effectively, taking into account the spatial heterogeneity and temporal dynamics of these events, the declustering strategy presented above was applied to the series of extreme events identified across all stations. This second declustering results in a set of new start and end dates for events that may occur at several stations in the domain, taking into account the lag time between peaks belonging to the same event (see Figure 4.5).

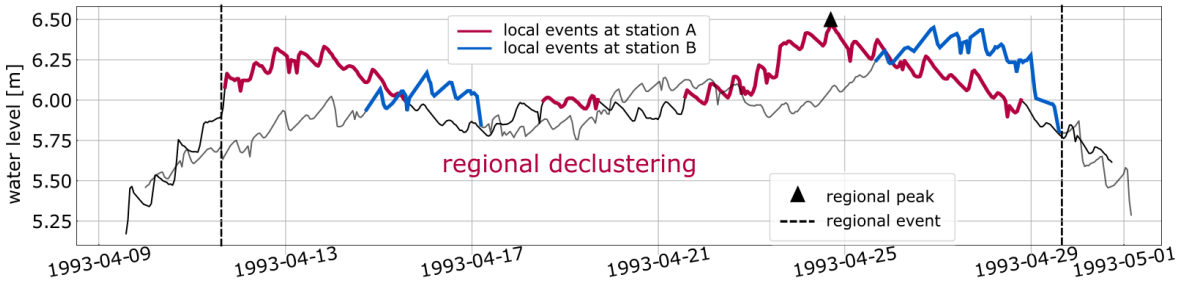


Figure 4.5: Regional event definition: declustering over the stations.

4.1.2 Workflow

In the afternoon of the first day of the workshop, a collaborative brainstorming session was conducted to refine and elucidate the problem aspect and specific question to tackle.

This process resulted in the identification of the major steps needed to solve the problem.

1. Feature engineering for event characterization: define some statistical metrics that effectively summarize the overall behaviour of high-water-level events. Specifically, the main goal was to develop some scores that differentiate between hydrological, local-scale meteorological, maritime, and astronomical drivers of these events.
2. Event dimension reduction: identify transformations of the water levels and/or the event to further reduce the event's dimensionality while retaining its meaningful properties.
3. Station and event classification: define the methodological steps for the systematic identification of each event driver(s) based on complementary climatological information and hydro-meteorological covariates; the objective of this step was to derive a classification metric that, on the one hand, recognizes the predominant type of extremes at each sampled location and, on the other hand, can be easily mapped in space at unsampled locations.
4. Model evaluation: estimate the spatial distribution of the summary features and classification metric(s) and evaluate their relevance as performance score for the hydrodynamical simulation and statistical reconstruction. The final goal was to test and possibly adapt the statistics defined at steps 1–3 to define both location-related and regional error measures describing the overall model performance.

Figure 4.6 displays a schematic representation created during the workshop to illustrate the interconnections between these main procedural steps.

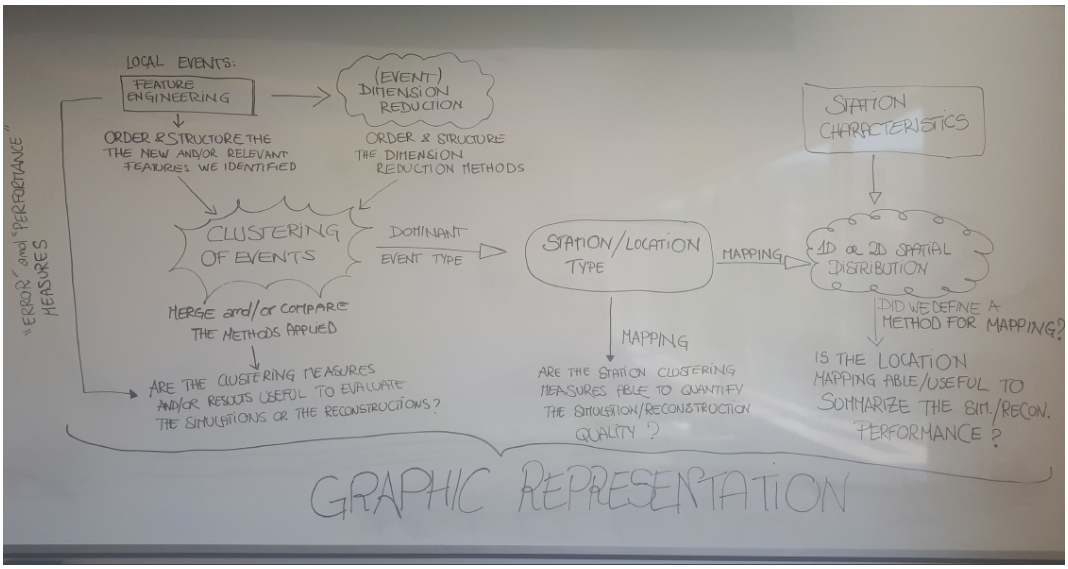


Figure 4.6: Problem solving workflow.

In the subsequent sections, the participants delineate the methodologies proposed to address the four aspects of the problem and their preliminary findings obtained during the workshop. The content is structured around the topic addressed by each of the three principal working groups: feature engineering, event and station clustering, and model evaluation metric.

4.2 Feature engineering for the event characterization

As a first step, we created new statistics to summarize several crucial event characteristics in a quantitative variable based on both the water levels and provided covariates. Within the dataset provided by ECCC, the group selected the covariates that could be treated as proxies for the process that may interact with the tides and river discharge to generate high water levels.

We first added a few additional statistics to those already available for the local and/or regional high-water-level events.

Peakness $[m]$: Ratio between the maximum and the mean filtered water level recorded during the event.

Event inter-time $[h]$: Time between the end of a regional event and the beginning of the next one.

Peak inter-time $[h]$: Time between two consecutive regional event peaks.

Then we identified two reference stations for the upstream and downstream signals influencing the water levels: Sorel-Lanoraie, in the fluvial section of the domain, and Sept-Îles, in the open Gulf of the St. Lawrence. At these two stations, the group decided to use the following signal decomposition based on five-yr harmonic analyses computed on annual sliding windows:

$$\mathrm{h}_{UP,t} = \mathrm{HYD} + f(Q_t) + FT_t, \qquad \text{upstream station - Sorel-Lanoraie,} \tag{4.1}$$

and

$$\mathrm{h}_{DW,t} = \mathrm{SEA} + f(S_t) + AT_t, \qquad \text{downstream station - Sept-Îles,} \tag{4.2}$$

where each variable is defined as follows (all units are meters):

HYD**:** sliding harmonic analysis intercept, representing the inter-annual variability of upstream water levels $[m]$ linked to medium- and long-term (e.g., decadal) hydrological cycles;

f($\mathrm{Q_t}$)**:** upstream water level residual (after the removal of HYD and the fortnightly tidal signal FT_t), proxy of the river streamflow that represents the hydrologically-induced water level variability;

SEA**:** sliding harmonic analysis intercept, representing the inter-annual variability of downstream water levels $[m]$ linked to medium- and long-term (e.g., decadal) oceanic cycles;

$f(S_t)$**:** low-passed downstream water level residual (after the removal of SEA and the astronomical tidal signal AT_t), proxy of the storm surge waves generated by large-scale meteorological systems.

Additionally, the river streamflow $[m^3/s]$ is directly collected upstream (Sorel-Lanoraie) and the wind speed $[km/h]$ recorded at 3 meteorological stations (Québec, Saint-Hubert and Montréal airports) were considered. The group decided not to use the wind direction for two reasons: the studied water level gauge locations were unevenly distributed between the St. Lawrence River's north and south shores, and the group wanted to start by approximating the domain on a single spatial dimension following the water line (centre of the river). The group computed the basic statistics (minimum, mean, maximum, standard deviation, and peak date) of the considered records observed over the 1970-2022 period.

4.2.1 Summaries for fixed-duration events

The duration of each event is a crucial characteristic in determining the causes of a high-water episode, and the declustering defined was aimed at preserving such a difference between events. For calculating certain statistics, however, the group found it impractical to work with regional events ranging from one hour to 1825 hours (roughly 2 months). During the workshop, some analyses were thus made by considering a time window of 13 hours before and after the local event peak. This fixed-length window corresponds roughly to a lunar day (24 hours and 50 min), and the procedure led to the identification of local events.

Moreover, to further homogenize the series between different locations and periods, the fixed-duration event analysis was conducted on scaled series: namely the filtered water level series was re-centred around the median and scaled using the median absolute deviation (MAD), to account for the fact that the distribution of water levels at some upstream locations is skewed.

Using the rescaled series and the fixed-length event set, the following statistics were also recomputed.

Amplitude $[m]$: Difference between maximum and minimum rescaled water levels observed in the 26h window. Note that in the context of tidal analysis, this variable is typically called "range."

Spatial influence $[-]$: Number of stations with water levels that exceed their marginal 95 percentile.

Normalized peakness $[m]$: Difference between the maximum and mean rescaled water levels during the 26h window, divided by the difference between the mean and minimum rescaled water levels during the fixed-length event; for each station j and event E_i this measure is computed as:

$$R_i = \frac{\max_{i \in E_i}(Y_{ij}) - \overline{Y}_{ij}}{\overline{Y}_{ij} - \min_{i \in E_i}(Y_{ij})},$$

where $\overline{Y}_{ij} = |n_i|^{-1} \sum_{i \in E_i} Y_{ij}$ is the event average rescaled water level, and n_i the event duration in hours. As the final step, R_i was centered by subtracting the median and the MAD of the R_i's was scaled (through computations at each site over the whole observation period, for both median and MAD).

Return levels $[m]$: A Generalized Pareto Distribution (GPD) was fitted to the 26h local event peak exceedances using the exceedances identified by ECCC, and the quantiles corresponding to various return periods were extracted from there (this means that the threshold level varies from site to site). When the number of observations was too small to reliably fit a GPD, the following plotting positions were considered to estimate the empirical probability distribution of the rescaled water level exceedances:

$$\text{rank}(Y_{ij})/(n_u + 1), \qquad \text{for } i = 1, \ldots, n_u,$$

where n_u indicates the number of events (exceedances). These empirical quantiles are then divided by 1.05 and mapped to the unit exponential scale for the clustering phase; this reduces the impact of extreme values.

Direction of the flow $[-]$: Kendall's rank correlation between the indices of site exceedances and the observed stations identifiers (1 to 16) for stations at which the cluster peak (i.e., the maximum observation during the time window used to define the event) exceeds the site-wise 0.9 quantile. This statistic is only defined when more than four sites are above the 0.9 quantile.

Annual timing $[-]$: Calendar day mapped to [0,1] and shifted by 58 days so that the year starts in March. Besides being sub-optimal for hydrological floods, this shift allows to separate roughly the 26h events connected to winter processes from those related to spring freshets. Conventional methods applied to the St. Lawrence watershed generally lead to the definition of hydrological years starting in periods of low flow (typically August or December).

Considering the fixed-duration event approach, the regional declustering was also re-defined using a geometrical approach: for each local extreme from a spatio-temporal plane event, consider the fixed-length extremes and the river km of the event (station location), measured as the distance from Pointe-Claire, as the *spatial* variable, and the event peak date as the *time*. Given the small time delay, we can aggregate the local events occurring in this time interval, to reconstruct a pattern that is expected to reproduce a regional event. We depict in Figure 4.7 the distribution of these patterns in the spatio-temporal plane.

When analyzing Figure 4.7 at a finer scale (e.g., on temporal windows of several days), three patterns can be distinguished, each corresponding to events of a different nature. The first type of pattern corresponds to monotone decreasing patterns, as schematized in Figure 4.8 (a). For such a pattern, the date of a local event decreases if its location along the St. Lawrence River increases. Therefore a monotone decreasing pattern corresponds to a regional event spreading *upstream* (Sept-Îles to Pointe-Claire). Conversely, a monotone increasing pattern [Figure 4.8 (b)] corresponds to a regional event which is spreading *downstream* (Pointe-Claire to Sept-Îles). Finally, a non-monotone pattern corresponds to a regional event that we call *complex* [Figure 4.8 (c)], which can, for instance, be the result of two events of opposite directions. According to this approach, the majority of the sampled

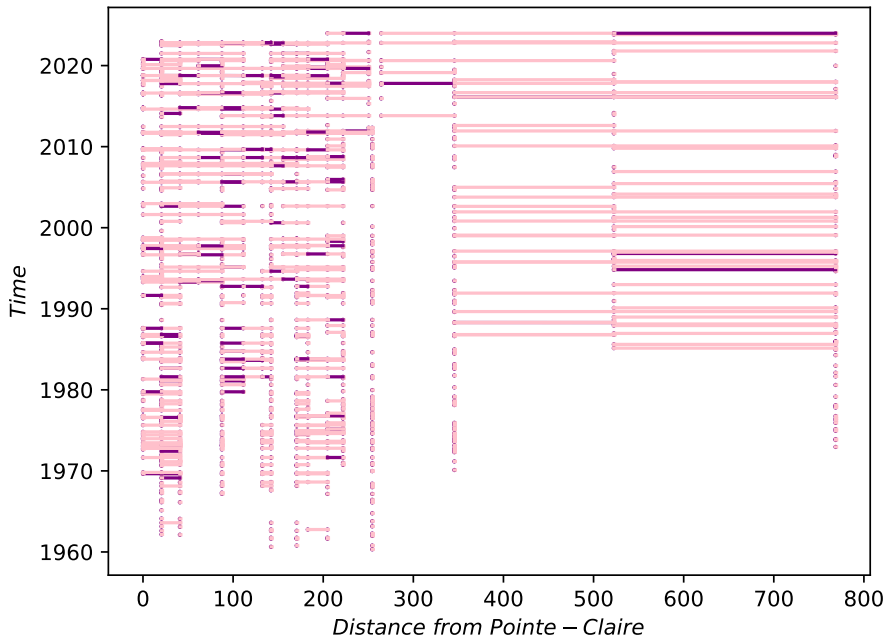


Figure 4.7: Spatio-temporal distribution of patterns obtained by aggregating local events occurring within a small delay of arbitrary amplitude (26h window).

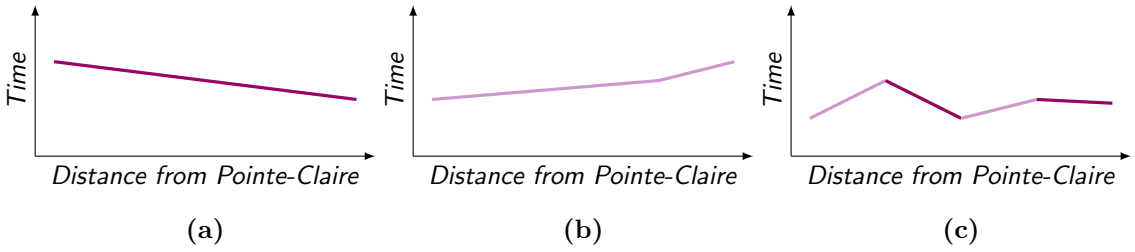


Figure 4.8: Analysis of patterns reproducing regional extreme events. (a) Monotone decreasing pattern corresponding to an upstream event. (b) Monotone increasing pattern corresponding to a downstream event. (c) Non-monotone pattern corresponding to a complex event.

events appear to be generated by hydrological processes originating upstream in the St. Lawrence basin.

Some measures aiming at quantifying event complexity were tested. They showed, however, a clear correlation – either positive or negative – with the peak amplitude. This confirms a known result in the literature: compound processes tend to damp the high-water-level peaks or generate the most intense extremes depending on the spatial location. It should be noted, however, that the uneven distribution of the station locations along the study domain introduces a strong bias in this analysis. While some methods could be improved by re-scaling the spatial dimension, we preferred to keep this descriptive method as a preliminary data exploration tool during the workshop.

Based on these descriptive analyses and discussions, participants decided to carry out investigations to define simple statistics capable of capturing the intensity of peaks at the same time as the direction of propagation of the high-water-level episodes. Event duration analysis, on the other hand, will only be possible after the refinement of the local and regional declustering criteria.

4.2.2 Dynamical characterization of local events: lagged regression parameters

On a regular basis ECCC uses interpolated water levels based on simplified dynamic relations, in order to reconstruct sparse or short water level series at a target location; this interpolation relies on sampled records taken at two reference stations. In an operational approach, the parameters used for the interpolation are in general estimated on an hourly basis through a lagged linear regression on the whole period for which records are available at the three locations. In this part of our work we adapted the ECCC methodology by estimating the interpolation parameters for each sampled local event. Hence the lagged regression parameters may be considered as statistics characterizing each event's dynamics as a function of signal variability at the two reference stations: Sorel-Lanoraie (*upstream* reference station) et Sept-Îles (*downstream* reference station).

Lagged regression model

Let $s_i \in 0, ..., N$ be a series of stations such that s_0 is the most (*upstream*) station, s_N the most (*downstream*) station, and the subscript list $0, ..., N$ is sorted according to the stations positions along the St. Lawrence River. The river is assumed to be a closed system between stations s_0 and s_N: thus for i such that $0 < i < N$ the water level at station s_i (denoted by wl_{s_i}) depends upon the water levels at stations s_0 and s_N (denoted respectively by wl_U and wl_D), with respective lags of τ_U and τ_D. The equation of the predictive model is thus

$$\mathrm{wl}_{s_i}(t) = \alpha + \beta \mathrm{wl}_U(t - \tau_U) + \gamma \mathrm{wl}_D(t - \tau_D), \qquad \forall i \in 1, .., N-1, \forall t \in \mathbb{R}. \tag{4.3}$$

The model parameters are computed for each extreme event at each station in two steps: first one estimates the temporal lags τ_U and τ_D through the maximization of the correlation between the variable $\mathrm{wl}_{s_i}(t)$ and the water levels $\mathrm{wl}_U(t-\tau_U)$ and $\mathrm{wl}_D(t-\tau_D)$. The coefficients α, β, and γ are then optimized by applying the least squares method. We have excluded events lasting less than two hours.

In order to evaluate quickly the quality of the estimated model, the prediction error was checked at each available station, excluding the two reference series (observed respectively at Sorel-Lanoraie and Sept-Îles). Figure 4.9 shows that the mean prediction error is less than 10cm for all stations upstream of Portneuf, when one considers the set of all sampled events. In some cases the approximation error is of the same order as the error in the observations measurements, i.e., a few centimeters. For downstream stations, however, the model exhibits greater prediction errors, in part because the height of the waves (and thus the signal amplitude during events) is higher in this section of the river than in the rest of the domain. An isolated instance of error larger than 50cm was obtained for certain events at the station of Vieux-Québec: this is probably due to a numerical error in the estimation of Equation (4.3). The participants did not have the time to solve this problem.

This lagged linear regression model allows one a good prediction of a station's water level in the case of an extreme event. It allows us to summarize extreme events using five parameters, which may then be compared to understand better the various event types. A physical interpretation of the two temporal parameters τ_U et τ_D will be used in the next section to identify the flow direction during a specific event.

4.2.3 Event direction

After some discussion, the participants agreed that the primary criterion for distinguishing events driven by hydrological processes from those driven by maritime processes should be the direction of water flow, with hydrological processes propagating from upstream to downstream and maritime processes from downstream to upstream.

Three methodological approaches have been tested for estimating the regional and fixed-length event direction.

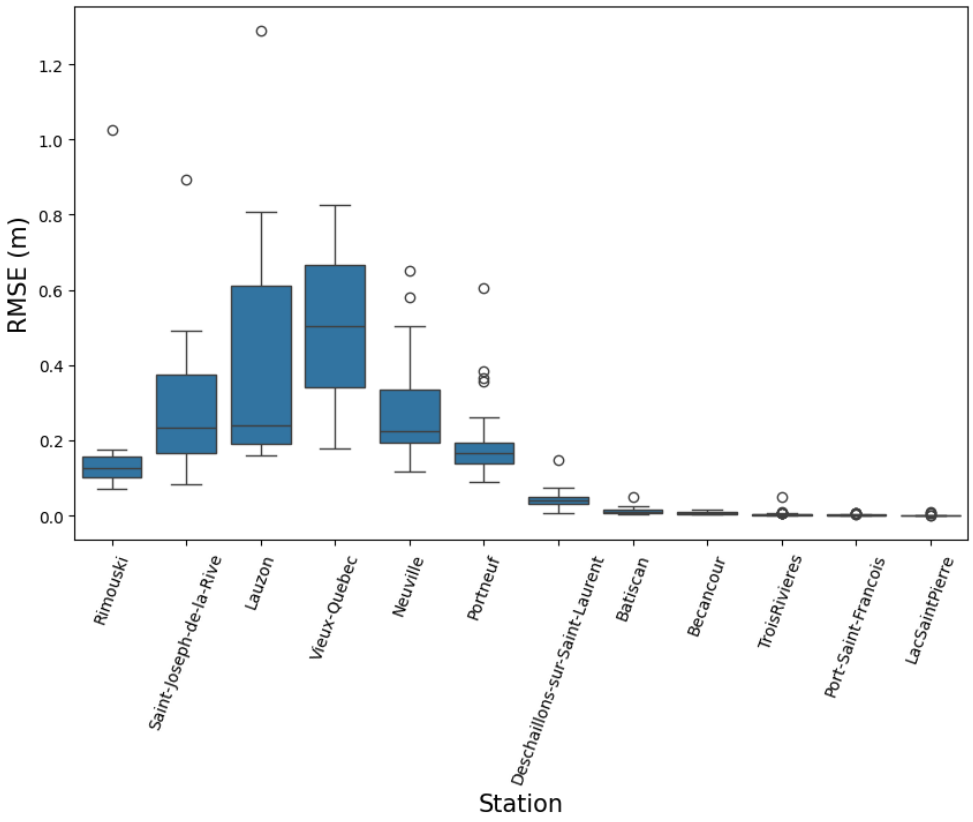


Figure 4.9: Boxplot by station of the model RMSE for every extreme event. The RMSE unit is the meter (m).

Peak direction - Computed for each regional event as the arithmetic average of the local event directions: for each regional event involving at least 2 stations, a value in $\{0, 0.5, 1\}$ is attributed to each peak based on the spatial location of the 2 adjacent peaks in time; Table 4.1 details the specific values taken by the direction statistics. The closer the peak direction is to 0, the more stations record an event flow going from upstream to downstream. Conversely, the closer the peak direction is to 1, the more the event is recorded flowing from downstream to upstream. Accordingly, it is possible to determine whether the wave of high water levels at a given station comes from upstream or downstream.

Table 4.1: Peak direction - Rules used to determine the direction value attributed to each peak within a given regional extreme event. Entries filled with dashes (–) indicate that the previous (or next) peak location cannot be determined within the event. The abbreviation "up" stands for upstream and "down" for downstream.

Previous peak location	up	–	down	down	down	up	–
Next peak location	down	down	up	–	down	up	–
Direction value	0	0	1	1	0.5	0.5	0.5

Correlation direction (`dirflow`) - Computed considering the rescaled water level in the 26h fixed-duration events: for each event, we selected the set of variables in the 26h time window that exceeded the 90th marginal percentile of the series for each site and computed Kendall's τ rank correlations between the direction of flow and the exceedances whenever at least four stations exhibited high water levels. The resulting measure takes positive values for events flowing from downstream to upstream while negative values represent events driven by upstream processes.

Dynamical model direction - Computed for each local event through the use of the lagged regression parameters: the signs of the two "lag parameters" τ_U and τ_D allow one to classify events as

coming from upstream (+,−) or downstream (−,+), caused by local meteorological conditions (+,+), or being influenced by the two directions of the waves (−,−). For instance, in the case of the (+,−)combination, if one wishes to predict the water level of a station at time t, then one needs the water level of the upstream station at time $t_U \leq t$ and the water level of the downstream station at time $t_D \geq t$. In other words an event characterized as a (+,−) event is an event originating upstream. The following table summarizes the four scenarios. This measure is interesting because it can be computed for any local event, not only for those having an impact on several stations. Indeed the direction of the water flow is determined through the lagged regression model, rather than through the lag with respect to the peaks of the other stations. Furthermore simple mathematical developments could allow one to incorporate the size of the lag into the definition of a continuous measure of direction.

lag parameters		Event origin
τ_U	τ_D	
+	+	Local
+	-	Upstream
-	+	Downstream
-	-	Compound (both directions)

Figure 4.10: Classification of events according to the flow direction, based on the signs of the lag parameters.

Some participants also highlighted the possibility of calculating simple direction statistics based on the geometric approach described in Section 4.2.1. Because of a lack of time, it was decided to retain the geometrical analysis as a descriptive method for data exploration.

4.3 Measures for the classification of events and stations

A team worked specifically on defining and testing various numerical approaches to discriminate between events of different natures based on the defined features. The goal is to identify distinct event clusters that would separate clearly in the feature space and be interpretable regarding the events' physical type and/or drivers. The teams further analyzed which proportion of overall events at each station belonged to the clusters that were found, i.e., which events and characteristics are important for modelling the high-water-level events at each particular station of the St. Lawrence River and fluvial estuary. Note that Montréal-Jetée, located at the Vieux-Port in Montréal, was excluded from the analysis due to its unique morphological features, which introduce some variability related to human activities into water-level time series.

4.3.1 Event clustering

In different teams, we explored various clustering methods to obtain a descriptive classification of 26h events. The objective is to define a clustering measure that can later be mapped to one or two spatial dimensions on the study domain. The explored methods included K-means, tree-based methods, and two types of mixture models based on Gaussian and Vine Copulas. PCA-based dimensionality reduction was initially considered for preparing the datasets but discarded because it did not significantly improve cluster separation and complicated the interpretation of the identified groups. The teams ultimately converged on using Gaussian Mixture Models (GMM), identifying either 3 or 4 clusters as optimal based on the computed Bayesian Information Criteria (BIC). Finally, the 4-cluster model was rejected in favour of the 3-cluster model, as the additional cluster did not enhance separation, and thus interpretation, of the events and stations. The 3-cluster model provided interpretable clusters when examining the separation along various variable axes.

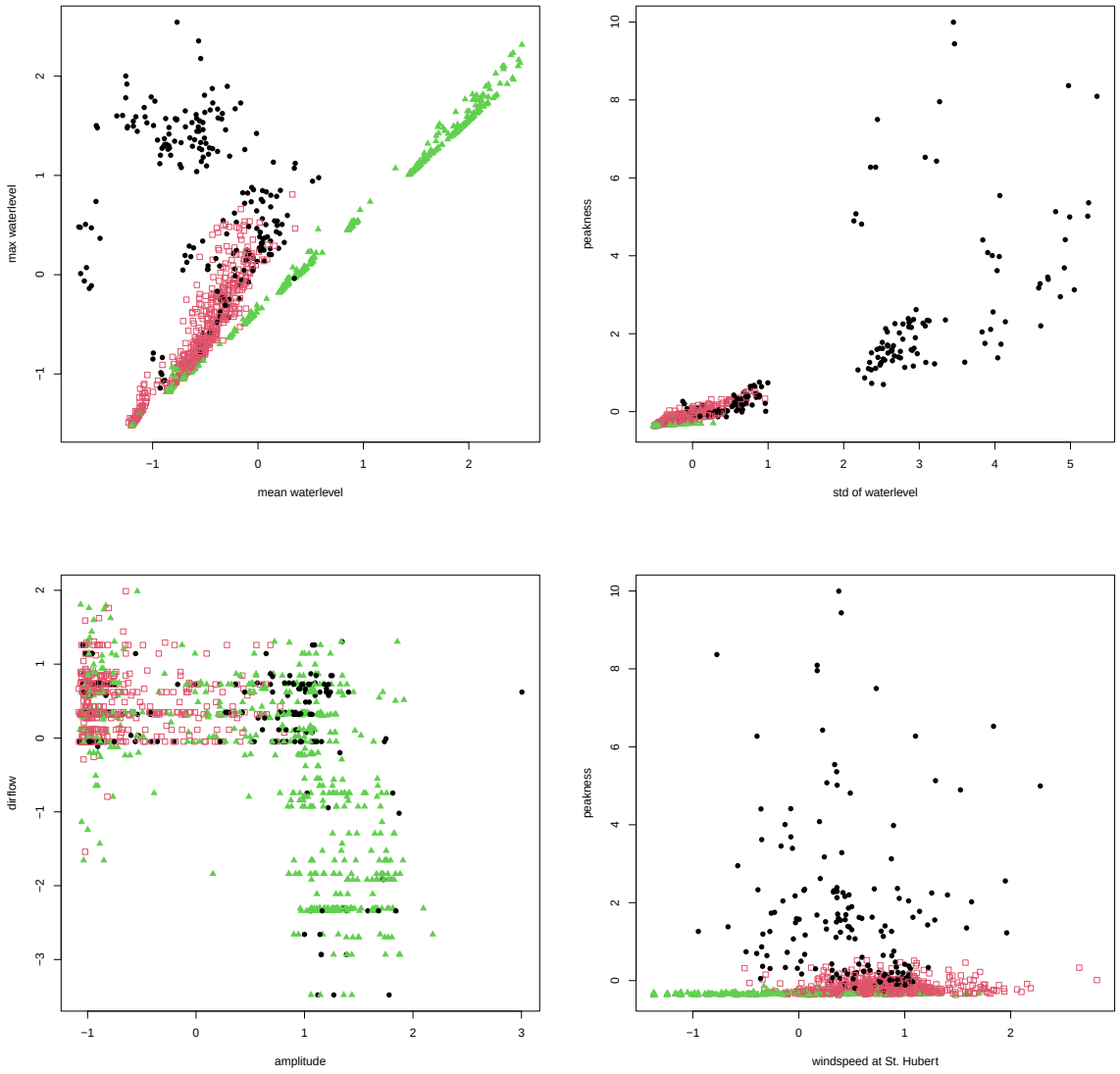


Figure 4.11: High-water-level event clusters (colours) along four selected pairs of event features (z-scores).

Figure 4.11 displays the scatterplots of the z-scores of four event feature pairs, indicating with the dot colour the group to which 26h events belong. In these scatterplots, the black points represent the events with high peakness, relatively small mean water level, and large water level standard deviation, indicative of events that overcome a tidal cycle (e.g., semi-diurnal cycles in the fluvial estuary section of the domain). In fact, it can be seen that there is a precise positioning of the black dots in the positive plane of the direction-amplitude scatter in the bottom-left panel of Figure 4.11. This indicates that these events mostly have a downstream-to-upstream direction. As an exception, a smaller group of events in the black cluster has negative direction and strong (positive) amplitude values. This suggests that a sub-cluster of the black events flowing from upstream to downstream can be identified. In this sense, such a sub-cluster could be interpreted as events driven by truly intense and more localized hydrometeorological processes (e.g., intense rain events occurring on specific sub-watersheds inducing local floods in tributaries).

The green-coded cluster of events displays low peakness and variability [Fig. 4.11 top-right and bottom-right] and widely varying amplitude values [Fig. 4.11 bottom-left], suggesting a possible influence of astronomical tides on this group. Interestingly, the red cluster presents a net separation from the other two groups in terms of correlation direction (`dirflow`) [Fig. 4.11 bottom-left], suggesting the association of these events with river-driven phenomena.

Based on these results, we concluded that the 3 mixture components achieve very good separation in feature space and are reasonably interpretable. Note, however, that caution should be used when interpreting the separation between the black and red clusters since 2 sub-clusters seem visible for the black-group events, one of which may be appropriately grouped with the red cluster. This distinction is particularly evident in Figure 4.11 from the fact that we get very clear separation within the black cluster when looking at the standard deviation of the water level in the upper right panel of Figure 4.11, and to a lesser extent in the upper left panel. Increasing the flexibility of the GMMs by allowing more components did not facilitate this separation. Further investigation could involve fitting a more flexible model, such as a restricted vine copula, initialized with the current cluster assignment to achieve the desired separation without imposing such separation manually. Alternatively, one could consider fitting a larger number of GMM components and subsequently grouping these components into clusters to achieve the desired separation.

4.3.2 Stations clustering

Figure 4.12 shows the probabilities of an event belonging to one of the three identified clusters of events for each station considered in the application. The change in the proportions of occurrences of the

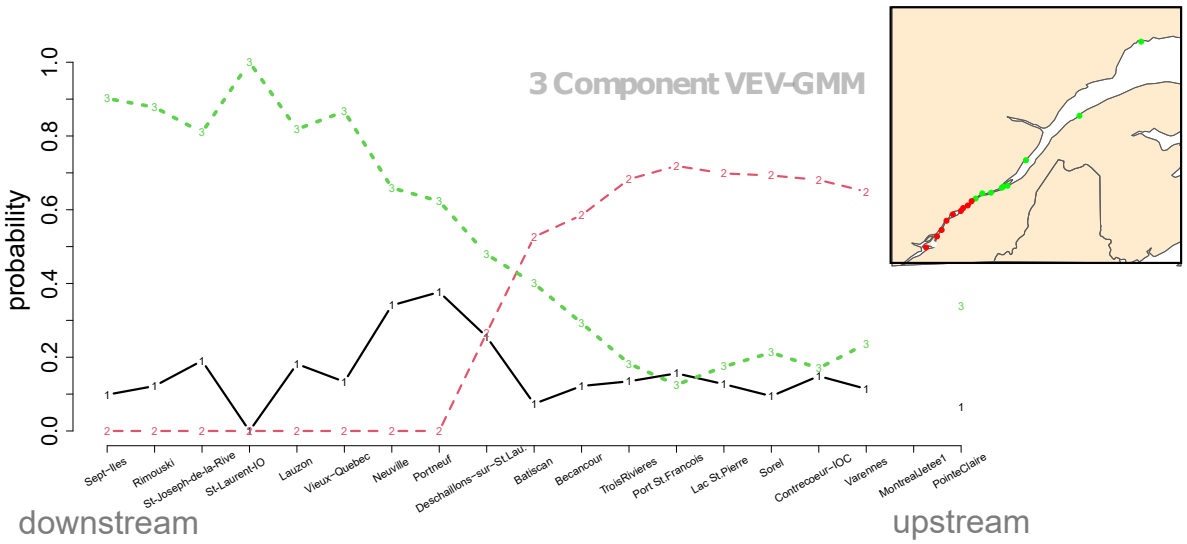


Figure 4.12: Cluster assignment proportions by station. Stations are ordered (from left to right) from downstream (Sept-Îles) to upstream (Pointe-Claire). Cluster colours are as in the previous figure.

3 clusters from upstream to downstream supports the interpretation that red events are influenced by river flow, as this influence is only relevant between Portneuf and Pointe-Claire (the most upstream station available). Similarly, the significant decrease in the proportions of green-cluster events when transitioning from the estuary (left) through the fluvial-estuary to the fluvial section (right) suggests that this cluster mainly represents the events generated by sea-related processes. Finally, the large proportion of black-cluster events in the central region of the fluvial estuary (i.e., between Lauzon and Deschallons-sur-St-Laurent) indicates that this cluster likely represents events induced by combined fluvial and maritime or astronomical conditions. This is likely due to the simultaneous occurrence of high streamflow and storm surges or elevated astronomical tides. As anticipated, the relatively gradual

variation in event proportions along the river suggests the feasibility of interpolating these proportions from stations with known cluster assignments in order to estimate them at ungauged locations. We suggest using the GMM classification probability estimates as a candidate measure to map the dominant type of high-water-level episodes over the study region.

4.4 Measures describing the model performance

This section reports the discussion and methodological exploration carried out to evaluate and possibly rank the statistical reconstruction and hydrodynamic model simulations. The evaluation target was the ability and effectiveness of the model in reproducing high water levels across various parameters. The analysis aimed to determine the relative performance of these methods in accurately simulating extreme hydrological events.

4.4.1 Error measures for hydrodynamic model evaluation

We started by considering the accuracy of the H2D2 hydrodynamic model simulations provided by ECCC for 3 high-water-level events. The shallow water model H2D2, developed at the *Institut National de la Recherche Scientifique* (INRS), is used by the Canadian Meteorological Centre (CMC) to produce daily water level forecasts for the study domain. H2D2 solves the Saint-Venant equations using a two-dimensional (2D) finite element discretization. The water levels at Saint-Joseph-de-la-Rive and the flows of the major tributaries draining into the St. Lawrence upstream section define the boundary conditions for the models. The other simulation inputs, including hourly wind and ice data, are based on the ECCC's Regional Deterministic Prediction System (RDPS) reanalysis outputs.

As shown in ECCC's presentation of the problem, there is a clear spatial pattern in the errors observed for the example simulations due to biases in both the signal mean and its amplitude. Overall, this results in a mean absolute error (MAE) between the observed and simulated series that decreases when moving from upstream to downstream. Moreover, a phase shift has been observed in the provided example, and is shown in Figure 4.13: at the St-Joseph-de-la-Rive station, in the first row, we can see both an error in the signal amplitude, most visible in water level maximum and minimum peaks [Fig. 4.13 top left], and a clear residual pattern due to a signal phase shift [Fig. 4.13 top center]. When analyzing the residuals between the two series as a function of the observed water level, the phase shift results in a doughnut pattern showing in the top-right panel of Figure 4.13: the systematic shift implies that the difference between series is never zero when the water level takes values close to the mean monthly water level used to filter the series. Therefore, we used this result to define a preliminary error measure that evaluates the phase shift by measuring the radius of the centred circle that does not report any data point.

As a second step, we analyzed how the phase shift behaves for the other stations and simulated events. Considering that the water level signal becomes non-stationary when moving upstream, the phase shift estimation should thus be reformulated to account for the squashed ellipsoid shape observed in the middle right panel of Figure 4.13. More specifically, the middle panel of Figure 4.13 also shows a signal phase shift for the Deschaillons-sur-Saint-Laurent station location, but this shift is less pronounced and skewedly distributed on the standardized water level values. Moreover a mean water level bias becomes evident. Considering that the harmonic patterns of the simulation are more symmetric than those of the observed data, the residual values clustered around two values: -0.25 and 0.75. This can be seen by looking at the range in the middle panel, or at the spread of points in the rightmost plot of the second row of Figure 4.13.

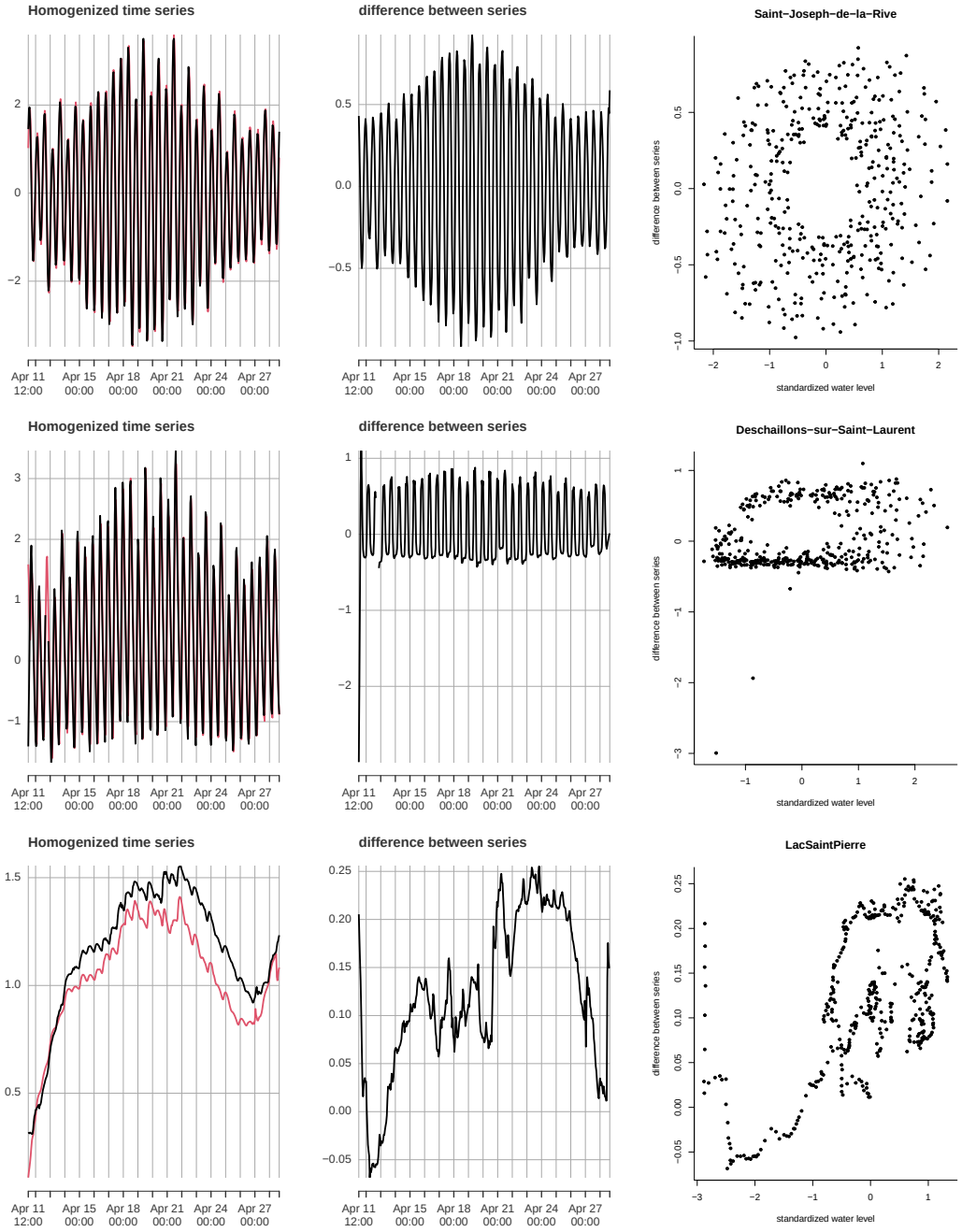


Figure 4.13: Hydrodynamic reconstruction at St-Joseph-de-la-Rive (top), Deschaillons-sur-Saint-Laurent (middle) and Lac-Saint-Pierre (bottom) for the extreme event of April 2011: time series of homogenized simulated and observed water levels (left), time series of differences (middle) and difference against the scaled observed homogenized water levels.

Finally, considering the residual for locations further upstream [e.g., Figure 4.13 bottom row, for the Lac-Saint-Pierre station], we noticed a clear temporal autocorrelation of the residuals, which is, in turn, explained by the correlation of the residual with the filtered water level series [Figure 4.13 bottom right]. Estimating the phase shift is challenging in this case, and the error measures should focus on the stationary bias affecting the mean simulated water levels. In this context, a wide range of simulation errors can be observed for a single event simulated across the entire domain. Analyzing more simulated events would be necessary to identify the phase shift's significant spatial and temporal patterns, which could be used to classify the events.

To define a single summary statistic that accounts for the different patterns observed in the previous examples, we consider a location-scale transformation with scale $a > 0$ and shift $b \in \mathbb{R}$ for the difference between the filtered simulated series S_i and the filtered observed water level:

$$\sum_{i=1}^{n} |a \cdot (S_i - b) - Y_i|. \tag{4.4}$$

This transformation can be used to estimate the following quantities.

1. `Radius`: The radius of the circle identified by the closest observation in the scatterplot formed by the standardized water series and the difference between standardized predictions and water series. Large values are indicative of phase shift.
2. `Asymmetry`: The ratio of interquartile ranges for the positive and negative residuals, capturing the asymmetry of the predictions. Values larger than unity indicate more spread in overpredictions (positive residuals) relative to underpredictions.
3. `Shift`: The location coefficient b is a robust estimator of the median bias and represents the amount of series shifting relative to the reconstruction.
4. `Scaling`: The scale coefficient a, which mostly captures the difference in the signal amplitude. Reconstructed series that are less variable than observations (e.g., since they do not capture extremes) have coefficients smaller than one.

Figure 4.14 reports the values of the proposed Equation (4.4) decomposition computed for each station on the 3 available simulations. Using the proposed measure, it is possible to determine which stations

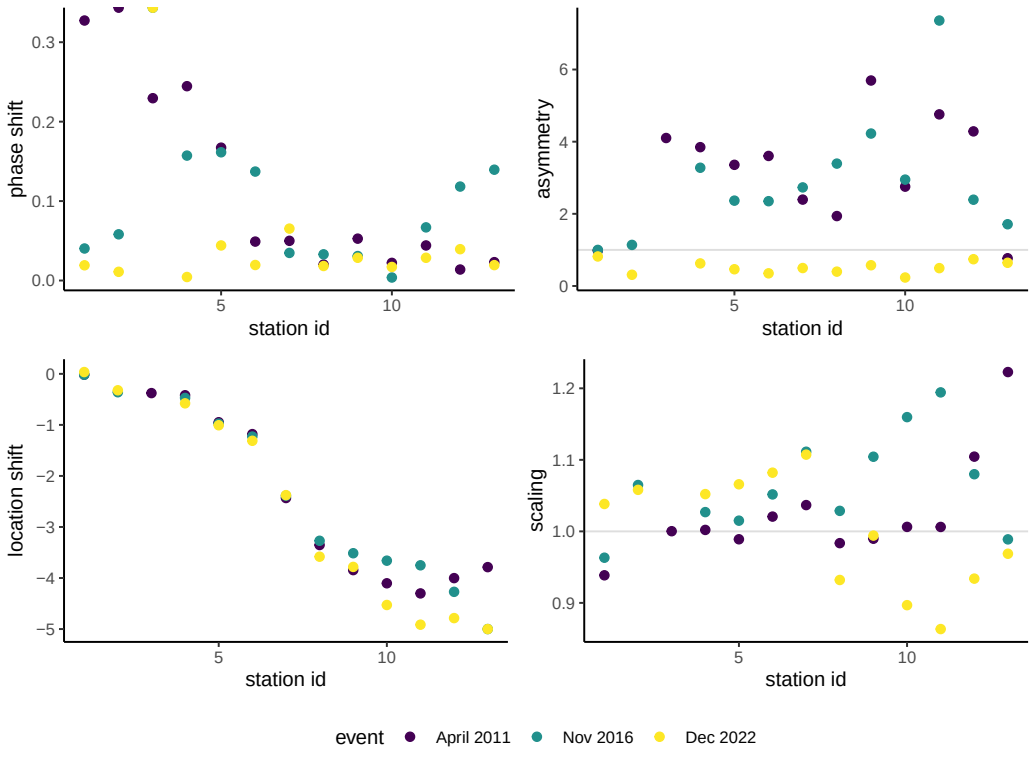


Figure 4.14: Statistical summaries for each station and simulated event. Clockwise: phase shift for radius, asymmetry measure, location, and scale. Stations are ordered downstream to upstream from top to bottom.

present biases in the mean due to a systematic decrease of the location shift values when moving upstream [Fig. 4.14 bottom left]. Likewise, it is possible to detect that signal dephasing mostly affects

the downstream stations for the spring event of April 2011, but the spatially variable phase shifts are visible for the other two simulated events, occurring in fall and winter [Fig. 4.14 top left]. This result is particularly interesting since the spring event is most likely triggered by a spring freshet and important upstream streamflow values. The asymmetry measure reported in the top right panel of Figure 4.14 further suggests that the positive prediction errors cover a larger range of values than underestimation errors, especially for upstream stations, for the April 2011 and November 2016 events. Finally, the scaling statistic reported in the lower right panel of Figure 4.14 did not show a consistent spatial pattern for the 3 events. A larger score variability, however, is observed for upstream stations (scaling scores between 0.8 and 1.25). This consideration should be combined with the fact that parameter A correlates as much with bias as with accuracy in terms of the amplitude of water level predictions. Consequently, the statistic seems challenging to interpret with only the 3 example simulations provided by ECCC.

According to the smooth patterns observed in Figure 4.14, it seems possible to interpolate the parameters of Equation (4.4) to ungauged locations with reasonable accuracy. In that sense, ECCC should consider decomposing the prediction error into the four proposed statistics to answer the problem's second question.

4.4.2 Measures based on Principal Component Analysis (PCA)

The other direction explored to evaluate the reconstruction and hydrodynamic models during the high-water-level events is based on comparing the ensemble of event features estimated on the observed and simulated series. With nearly all variables described in Section 2, each extreme event was characterized by 55 features. One way of summarizing the information contained in this large set of statistics is to use dimension-reduction methods such as Principal Component Analysis (PCA). In simple terms, PCA is a linear transformation technique that allows one to project a large set of covariates into a lower-dimensional space while retaining as much variability as possible. These components are defined by the eigenvectors of the original data covariance matrix, and they thus form an orthonormal basis.

In our application, once the features were scaled using standard z-scores, PCA was applied at each station to the set of features estimated for each local event. By examining the variance explained by each principal component, we determined that the first 5 components generally accounted for slightly more than 70% of the 55-feature variance. The linear combinations used to compute the first five components were thus stored and subsequently applied to the corresponding features of the high-water-level events simulated by the harmonic model in the first configuration. This allowed the construction of an evaluation metric comparing the 5 principal component values in observed and simulated events. More specifically, the differences $\Delta PC^{(i)} = PC^{(i)}_{obs} - PC^{(i)}_{sim}$ for $i = 1, 2, 3, 4, 5$, as well as the L_2 norm of the 5-element vector of the $\Delta PC^{(i)}$ values were considered in our tests for a direct comparison of the feature combinations between the simulated and observed events. Figure 4.15 shows the computed $\Delta PC^{(i)}$ values for an example event. In this example, the largest differences are found for the fifth component ($\Delta PC^{(5)} = 1.9963$), which can be mostly interpreted as a difference in the min and peakness, which is itself computed from the max and mean. The error seen in the figure, mostly bias, strongly affects the min, mean, and max, and hence the fifth component.

In summary, by analyzing the coefficients of the linear combinations, particularly those associated with the first components, we can infer which combinations of features are well reproduced in the simulations and which combinations present larger errors. It would be interesting to explore this method using only the key water level features: minimum, mean, maximum, standard deviation, amplitude, peakness. The principal components might then be easier to interpret.

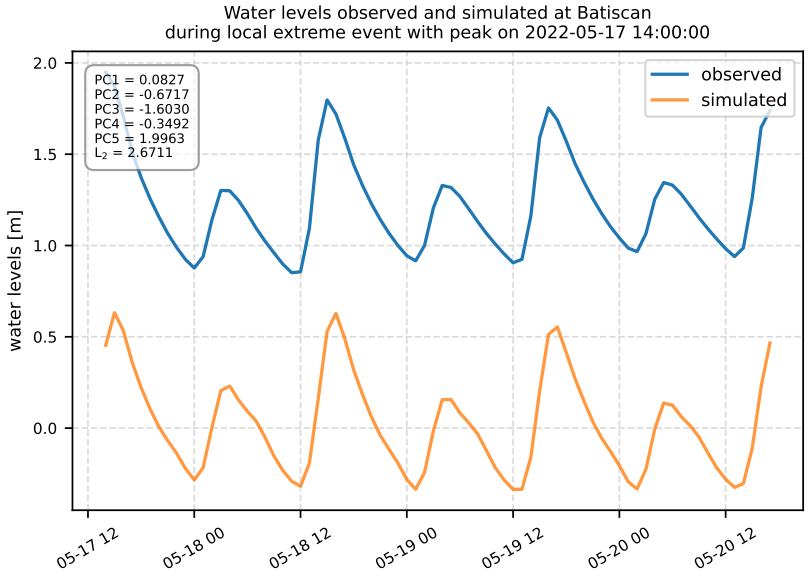


Figure 4.15: Observed and simulated water levels during an extreme event in May 2022 at the Batiscan station. The model used was the harmonic model configuration 1.

4.5 Concluding remarks

We sought to define some statistical metrics that characterize the dominant typology of events at each location and subsequently assess the spatial distribution of this metric along the river continuum. To this end, we first created useful statistics to describe key event features during the workshop, with special attention to event direction proxies. Subsequently, we explored some possible methodologies for combining the event characteristics into summary measures that can be used in event classification. Finally, we investigated how the summary measures could be adapted to the definition of evaluation scores for different types of event predictions and water level numerical simulations. Our preliminary tests indicate that the adequate estimation of basic event statistics, such as the duration and direction, is sufficient for the event classification via conventional clustering methods. The clustering measures can then be used for mapping the classification at unsampled locations since they present fairly smooth variability across the provided stations.

Regarding the numerical model evaluation, it was found easier to define summary metrics based directly on the water level prediction residuals than to evaluate the error in the estimated event statistics or summary features. In doing so, we defined some evaluation scores capable of assessing the overall performance of numerical models in terms of the phase and magnitude of prediction errors. These scores also show a clear spatial distribution between stations and thus seem easy to interpolate spatially. A more extensive application on a larger sample of simulated events, however, is required to validate this hypothesis. It is anticipated that Environment and Climate Change Canada (ECCC) will utilize some of our findings in the coming months to address this question.

In addition to those described in the report, one notable outcome of the week is the creation of an elegant interactive dashboard (using `plotly` in `python`). It visually encapsulates some of our findings and could be easily extended by ECCC to include other statistics. Some screenshots of this dashboard are shown in Figure 4.16.

In summary, we consider that the workshop allowed fruitful continuous exchanges and produced valuable insights for ECCC. From a methodological point of view, the data analysis pipeline we

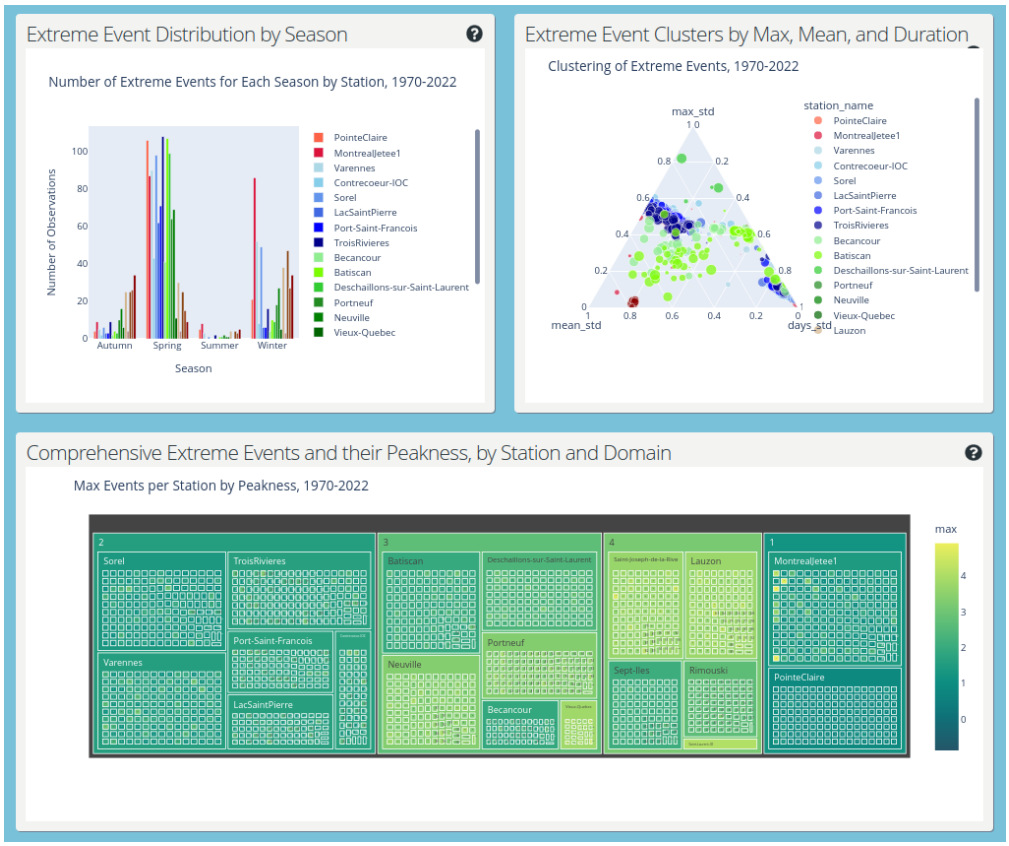


Figure 4.16: Screenshot of the interactive dashboard created during the workshop.

created during the discussions was our most impactful product, and the entire analysis itself helped us understand the essential problem.

Finally, we were very fortunate to have a team with members from diverse backgrounds. Despite our shared passion for mathematics and statistics, the best part of the week may have been getting to know one another!

5 IATA: Estimating turbulence duration and the likelihood of turbulence occurring

C. Sean Bohun [a]
Ismael El Yassini [b]
Cecilia Fan [c]
Ahmed Harrabi [d]
Pierre Houedry [e]
Slim Ibrahim [f]
Wuding Li [g]
Michael R. Lindstrom [h]
Ronnie Liu [c]
Juliana Schulz [c]
Lingyi Yang [i]
Dasen Ye [c]

[a] *Ontario Tech University*
[b] *University of Waterloo*
[c] *HEC Montréal*
[d] *Politecnico di Milano, Milan*
[e] *Université de Rennes*
[f] *University of Victoria*
[g] *Université de Montréal*
[h] *The University of Texas Rio Grande Valley*
[i] *University of Oxford*

November 2024
Les Cahiers du GERAD

Abstract: *At the 14th Montreal Industrial Problem Solving Workshop, the International Air Transport Association (IATA) presented an extensive dataset of measured turbulence values (Eddy Dissipation Rate) collected by thousands of aircraft over several years. With these data, IATA requested estimates for how long turbulence persists and for the probability of a given aircraft encountering turbulence given historical and live data. Through a series of approaches, we present preliminary quantitative findings and a preliminary toy model for the development of turbulence.*

5.1 Introduction

The Federal Aviation Administration (FAA) defines turbulence as the irregular motion of an aircraft in flight, caused by a rapid variation of atmospheric wind velocities (FAA-H-8083-28, Aviation Weather Handbook, 2022). According to [1], turbulence is estimated to cost roughly 200 million USD annually in the United States. Turbulence is also the leading cause of injuries to cabin crew and passengers in non-fatal accidents (FAA). Aircraft turbulence can also cause brand damage and contribute to fear of flying.

There are various causes of turbulence, such as convective currents, obstructions in the wind flow, and wind shear (FAA-H-8083-28, Aviation Weather Handbook, 2022). Some forms of turbulence are more predictable, such as that caused by mountain waves or storms, and in some instances an aircraft can avoid such turbulent events. Clear-air turbulence (CAT), which is an invisible form of turbulence, is a more difficult phenomenon [1]. According to a recent study [1], CAT has increased in both frequency and duration over the past years and is expected to continue to do so in response to climate change. In particular, [2] projects a 149% increase in the frequency of severe turbulence events.

It is challenging to predict and manage aircraft turbulence for several reasons. First, pilot reports are generally subjective as different sized aircraft will experience turbulence differently. CAT, in particular, cannot be detected by weather radar. In addition, forecasts are often hours long and inaccurate. As a step towards addressing these issues, the International Air Transport Association (IATA) Turbulence Aware tool was created as a data-driven tool for reporting and managing turbulence in real time. Recent technical advancements have equipped aircraft with the ability to measure objectively the state of the atmosphere around the aircraft and report this data in real time. In particular, turbulence is measured using the Eddy Dissipation Rate (EDR), which is a turbulence intensity metric measuring the state of the atmosphere around an aircraft in flight. It typically takes on values ranging from 0 to 1 $m^{2/3}/s$. The higher the dissipation rate, the higher the atmospheric turbulence. Note that the EDR is an aircraft-independent absolute value and thus allows one to measure turbulence objectively.

With rich data collected through the IATA Turbulence Aware tool, several questions related to turbulence can now be further studied. Three main questions were explored at the Fourteenth Montreal Industrial Problem Solving Workshop (IPSW) [3]. Here they are.

1. How long does turbulence last? Furthermore, does the duration of turbulence depend on the time of the year, altitude, wind speed, temperature, or any other factors? Meteorologists have been attempting to answer this question for a very long time. With the IATA Turbulence Aware tool, it is the first time that a rich enough dataset of objective turbulence measures is available, allowing us to address this question.
2. Given the aircraft's location, trajectory, and speed, can we determine the likelihood of turbulence ahead of the aircraft based on live and historical turbulence data? Can the wind speed and direction, temperature and location, historical and live data, all be taken into account in determining the likelihood of turbulence ahead? Such information would be invaluable, as it could be dispatched to pilots in real-time thus allowing them to make informed tactical decisions and potentially avoid turbulence.

3. Can we model the number of thermal-based turbulent events based on the EDR measures and cloud cover data at low levels? This is particularly relevant for the descent phase of an aircraft, which tends to be quite "bumpy." Addressing this question would be helpful for pilots, as there is a lack of guidance and forecasts for the descent phase of flights.

In Section 5.3, we present our analysis for solving problem 1; in Section 5.4, we study problem 2; and in Section 5.5, we study a toy model for the physical phenomena and qualitative behaviours of turbulence. Problem 3 was not studied, due to time constraints. A conclusion of this work is given in Section 5.6.

5.2 Dataset

The IATA Turbulence Aware tool allows for objective measures of the intensity of turbulence via the EDR measured from an aircraft sensor. While continuously monitored, the actual reported values provide the average and peak EDR value over a one-minute time span. In addition to the EDR measures, each record provides the 4D position of the aircraft, along with measurements of the wind speed and direction, temperature, as well as information on the flight, including flight number, the departure airport and the arrival airport. Note that whenever an aicraft is actively in a turbulence state (i.e., the EDR measure is greater than 0), measurements are recorded each minute. Otherwise, "heartbeat" measures are taken at 15-minute intervals.

The dataset available at the IPSW included the following fields:

measurement_observationTime: a timestamp of format "yyyy-mm-dd hh:mm:ss" in UTC time;

measurement_altitude: the altitude in feet, based on air pressure readings;

measurement_latitude: the latitude in degrees;

measurement_longitude: the longitude in degrees;

measurement_temperature: the temperature in degrees Celsius;

measurement_wind_speed: the wind speed in knots;

measurement_wind_direction: the direction of the wind in degrees clockwise relative to magnetic north;

measurement_edr_peak_value: the peak measurement of EDR over an interval of one-minute length;

metadata_tafi: the TAFI ID, a unique identifier for each flight that takes off (two distinct flights between the same airports with the same flight number have different identifiers) to track data across a given flight; and

other fields, such as departure/arrival airport, mean EDR reading, and peak time within the one-minute interval, were also included in some datasets.

Conversions were made to convert timestamps to times in seconds relative to a fixed date. Figure 5.1 depicts a Python `pandas` dataframe storing one set of records.

5.3 Approaches for Problem 1

5.3.1 Clustering

Data preparation and use

The clustering analysis was carried out on the dataset '**Top city pairs by no of flights and peak EDR and data for three flights.csv**', containing the most popular flights between pairs of cities spanning approximately 120 days.

Additional data were computed from the given data for the analysis. Firstly, Cartesian coordinates were computed to represent datapoints. Let (a, α, β) denote the altitude, longitude, and latitude of a

	measurement_observationTime	measurement_altitude	measurement_latitude	\
0	2024-02-12 19:08:10	5200	33.693	
1	2024-02-12 19:09:10	5100	33.694	
2	2024-02-12 19:10:10	4100	33.694	

	measurement_longitude	measurement_temperature	measurement_wind_speed	\
0	-84.717	11.0	30	
1	-84.656	11.0	31	
2	-84.595	11.5	32	

	measurement_wind_direction	measurement_edr_algorithm	\
0	166	NCAR_V3	
1	161	NCAR_V3	
2	139	NCAR_V3	

	measurement_edr_peak_value	measurement_edr_mean_value	...	\
0	0.04	0.02	...	
1	0.08	0.04	...	
2	0.06	0.02	...	

	metadata_id	metadata_tafi
0	4579b2ba-0000-4e31-a244-ddc6d7cccfa2	001f994f-9a4c-43e4-9273-32bcefb097f1
1	084321ac-93d8-465a-9527-df10f06b07f5	001f994f-9a4c-43e4-9273-32bcefb097f1
2	ecded18a-6526-44e9-8e3b-c44df0aaf9c7	001f994f-9a4c-43e4-9273-32bcefb097f1

Figure 5.1: The first three records of one particular dataset – individual datasets examined may have had more or fewer fields.

measurement after converting degrees to radians. Then we compute

$$x = (a+R)\sin\left(\frac{\pi}{2}-\beta\right)\cos\alpha$$
$$y = (a+R)\sin\left(\frac{\pi}{2}-\beta\right)\sin\alpha$$
$$z = (a+R)\cos\left(\frac{\pi}{2}-\beta\right)$$

where $R = 2.093 \times 10^7$ ft is the mean earth radius (we have used a spherical earth approximation). From Cartesian coordinates, distances can be readily computed.

From the wind speed and direction, we can infer information about the vector form of the wind velocity. Lacking specific data about the wind components up or down, for the sake of this analysis, we model the wind velocity U as having only two components $U = (U_1, U_2)$ in the (local) plane while varying in $\mathbb{R}^3$. With a wind speed w and direction θ, we define

$$U = (w\sin\theta, w\cos\theta)$$

to give the East- and North-components of the wind velocity.

We also added various gradient information for fields such as temperature, a scalar, and velocity, a vector. Let $f : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^n$ be assumed smooth: f represents the temperature (with $n = 1$) or the wind velocity (with $n = 2$). Let $x : \mathbb{R} \to \mathbb{R}^3$ represent the airplane's trajectory. We concern ourselves with the evolution and changes in $F(t) = f(x(t), t)$. This corresponds to the measurements taken within the aircraft during its flight at position $x(t)$ and time t.

Let two readings be taken at times s_1 and s_2 with $s_1 < s_2$. Then for $i = 1, ..., n$,

$$F_i(s_2) - F_i(s_1) = (s_2 - s_1)F_i'(r_i)$$

$$= (s_2 - s_1)(\nabla f_i(x(r_i), r_i) \cdot x'(r_i) + f_{i,t}(x(r_i), r_i)) \tag{5.1}$$

where $r_i \in [s_1, s_2]$. Similarly, for $j = 1, 2, 3$,

$$||x_j(s_2) - x_j(s_1)|| = (s_2 - s_1)||x_j'(r_j^*)||$$

where $r_j^* \in [s_1, s_2]$.

We now consider two different discrete difference quotients: we can divide Equation (5.1) by the time interval $s_2 - s_1$ or by the distance between measurement points,

$$||x(s_2) - x(s_1)|| = \sqrt{\sum_{j=1}^{3} (s_2 - s_1)^2 ||x_j'(r_j^*)||^2}.$$

Dividing first by $s_2 - s_1$ and then by $||x(s_2) - x(s_1)||$, we obtain

$$\Delta_2 F_i = \nabla f_i(x(r_i), r_i) \cdot x'(r_i) + f_{i,t}(x(r_i), r_i) \tag{5.2}$$

$$\Delta_1 F_i = \nabla f_i(x(r_i), r_i) \cdot \frac{x'(r_i)}{\sqrt{\sum_{j=1}^{3} ||x_j'(r_j^*)||^2}} + \frac{f_{i,t}(x(r_i), r_i)}{\sqrt{\sum_{j=1}^{3} ||x_j'(r_j^*)||^2}}. \tag{5.3}$$

The first formula is effectively the total rate of change of F_i with respect to t: it describes how f_i changes from the airplane's perspective. The second formula is more or less a directional derivative of f_i in the direction of motion, with an additional component. In the limit, with $s_2 \downarrow s_1$, and replacing s_1 with s, we obtain

$$\lim \Delta_2 F = F'(s) = \nabla f(x(s), s) \cdot x'(s) + f_{i,t}(x(s), s)$$

$$\lim \Delta_1 F = \nabla f(x(s), s) \cdot \hat{v} + \frac{f_{i,t}(x(s), s)}{||v||}$$

where $v := x'(s)$ and $\hat{v} = v/||v|| hold$. This is written in vector form and when $n > 1$, ∇f is a Jacobian.

In the clustering, we opt to compute Δ_2 for the temperature and velocity and we focus on the magnitudes, computing over $[s_1, s_2]$,

$$|\Delta_x T| = \frac{|T(x(s_2), s_2) - T(x(s_1), s_1)|}{||x(s_2) - x(s_1)||} \tag{5.4}$$

$$|\Delta_x U| = \frac{||U(x(s_2), s_2) - U(x(s_1), s_1)||}{||x(s_2) - x(s_1)||}. \tag{5.5}$$

By sorting the data according to the TAFI ID criterion and then the time criterion, the generalized gradients can be computed. We first filter out all records whose TAFI ID appears only once since differences cannot be computed for these records. For those with two or more TAFI IDs, we compute the gradients; since differencing reduces the length of the dataset and to avoid losing data, the last gradient for a given TAFI ID is set equal to the second last gradient for that ID.

Identifying turbulence events

Because a small aircraft will experience noticeable instability at an EDR above 0.13, we select all data with EDR-values above 0.13 for the task of identifying turbulent events.

From industry knowledge, data that are older than 4 hours have "expired" and are not relevant. To err on the side of caution, we break the dataset into 6-hour chunks to help ensure each chunk can "see beyond the expiry time." These chunks are from many different locations and span more time than a typical turbulence event.

Our next stage is to cluster each chunk spatially based on the Cartesian (x, y, z) coordinates of each reading. We use kmeans and choose the cluster number so as to optimize the Silhouette score [4].

Finally, these spatial clusters may include multiple turbulence events, so we cluster each of these clusters in time, again with kmeans and Silhouette scoring for choosing the number of clusters.

We define these resulting temporal clusters within spatial clusters within chunks as **turbulence events** and compute summary statistics for these events.

In the dataset studied, the chunking resulted in 19 different segments ranging from 5 to 1588 records (with a median of 326 records). From the 19 segments, the spatial clustering resulted in 52 clusters from 1 to 1066 records (with a median of 32 records). From those 52 spatial segments, the temporal clustering resulted in 180 temporal clusters from 1 to 621 records (with a median of 12 records). In Figure 5.2, an example of the spatial and temporal clusters is given. Hence we work with 180 spatiotemporal clusters, which we refer to as turbulence events.

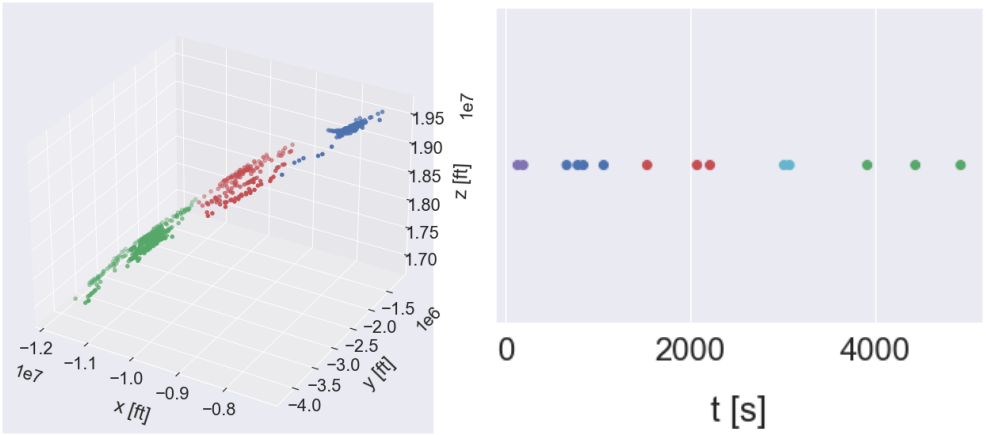


Figure 5.2: Examples of the clusters formed. Left: spatial clusters for a randomly chosen time chunk. Right: temporal clusters for a randomly chosen spatial cluster with times relative to the start of the dataset. Within each plot, different colors indicate data belong to different clusters.

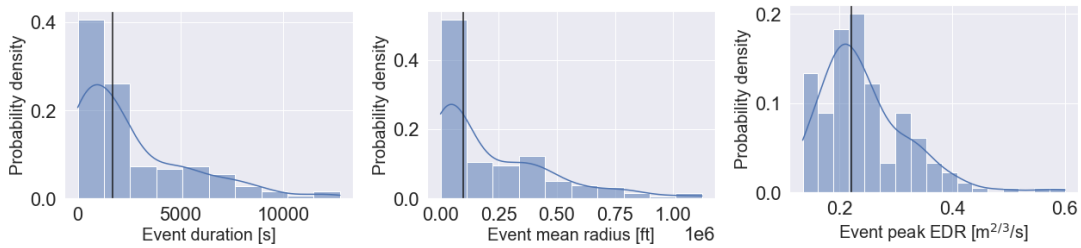


Figure 5.3: Distributions of duration, mean radius, and intensity. Vertical bars mark the medians, which are respectively: 1700 **s,** $100,000$ **ft, and** 0.22 **m**$^{2/3}$**/s.**

Characterizing turbulence events

From the clustering, we can concern ourselves with the characteristics of the size, duration, and intensity of the turbulence events identified. We define the size of an event as the mean distance to the cluster centroid, the duration as the difference between the latest and earliest event times of the cluster, and the intensity as the peak EDR measured across all points in the cluster.

The distributions of the sizes, durations, and intensities are given in Figure 5.3. The median values are 100,000 ft (approximately 30 km), 1700 s (approximately 28 minutes), and 0.22 m$^{2/3}$/s for the sizes, durations, and intensities, respectively.

For an exploratory analysis, we consider the relationship between different properties of the clusters. We continue to define the size, duration, and intensity of the clusters as above, but we add altitude (the mean altitude of points in the cluster), mean difference quotient of temperature magnitude (Equation (5.4)), and mean difference quotient magnitude (Equation (5.5)). While not exact, we treat these quantities as heuristic estimates for a directional derivative of temperature and a "directional" shear rate, respectively.

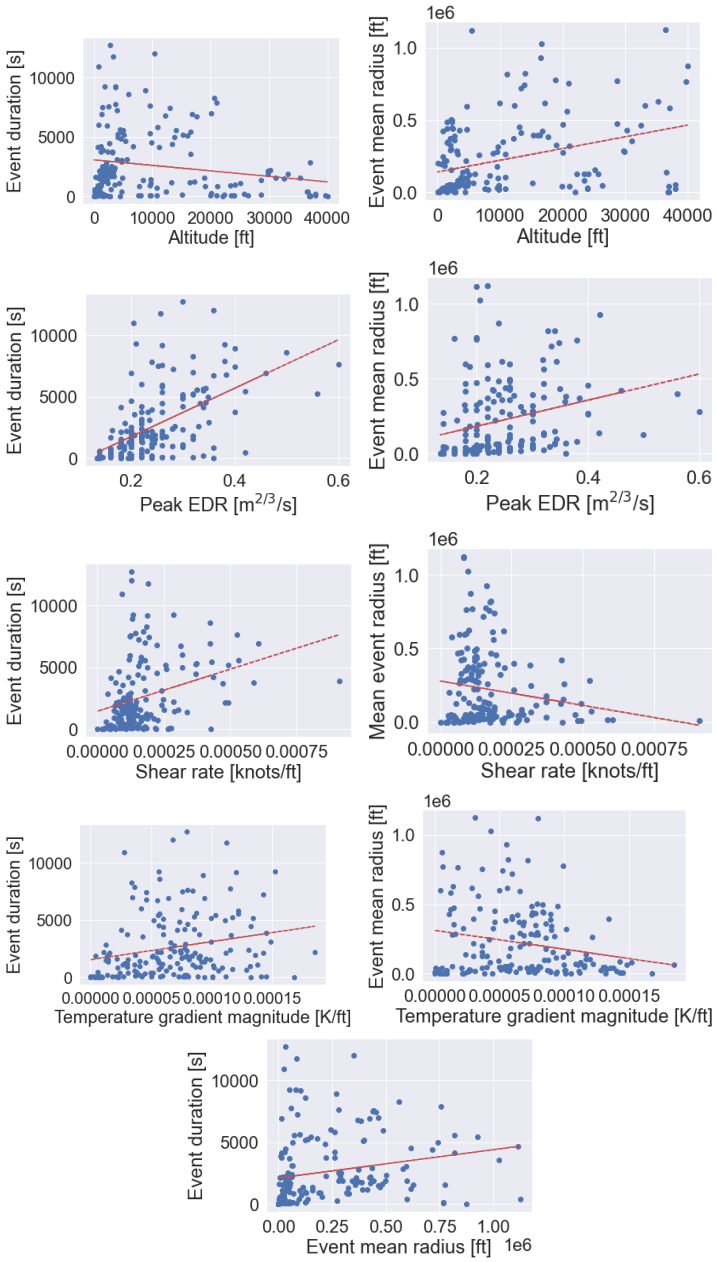


Figure 5.4: Scatterplots of trends between turbulence event statistics. The respective pairs of R and p-values are: $(-0.18, 0.013)$, $(0.36, 8.7 \times 10^{-7})$, $(0.56, 5.2 \times 10^{-16})$, $(0.27, 1.9 \times 10^{-4})$, $(0.31, 2.4 \times 10^{-5})$ $(-0.17, 0.025)$, $(0.22, 0.026)$, $(-0.21, 0.0042)$, and $(0.21, 0.0054)$ read lexicographically.

Identifying trends

In Figure 5.4, we illustrate the relationships with scatterplots and the lines of best fit. The data are quite noisy and the R^2-values are 0.3 or less, but the trendlines point to possible patterns worthy of further exploration.

Summary

By using the clustering analsyis, we were able to identify groups of records considered as isolated turbulence events, which we were then able to study. The summary statistics of these events have yielded a possible ballpark figure for the duration of turbulence: **the median duration of turbulence is approximately** 28 **minutes and it tends to be localized to a region with a characteristic radius of approximately** 30 **km.** We also identified possible patterns in turbulence that are found on very large scales, such as more intense turbulence events lasting longer.

These results are promising, but there are limitations to this analysis. There is a lot of noise in the data and linear relationships do not capture a lot of the variance. Some of the estimated durations may have been artificially deflated due to the 6-hour time chunking and a more careful filtering process could remove data that are too close to the end of the time chunk.

It would be of interest to explore these patterns on more of the IATA datasets and to try to find empirical formulae for the EDR rating, turbulence size, and duration from properties of turbulence events.

5.3.2 Classifier

Data preparation and use

The dataset used in this study is aggregated from three separate files, covering comprehensive flight data across the US during the last week of April 2024:

- *East to West Coast US Last week of April 2024*;
- *East Coast US Last week of April 2024*; and
- *West Coast US Last week of April 2024*

The data from these files were loaded and combined to create a comprehensive dataset. The dataset encompasses a variety of flight-related variables as described previously.

The initial phase involved loading the dataset from the CSV files and converting the *observationtime* column to a datetime format. Unnecessary columns such as *id*, *tafi*, *departureAerodrome*, *destinationAerodrome*, *direction*, and *mean* were then removed. The dataset was filtered to include only observations at or above 30,000 feet, as high-altitude turbulence is of primary concern. Thus we only look at turbulence events that happen during the flying phase and not the ones that occur during takeoff or landing. An indicator of turbulence, *turbulence_presence*, was created, with a value of 1 if the peak EDR is greater than 0.13 and 0 otherwise. The data was then sorted by *observationtime* to facilitate time-series analysis. A visualization of turbulence events in time series is displayed in Figure 5.5.

Grid formation. A key step in this analysis is the formulation of a geospatial grid. The data is converted into a GeoDataFrame to enable spatial analysis and the geographical bounds of the data are determined. The area is divided into a grid with cells of 200 km x 200 km. An example of the grid is shown in Figure 5.6. Note that the altitude is ignored as part of the grid formation. This grid was represented by polygons, and each observation was assigned to a corresponding grid cell based on its geographical coordinates. The step of formulating grids leads to calculating the local gradient, described below.

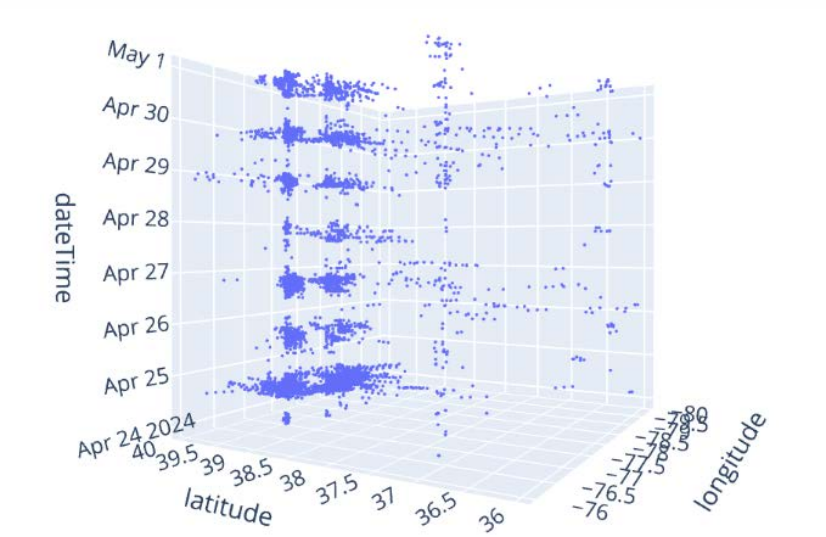


Figure 5.5: Time Series Events of Turbulence.

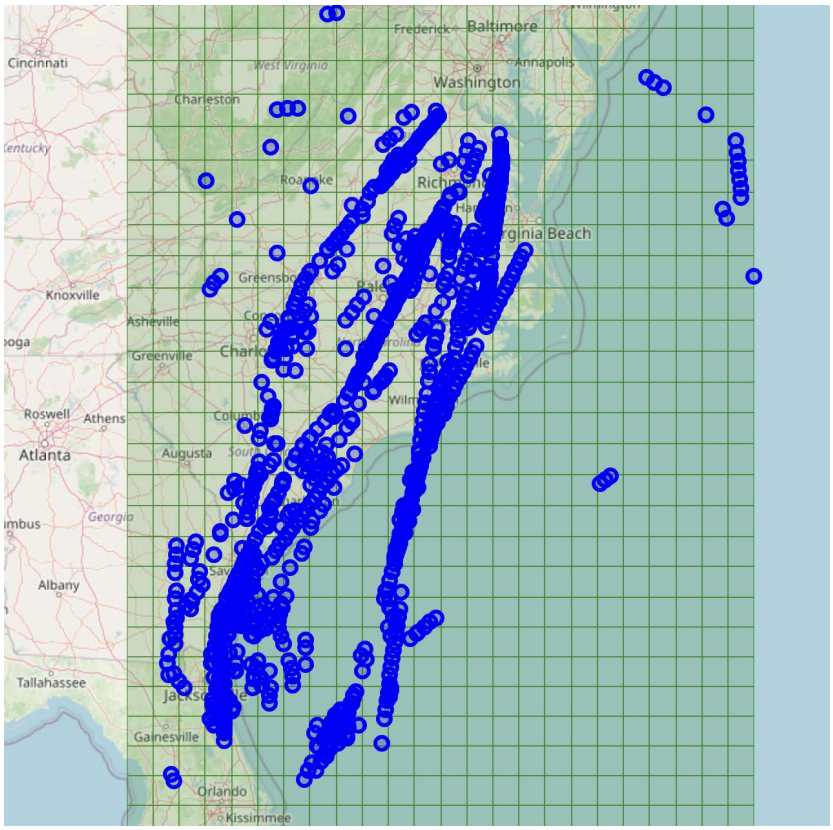


Figure 5.6: Separating the Dataset into Grids by Latitude and Longtitude.

Gradient calculation. Local and global gradients for temperature, wind speed, and peak EDR were calculated for each grid cell. Local gradients capture the rate of change over short time intervals (up to 900 seconds) between consecutive observations. The local gradients for temperature (T), wind

speed (W), and peak EDR (E) were calculated using the following formulas:

$$\text{gradient\_temp}_i = \frac{T_{i+1} - T_i}{\Delta t_i},$$

$$\text{gradient\_wind}_i = \frac{W_{i+1} - W_i}{\Delta t_i},$$

$$\text{gradient\_peak}_i = \frac{E_{i+1} - E_i}{\Delta t_i},$$

where Δt_i is the time difference between consecutive observations at records i and $i+1$.

Global gradients represent the rate of change over longer durations, capturing broader trends. The global gradients for temperature and wind speed were calculated over continuous observation periods using these formulas:

$$\text{global\_gradient\_temp} = \frac{T_{\text{end}} - T_{\text{start}}}{\Delta T},$$

$$\text{global\_gradient\_wind} = \frac{W_{\text{end}} - W_{\text{start}}}{\Delta T},$$

where T_{start} and W_{start} are the temperature and wind speed at the start, and T_{end} and W_{end} are the values at the end of the observation period, with ΔT being the total duration.

Turbulence duration calculation and classification. The duration of turbulence was calculated using the following approach. The data within each grid cell was sorted by *observationtime* and the time difference between consecutive observations (Δt) was computed. A mask was applied to identify time differences less than or equal to 900 seconds. To avoid division by zero, any zero time differences were replaced with 1 second. A division by zero can happen when two planes in the same cell encounter a turbulence at the same time. Adding a 1 second time difference was a way to avoid division by zero but still retain the reports that are very close in time. Local gradients for temperature, wind speed, and peak EDR were then calculated for these time intervals.

An empty series for *turbulence_duration* was initialized. A loop was used to iterate through the observations and calculate the duration of continuous turbulence events. If the time difference between consecutive observations was less than or equal to 900 seconds, the duration was accumulated. When a gap greater than 900 seconds was encountered, the accumulated duration was assigned to the relevant observations, and the gradients over the entire duration were calculated. This process ensured that both the local and global trends were captured accurately.

The descriptive statistics for turbulence duration are given below.

```
count                      1952497
mean     0 days 01:27:48.550094665
std      0 days 01:38:18.051322811
min         0 days 00:00:00.002000
25%                0 days 00:24:17
50%                0 days 00:55:20
75%                0 days 01:55:00
max                0 days 15:25:10
```

Based on the descriptive statistics, we subdivide the turbulence events into 3 classes: those lasting less than 25 minutes, 25 - 60 minutes, and more than 60 minutes. We observe that this roughly amounts to 25% of our data having intervals less than 25 minutes, slightly more than 25% having intervals between 25 and 60 minutes, and slightly less than 50% of the intervals being longer than 60 minutes.

Random forest classifier

The primary model used for predicting turbulence durations was the Random Forest Classifier. The dataset was split into training and testing sets in a 70-30 ratio. The model was trained using the training data and evaluated on the test data. This process was repeated 10 times to ensure consistency and reliability of the model's performance. The average accuracy of the Random Forest Classifier across all iterations was approximately 0.64, indicating a reasonable effectiveness in predicting turbulence durations based on the provided features. It is also worth noting that our classifier performed very well on the class of turbulence events that are longer than 60 minutes (with an accuracy of 0.85), which was expected as this class is far more represented than the other two classes.

Summary

This study presented a data-driven approach to forecast aircraft turbulence using observational data aggregated from all US flights. The methodology involved cleaning and preprocessing the data, formulating a geospatial grid, calculating local and global gradients for key variables, and categorizing turbulence duration into bins. The primary model used for prediction was the Random Forest Classifier, which demonstrated reasonable effectiveness in forecasting turbulence durations based on the selected features.

One possible improvement to this approach is the method of grid formation. Currently the grids are formed arbitrarily, which may not accurately represent the spatial distribution of turbulence events. A more valid approach would be to form the grid by initially clustering the turbulence events, ensuring that the grid cells better capture areas with similar turbulence characteristics. This could potentially enhance the model's performance and provide more accurate predictions.

5.4 Approaches to Problem 2

5.4.1 Neural network

Neural networks encompass a broad spectrum of architectures, each distinguished by unique strengths suitable for specific applications. The Probabilistic Neural Network (PNN), a variant of the feedforward neural network, is particularly noted for its foundational principles in probability theory and its effectiveness in classification tasks [5]. This section delves into the architecture, motivations, and empirical outcomes in the implementation of the PNN methodology.

The core mechanism of PNN involves a kernel-based approximation to compute the posterior probabilities of a Bayesian network for class prediction [6]. Structurally, the PNN is organized into several distinct layers: an input layer, a pattern layer, a summation layer, and an output layer. It is highly regarded for its ability to approximate a Bayesian classifier with sufficient training samples, showcasing its robustness and accuracy in statistical classification.

Inspired by recent advancements reported in [7], our study employed a PNN model to predict energy dissipation rates in geophysical turbulent flows. The cited work demonstrates how PNNs adeptly capture the distribution tails in simulations of decaying turbulence, replicating conditions similar to those found in oceanic environments. The integration of underlying physical principles such as density gradients and velocity structures into the PNN contributed to its enhanced predictive performance, surpassing traditional theoretical models. This successful application underlined the model's capability and prompted its further adaptation to accurately represent complex fluid dynamics in stratified flows.

Architecture

Input Layer: Acts as the gateway for input features, where each neuron corresponds to one input feature.

Pattern Layer: Computes the Euclidean distances between the input vector and the training vectors, applying a radial basis function to evaluate similarity measures.

Summation Layer: Aggregates the outputs from the pattern layer, compiling the contributions by class to generate probabilistic outputs.

Output Layer: Interprets the aggregated probabilities to determine the final classification, effectively deciding the predicted class based on the highest probability.

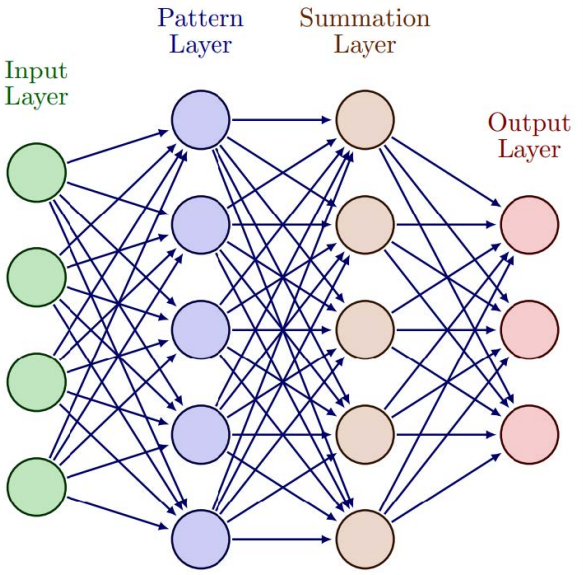


Figure 5.7: Probabilistic Neural Network Structure. The network consists of four layers: Input Layer, Pattern Layer, Summation Layer, and Output Layer.

Outcome

The deployment of the Probabilistic Neural Network (PNN) model in our study has yielded several notable outcomes.

Improved Predictive Accuracy: Our PNN model, inspired by advances in predicting energy dissipation rates in geophysical turbulent flows, has demonstrated enhanced accuracy in forecasting turbulence. This is particularly due to its ability to capture robustly the tails of output distributions, essential for modelling the stochastic nature of atmospheric conditions.

Effective Uncertainty Modelling: By leveraging Gaussian distributions for outputs, the PNN model efficiently incorporates prediction variance, allowing for more reliable and trustworthy predictions in aviation safety applications.

High Fidelity in Complex Fluid Dynamics: The model's capability to integrate physical principles such as density gradients and velocity structures has led to high-fidelity simulations of stratified flows, aligning closely with real-world atmospheric phenomena.

Fast and Efficient Training: The inherent nature of PNN, which does not require iterative training procedures, has significantly reduced training times. This efficiency is crucial for real-time applications where rapid model updates are necessary.

Robustness to Noisy Data: The probabilistic framework of the PNN has shown resilience against noisy input data, enhancing the robustness and reliability of our turbulence predictions.

Overall, the integration of the PNN model into our framework has significantly advanced our capabilities in predicting turbulence, showcasing its potential to improve navigational safety and

efficiency in aerial transportation. Moving forward, we aim to incorporate additional meteorological data and further refine our model to achieve even greater precision in our predictions.

Procedure

The procedure used in our study to develop and deploy the PNN model for turbulence prediction involved several key steps.

Data loading and preprocessing. Data was loaded from a CSV file containing 1 million random samples of West Coast US data over the last two years. The observation times were parsed into datetime objects to facilitate temporal analysis. Covariates used in the model included altitude, wind speed, temperature, humidity, and barometric pressure, chosen for their direct influence on turbulence conditions.

Grid system and feature engineering. A grid system of 50 km x 50 km was established to segment the data, aiding in localizing predictions and identifying spatial patterns in turbulence occurrences. Latitude and longitude were binned to create grid cells and the target variable, turbulence probability, was defined based on EDR values.

Model training and evaluation. The PNN model was built using TensorFlow, with a custom loss function to account for prediction variance. The model architecture included layers to predict both the mean and variance of the turbulence probability. The training-testing split was 70:30. The model predicts the mean and variance of the probability of turbulence for the next 30 minutes and over the entire duration of the flight, aiming to provide pilots with both immediate and long-term insights into turbulence risks.

Visualization

Results were visualized using the Folium library, creating maps that show the locations of turbulence occurrences. Green markers indicate no turbulence, while red markers indicate turbulence. The following two images were generated using Folium.

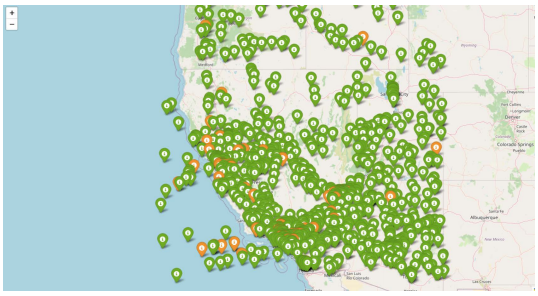


Figure 5.8: Map of the West Coast with indexed EDR data points, illustrating the geographic distribution of turbulence events.

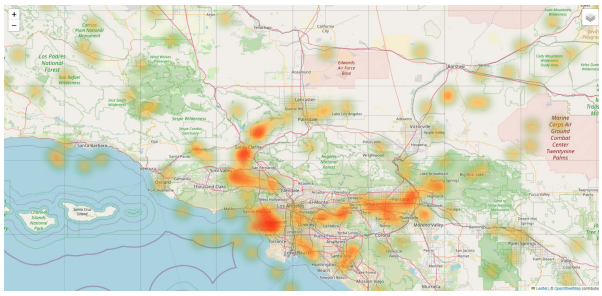


Figure 5.9: Heatmap of EDR distribution over the West Coast, providing a visual representation of turbulence intensity.

Key takeaways from the model include the identification of sudden changes in barometric pressure and high wind speeds as strong predictors of turbulence. This information is highly interpretable, helping pilots make informed decisions to avoid or prepare for turbulent conditions.

5.4.2 Logistic regression

Motivation and data exploration

The second problem is to determine the likelihood of turbulence ahead of an aircraft based on live and historical turbulence data. Specifically, based on the information available to us, such as the wind

speed and the aircraft's position, we want to predict whether turbulence will exceed a certain peak EDR threshold (given by IATA) that signals the occurrence of the turbulence, i.e., to assign a value to a binary "turbulence indicator" Y.

$$Y = \begin{cases} 1 & \text{if Peak EDR} \geq 0.13, \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

Since we formulate the problem as a binary classification problem, where the outcome is categorical with two possible values, logistic regression can be used to predict the probability of turbulence occurrences. As we will see in the next subsection, Y will be used to compute the odds, which is the target variable of our model. A crucial assumption we are making is that the relationship between independent variables and the log-odds of the dependent variable is linear.

The dataset we used is the **1M East-Coast random data for 2023-2024**. Given that there are a great number (1 million) of data points, we chose to work with a subset of 425,000 observations to improve converging efficiency. We used 82% of the data points (350,000 observations) as training data and 18% (75,000) as testing data. We removed rows that contain $XXXX$ as the ICAO Airport code (departure and destination) since they are invalid codes and these rows always have an EDR of 1.

We used the following independent variables: aircraft's location (longitude, latitude, and altitude), wind speed, temperature, wind direction, and seasonality (in which the four seasons are represented by 3 dummy indicator variables, as illustrated later). The Probability Density Function Graphs are displayed in Figure 5.10.

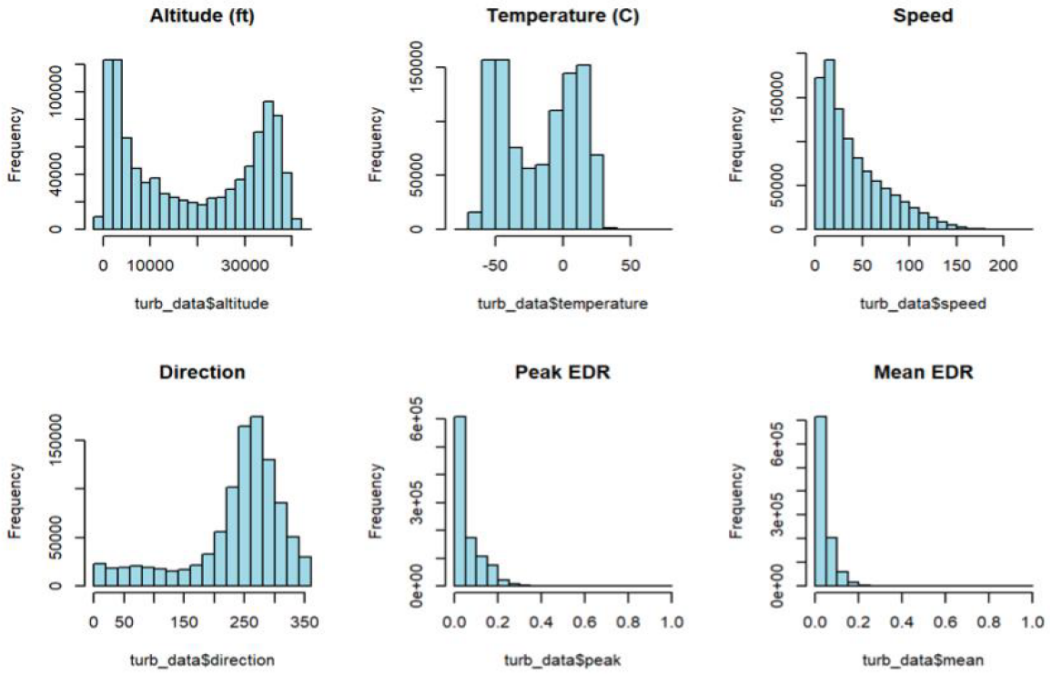


Figure 5.10: PDFs for Independent Variables.

Both altitude and temperature appear to follow bimodal distributions. Their distributions are similar as intuitively altitude and temperature are correlated. Speed is right-skewed since the majority of the dataset pertains to flights where the wind speed is under 100 knots. Peak EDR and mean EDR are also right-skewed with a similar data range. We focus on forecasting peak EDR as it incorporates more of the extreme turbulence conditions. The wind direction is left-skewed since the majority of the wind comes from the West (270 degrees).

We also produced a visualization of turbulence occurrences over our sample data. As shown in Figure 5.11, in our dataset, 15% of data show the presence of turbulence (peak EDR ≥ 0.13). We can see that the turbulence occurred more over land than over water, and it is concentrated along the coastline. The areas that have the strongest turbulence are mainly big cities: New York, Washington D.C., and Miami, as shown in the green rectangles. The turbulence with peak EDR value occurred around Orlando.

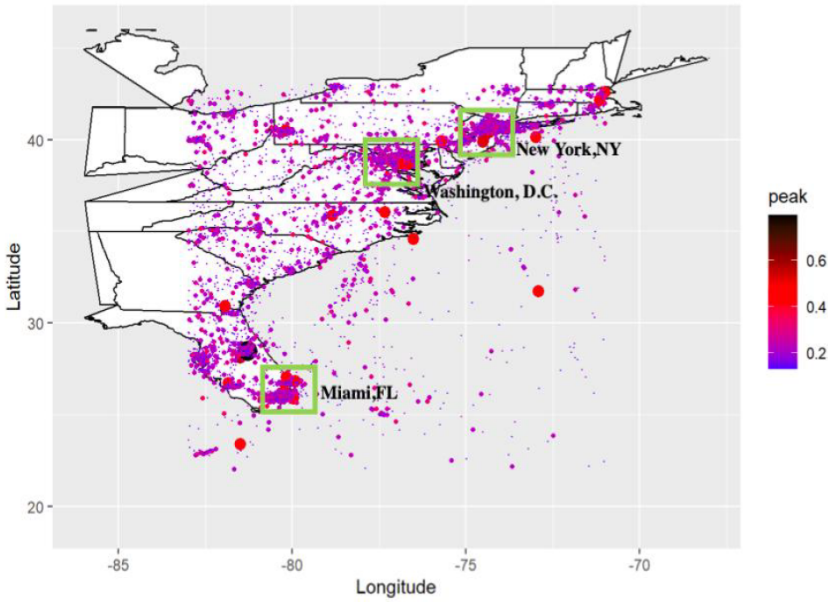


Figure 5.11: Turbulence of East Coast USA (2023-2024).

Model fitting

The model that we fitted is a logistic regression model taking the following form:

$$\begin{aligned}\log\left(\frac{\pi}{1-\pi}\right) =& \beta_0 + \beta_1 L_y + \beta_2 L_x + \beta_3 V + \beta_4 T + \beta_5 D \\ &+ \beta_6 S_{spring} + \beta_7 S_{summer} + \beta_8 S_{winter}.\end{aligned} \tag{5.7}$$

On the left-hand side of the equation, the term inside the bracket is the odds of turbulence occurring, with $\pi = \hat{P}(Y = 1|X)$ holding and X representing the matrix of our covariates (position, temperature, wind direction, etc.). One can easily show that the above equation is equivalent to

$$\hat{P}(Y = 1|X) = \frac{\exp(\beta^T X)}{1 + \exp(\beta^T X)}. \tag{5.8}$$

On the right-hand side, L_y and L_x are the latitude and longitude (in degrees), V is the speed of the wind (in knots), T is the temperature (in °C), D is the wind direction (in degrees), and finally, S_{summer}, S_{spring}, and S_{winter} are indicator variables for the seasons. For example, for summer, $S_{summer} = 1$ while $S_{spring} = S_{winter} = 0$. For autumn, $S_{summer} = S_{spring} = S_{winter} = 0$.

We also omitted the altitude variable since there is a strong collinearity between altitude and temperature (see Figure 5.12): thus including both covariates would jeopardize the model's performance. The model was fitted using the "**glm**" function in R. The output is shown in Figure 5.13.

The first thing we observe is that all the β coefficients have small p-values and therefore are significantly different from zero, which means that all the covariates have significant effects on the odds

of turbulence occurring. One of the biggest advantages of regression models is their interpretive power. Those β coefficients allow us to quantify the effect that each covariate has on the target variable. In Figure 5.13, however, the displayed coefficients are associated with **log**(**odds**) rather than the odds themselves. Hence we need to take the exponentials of those coefficients in order to evaluate their respective impacts on the odds.

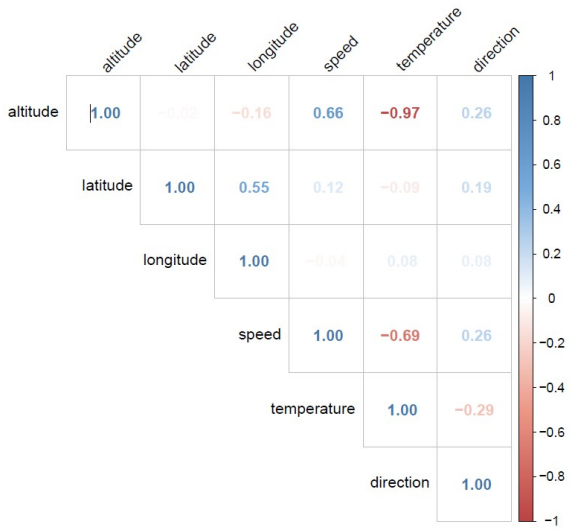


Figure 5.12: Correlation coefficients between the covariates: note the extremely high value between altitude and temperature.

```
Call:
glm(formula = peak_binary ~ latitude + longitude + speed + temperature +
    direction + season, family = "binomial", data = turb_data.train)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.645e+00  1.982e-01  -8.299  < 2e-16 ***
latitude      2.942e-02  1.231e-03  23.897  < 2e-16 ***
longitude     1.600e-02  2.112e-03   7.574 3.62e-14 ***
speed         6.016e-03  2.702e-04  22.259  < 2e-16 ***
temperature   4.190e-02  3.453e-04 121.338  < 2e-16 ***
direction     1.820e-04  6.069e-05   2.998  0.00272 **
seasonSpring  5.705e-01  1.402e-02  40.696  < 2e-16 ***
seasonSummer -1.327e-01  1.532e-02  -8.659  < 2e-16 ***
seasonWinter  3.857e-01  1.512e-02  25.502  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 300434  on 349999  degrees of freedom
Residual deviance: 264827  on 349991  degrees of freedom
AIC: 264845

Number of Fisher Scoring iterations: 5
```

Figure 5.13: Output of the “glm” function of R.

Interpretations

Position of the aircraft: For every increment by one degree in **latitude** (towards the North), the odds of turbulence occurring are multiplied by $\exp(0.029) \approx 1.030$, which represents a 3.0% increase in the odds of turbulence when all the other variables remain constant. If the aircraft travels one degree in **longitude** (towards the East), we will have a multiplication of $\exp(0.016) \approx 1.016$, a 1.6% increase in the odds. Based on this, we can conclude that moving towards the North-East would give us a higher chance of turbulence only on the East Coast of the USA.

Wind speed and direction: An increment of one knot in wind **speed** results in a 0.6% increase in odds of turbulence when all other variables remain unchanged. Although the coefficient is statistically significant, the wind speed doesn't impact that much the odds of turbulence compared to other variables.

Furthermore, we directly included the wind **direction** in the logistic regression model. We observed a 0.01% increase in odds per degree, which tells us that its impact on turbulence is negligible. We later realized, however, that this interpretation is not accurate because of the periodic nature of the wind direction. For example, a one degree increase from 359.5 degrees yields 0.5 degrees (which is 360.5 minus 360.0), not 360.5 degrees. For future models that contain periodic data, Flury and Levri (1999) [8] suggest to convert it into two variables: $C = \cos\left(\frac{2\pi D}{360}\right)$ and $S = \sin\left(\frac{2\pi D}{360}\right)$, where D represents the wind direction in degrees. This ensures that we take into account the periodic nature of the wind direction while having a better way of predicting turbulence.

Temperature and seasonality: If the **temperature** increases by one degree Celsius, the odds of having turbulence increase by 4.3% when all other variables remain constant. We can also see from the values of the coefficients that **season** seems to have the largest impact on the odds of turbulence occurring. Especially in the spring, the odds of turbulence occurring increase by 77% compared to autumn. Indeed, by looking at the distribution of turbulence across seasons displayed in Figure 5.14, we see that in spring and winter (47% increase in odds compared to autumn), we have significantly more turbulence than in autumn and summer (12% decrease in odds compared to autumn).

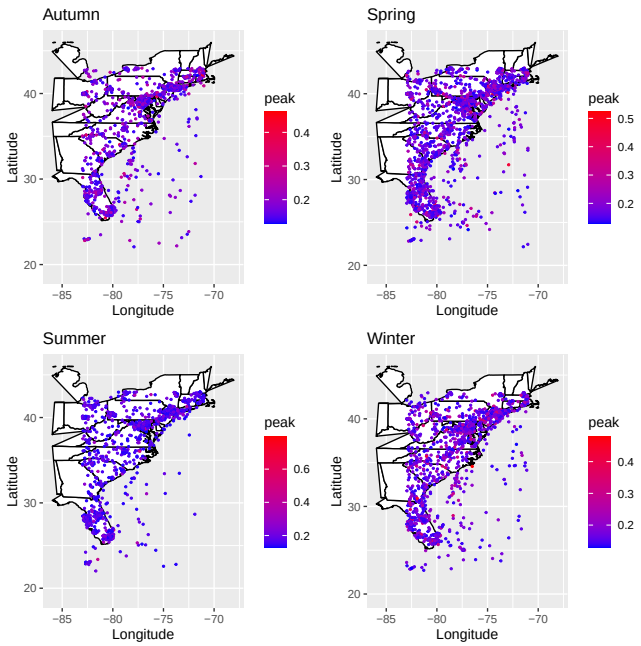


Figure 5.14: Distribution of turbulence (EDR ≥ 0.13) in the East Coast across seasons.

Performance measures

Next, we would like to know how well the model performs on the test data. After predicting using the test data, the ROC curve of the model was plotted. It is shown in Figure 5.15.

The AUC (Area Under the Curve) value was found to be **74.5%**, and the best threshold value was found to be **0.206**. Using this threshold, the performance statistics and the confusion matrix were computed and are displayed in Tables 5.1 and 5.2.

We can see that our logistic regression model has achieved a commendable performance on the test data, even with a limited number of covariates.

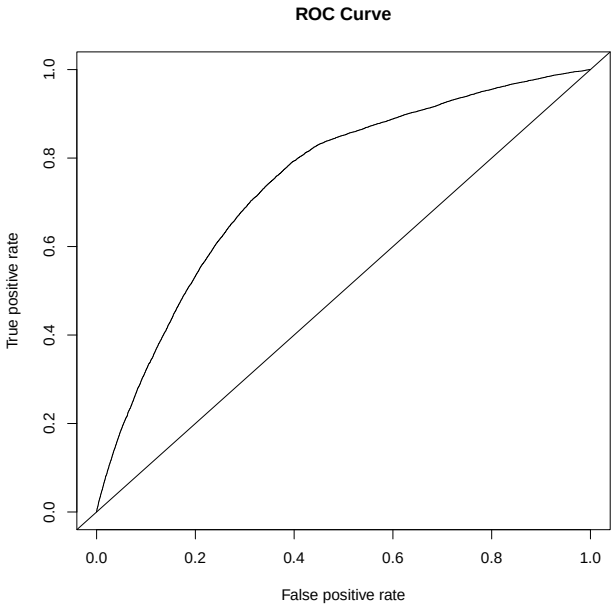


Figure 5.15: The ROC curve of the logistic regression model on the test data.

Table 5.1: Performance statistics of the fitted model on the test set.

Accuracy	69.8%
Sensitivity	68.7%
Specificity	70.0%
AUC	74.5%

Table 5.2: Confusion Matrix.

	Actual	
Predicted	Negative	Positive
Negative	44420	3615
Positive	19034	7931

Recommendations and limitations

Flight scheduling for Spring and Winter: Thanks to the interpretive ability of regression models, we can make a few recommendations about avoiding turbulence based on the values of the regression coefficients. As mentioned above, seasons seem to play an important role in the likelihood of turbulence. Since we expect to have more turbulence in spring and winter, it might be a good idea to use air routes that have fewer turbulence records in spring and winter. It is also a sound idea to arrange fewer flights but use the larger and less turbulence-sensitive wide-body jets in the remaining flights, while in autumn and summer more flights can be arranged with small commercial jets.

Adjusting the aircraft's position to decrease the odds of turbulence: Furthermore we usually tell the pilot to descend the plane when there is a turbulence event, but if the temperature is rising, the model suggests to move upward. This would decrease the temperature and reduce the odds of turbulence. The plane, however, cannot go too high in order not to waste fuel. We can also suggest to the pilot to lean towards South-West until no turbulence is detected.

Although our model was able to achieve a relatively good performance, there are a few shortcomings and limitations in our work. First, when choosing the covariates, we only used the *vanilla* variables which

were neither transformed nor multiplied with other covariates to include the interaction effects. This made our model over-simplified for real-world scenarios and inevitably, hurt the model performance. Second, in Figure 5.12, we can see that speed and temperature also have a relatively strong correlation (around -0.69). Although it's not as strong as the one between temperature and altitude, including both variables in the model introduces some level of collinearity, which can also damage the model's performance. Finally, our analysis only used the East Coast dataset throughout the entire work: hence it is probable that not all the results presented above are valid in other datasets. A future extension of this work could consist of testing the model with other datasets, as well as including the transformations of the covariates along with the interaction effects.

5.4.3 Kriging

The Kriging Model is a powerful tool, specifically designed to make inference in spatial data. It was first introduced by Danie G. Krige in gold mining applications, then used in environmental science, natural resources, etc. The Kriging model is used to predict unknown values at a specific location z in a space D. The predicted values are computed through a linear combination of known values. This method is often confused with a regression model or Inverse Distance Weighted Interpolation, but the difference lies in the way the weights are determined. In a Kriging prediction, the weights are only determined after estimating a variogram model from the data. A variogram is a model representing the spatial covariance structure. On the other hand the Inverse Distance Weighted Interpolation uses the inverse distance raised to a power (usually the power 2) and the regression methods use weights that minimize a loss function on a training set of data.

The Kriging predictor estimates values that are the Best Linear Unbiased Predictor (BLUP), meaning that the estimates are the ones that minimize the prediction error in that specific location [9].

Problem formalization

The turbulence phenomenon is modeled as a random field

$$\{z_s, s \in D\},$$

where

- s is the location in which we observe the value z and is represented by two coordinates, namely longitude and latitude, and $D \in \mathbb{R}^2$ is a two-dimensional space, and
- $z_s \in [0,1], z_s \in \mathbb{R}$ is the EDR value in location s; note that z_s can be either observed (known) if its value is included in the dataset, or unknown, in which case it has to be predicted.

Kriging second order stationarity assumptions. The Kriging model makes the following assumptions.

- **Constant mean**: $\mathbb{E}[z_s] = m_z$
- **Spatial Homogeneity**: $\text{Cov}(z_{s_i}, z_{s_j}) = C(h), \quad h = ||s_j - s_i||, \quad \forall s_i, s_j \in D$, where C is the covariogram.

Model fitting

From the dataset East Coast US of the last week of May 2024, we selected all the EDR reports within the interval 00:00-01:00 UTC on 25th April 2024. Figure 5.16 shows the estimated variogram from a spherical model: we measure the nugget ($\tau^2 = 0.002$), the sill ($\sigma^2 = 0.00525$), and the range ($R = 7.5°$).

Interpretation: The nugget effect is different than zero, this may be due to measurement error. R is the range, i.e., the distance beyond which there is no correlation between two locations. When the sill does not converge to a constant value, the process may not be stationary, in our situation we can assume second order stationarity.

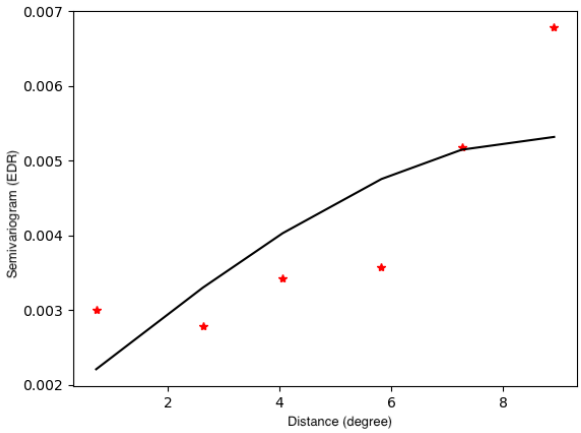


Figure 5.16: Variogram: $\gamma(h) = \frac{1}{2}\mathbb{E}[(Z_{s_i} - Z_{s_j})^2], \quad \forall s_i, s_j \in D, \quad h = ||s_i - s_j||$.

From the variogram we determine the weights λ_i for n neighboring points, and then we make predictions using Equation (5.9) in a grid with latitude and longitude within [35°,35°] and [-78°,-70°], respectively, and a granularity of 0.01°.

$$\hat{z}_s = \sum_{i=1}^{n} \lambda_i z_i \tag{5.9}$$

In figure 5.17, we depict the data used to build the variogram along with the Kriging prediction.

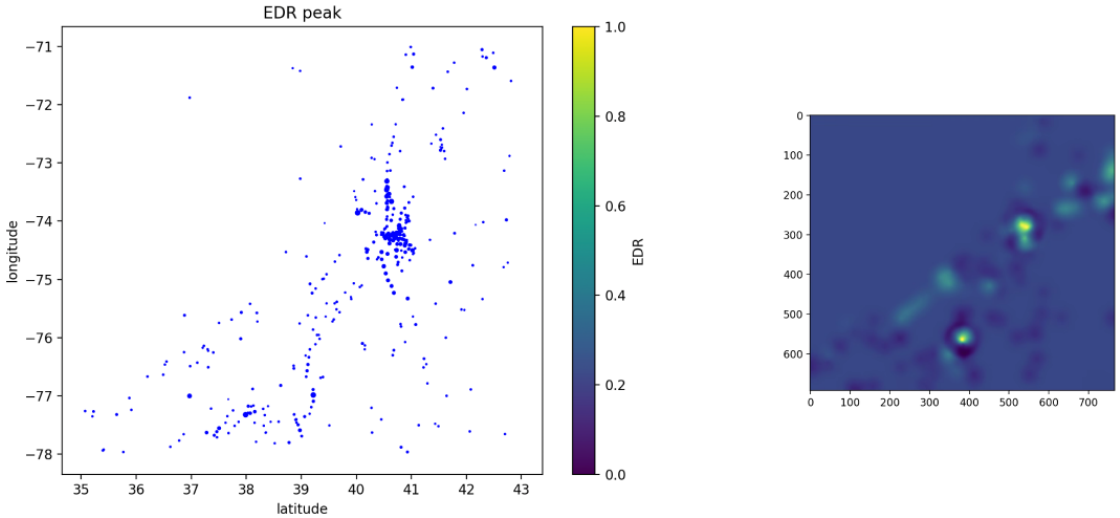


Figure 5.17: Left: the spatial data used to interpolate the values ranging over [35°,45°] in latitude and [-78°,-70°] in longitude with a granularity of 0.01°. Right: Kriging predictions. A bright colour indicates a high likelihood of encountering turbulence, whereas darker regions are associated with a lower EDR prediction.

Model evaluation

Using an independent test set to evaluate a model is not possible due to data scarcity; however, one can still gain some insight through evaluation on the training set. We recall that Kriging does not enable one to learn directly as in regression, but enables one to learn the covariance structure. This allows us, to some extent, to use a performance metric on the training set.

Table 5.3 reports the Mean Square Error MSE=0.0052, the Mean Absolute Error MAE=0.0630, and the Root Mean Square Error RMSE=0.072. Predictions are made over the timeframe 00:00-01:00 UTC of April 25th, 2024. Note that these predictions give us the likelihood of encountering a turbulence event in that interval, but do not provide information on how much it will last.

Table 5.3: Performance metrics.

0.0052	0.0630	0.0720
MSE	**MAE**	**RMSE**

Model limitation

The model does not learn the general pattern of the turbulence phenomena. Instead, it uses real time data to predict the turbulence in a wider region. The limitation in the data comes in the form of unevenly distributed samples in terms of the following parameters.

- Altitude level: most of the time a plane flies above or below 20000 feet (see Figure 5.18).
- Latitude and Longitude: a plane follows roughly the same path, which is the shortest path between airport hubs.
- Time of the day: during the night we have much fewer real-time data, which may make the model fail to capture the turbulence events (see Figure 5.19).

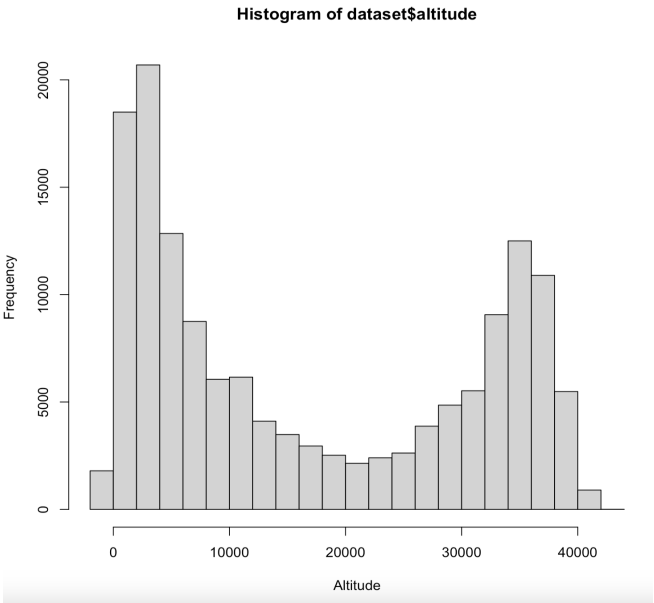


Figure 5.18: EDR report per altitude.

In fact, unevenly distributed data coupled with bias in the dataset (high EDR reports are more likely to be reported than low EDR, due to the reporting system) are a threat to Kriging assumptions. And since we are using a one-hour time frame of turbulence occurrences to make predictions, the Kriging assumptions may not be satisfied at all times.

Future work

To solve distribution issues and the bias in the dataset, one may opt for data imputation. In other words, we insert into the dataset low EDR values along the path of a plane's trajectory that may have not been reported. This approach will provide a dataset that is representative enough of the turbulence

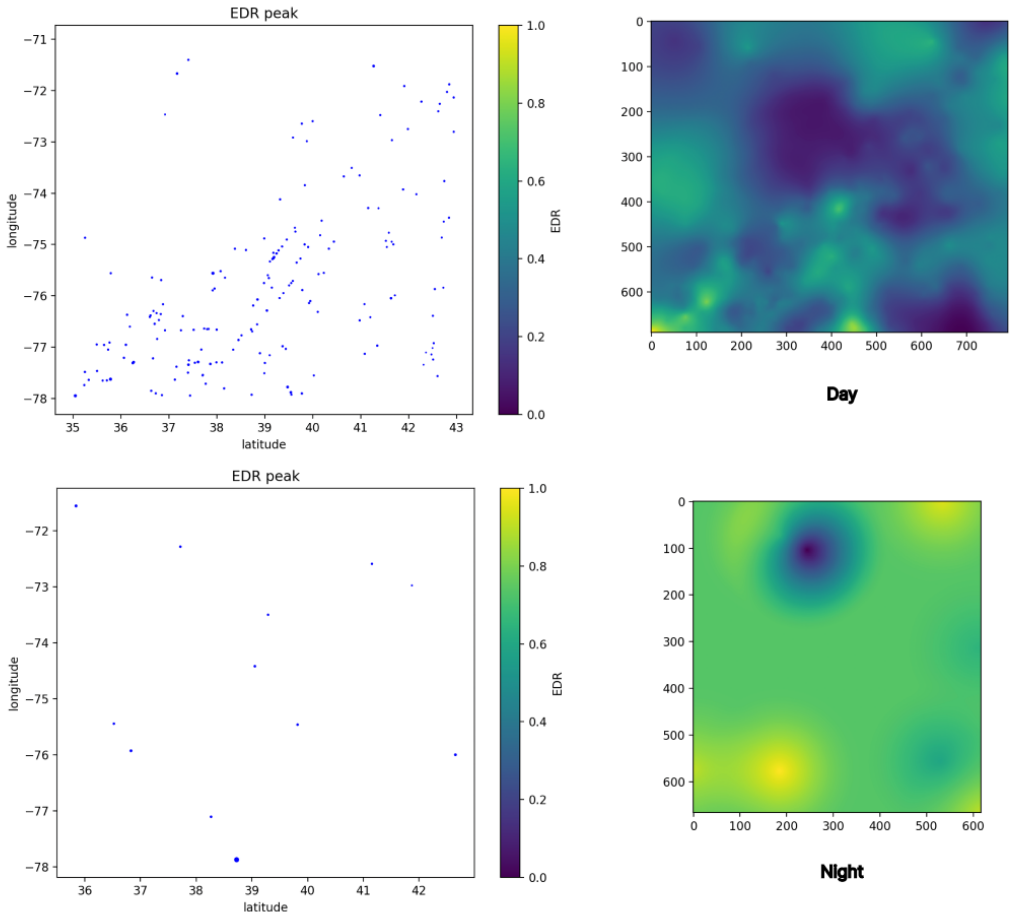


Figure 5.19: Kriging predictions during day and night.

phenomena. Then the homogeneity and anisotropy assumptions (i.e., the spatial autocorrelations do not depend on the direction) can be satisfied.

Experimenting with the Kriging model in higher dimensions (e.g., 3D, 4D), despite the exponential increase in computation with respect to the grid granularity and dimension, may be worth pursuing. Interesting experiments could include other variables, such as temperature and wind speed, into the model.

5.4.4 Flight path trajectory study

Motivation

In the methods discussed so far, the emphasis has been on the measurements of EDR and how these measurements are related or can be inferred based on their spatiotemporal proximity. We can also formulate the problem of EDR inference from the perspective of an individual aircraft. Recall that when the EDR observed is greater than 0, the aircraft will send a report every minute, otherwise there will be a report every 15 minutes (the latter are known as "heartbeat" reports). These reports form a time series, where at each time point we collect information on location, basic weather data, and EDR values as described in Section 5.2. The question that we are interested in exploring here is: given the information of the trajectory so far, can we predict the peak EDR at the next report for each individual aircraft?

Methods

We formulate this as a regression problem and enrich the data at each turbulence report so that we include information on the change in the location (i.e., the change in latitude, longitude, and altitude) since the previous report, as well as adding the previous mean and peak EDR values. The reason for these additional variables is because we want to make a prediction of peak EDR values at each time of a new report, and by including the extra information, we are essentially lifting the problem so that it is approximately Markov.

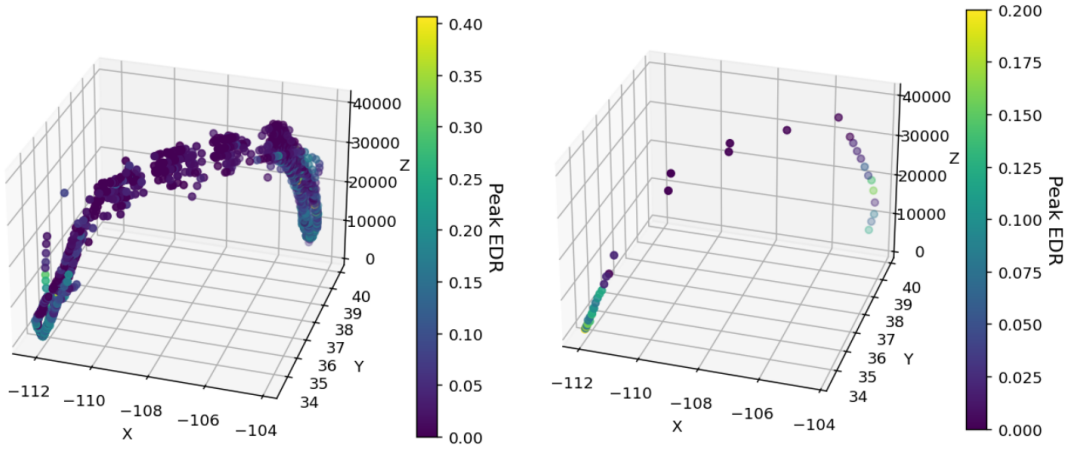
We analyze the East to West coast flight data in the last week of April 2024. To fit and evaluate the parameters of the models, we use 80% of the data as the training set and 20% of the data as the testing set. Our target variable is the (scalar) peak EDR at the next report and the predictors are the data from the reports with the augmented variables discussed above. The data was standardized before being fed to the model.

We utilize and compare two methods: linear regression and Gated Residual Units (GRU). Our baseline, naive model is a simple linear regression model (with intercept). Recurrent neural networks (RNNs) are neural networks aiming to capture the time-dependent behaviour through the updating of a hidden state. RNNs suffer from the vanishing/exploding gradient problem. GRUs were designed to alleviate this problem and have been shown to be effective in applications [10].

We use a GRU to model this problem with a fully connected layer (sigmoid activation) to map to the output. The Adam optimizer [11] is used to minimize the mean square error loss between the predicted peak EDR and the actual peak EDR.

Results

First we make some observations concerning the patterns spotted in the data. Most of the turbulence events occur during take-off and landing, as can be seen in Figure 5.20. In Figure 5.20, we have included all of the Denver to Phoenix flights: note that the high EDR values are found at the ends of the trajectories.



(a) Trajectories over the entire week. **(b) A single trajectory.**

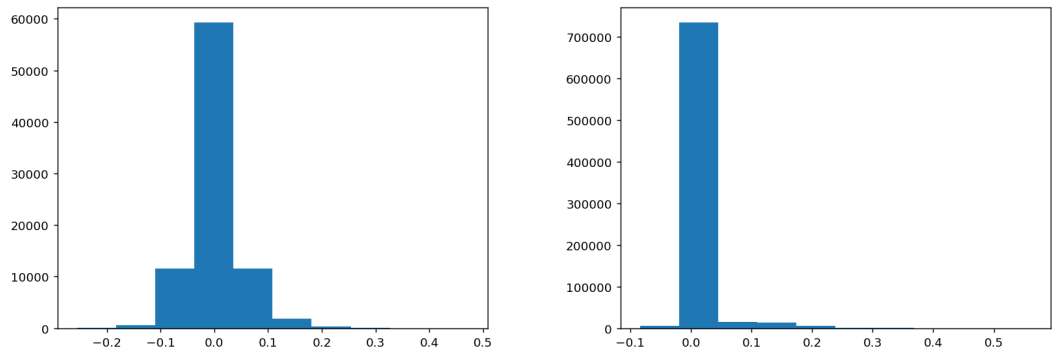
Figure 5.20: 3D plot of the location of the turbulence reports and the peak EDR values of flights from Denver to Phoenix in the last week of April.

The hyperparameters of the final fitted GRU model are given in Table 5.4. Over the timeline of this project, there was not sufficient time to do a thorough hyperparameter optimization and these hyperparameters could be refined in future work. The residuals of the linear regression and GRU models

are displayed in Figure 5.21. Regression has achieved a RMSE of 0.045 whilst the more advanced GRU model achieved a lower RMSE of 0.029.

Table 5.4: Hyperparameters for the final GRU model.

Hyperparameter name	Value
epoch	20
num hidden neurons	16
learning rate	0.001
batch size	64



(a) Regression residuals, RMSE: 0.045. **(b) GRU residuals, RMSE: 0.029.**

Figure 5.21: Residuals for the two methods.

By making inference and prediction at the aircraft level, we can also reconstruct the path of predicted (peak) EDR values for each model, an example of which can be seen in Figure 5.22. This has the potential to be developed into a tool that pilots/air traffic controllers would use to determine whether to change the flight course or not.

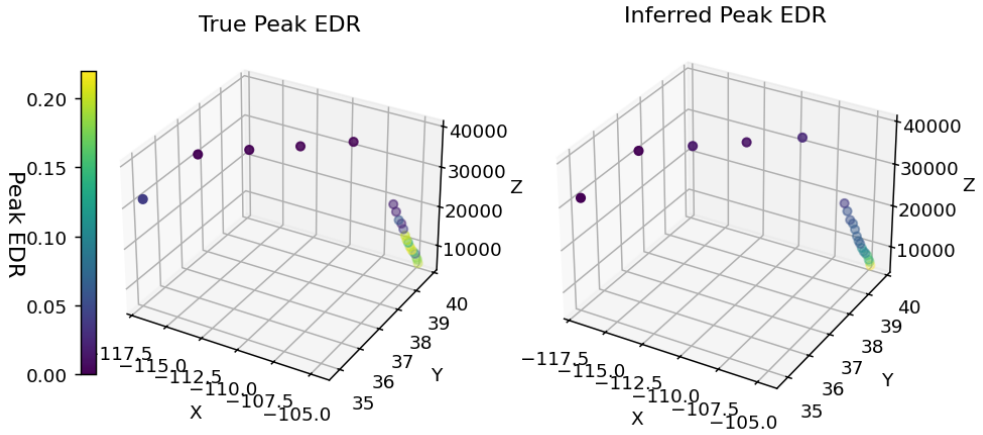


Figure 5.22: True peak EDR vs inferred peak EDR (from the GRU model).

Summary and potential developments

In this section, we have seen that the information from EDR reports forms a time series for each aircraft. With data from current reports and augmented data from previous reports, we can use the

information to infer whether there is still turbulence when an aircraft moves to a new location. Further enhancement could be achieved by using the complete trajectory of the flight and making predictions more than one step ahead. In addition, by quantifying uncertainty, we could make density forecasts rather than point forecasts alone. We could also incorporate the current information of other flights into the region (perhaps leveraging other methods discussed in this report).

5.5 Toy model

One of the problems that we explored was to determine whether there is a model for 'turbulence' that can mimic the observed behaviour. Such a model should have a foundation in the physical process and be applicable on the time scale seen in the data observations of measured EDR values. Some of the basic assumptions include, for simplicity, assuming that the air density $\rho(x,t)$ evolves in one spatial dimension which is associated with the flight path. The model will assume that mass is conserved, and that the velocity dependence of the density is a property that can vary from one air mass to another. In this particular way of viewing the interaction of air masses, the turbulence is associated with discontinuities in density ρ.

With respect to the existence and uniqueness of conservation laws with a discontinuous flux, the situation is far from trivial. For clarity, consider a model of the form

$$\frac{\partial \rho}{\partial t} + \frac{\partial A(x,\rho)}{\partial x} = 0, \qquad x \in \mathbb{R}, t \in \mathbb{R}_+, \tag{5.10}$$

$$\rho(x,0) = \rho_0(x) \in L^\infty(\mathbb{R}). \tag{5.11}$$

If we assume the following properties for $A(x,\rho)$,

1. $A(x,\rho)$ is continuous at all points of $\mathbb{R} \setminus \mathcal{N}$ where is $\mathcal{N}$ is a closed set of measure zero,
2. $\exists$ continuous functions f, g such that $\forall x \in \mathbb{R}$, $f(\rho) \leq |A(x,\rho)| \leq g(\rho)$,
3. $\forall x \in \mathbb{R} \setminus \mathcal{N}$, $A(x,\cdot)$ is locally Lipschitz and one-to-one from $\mathbb{R} \to \mathbb{R}$,

then a unique solution exists. For the purposes of this report, we have set these details aside, and refer the reader to [12] and the references therein.

As a base case consider a velocity dependence of $v(\rho) = 1 - \rho$, where $0 \leq \rho \leq 1$, and associated flux $j = \rho v(\rho) = \rho - \rho^2$. With this prescription, the density satisfies the equation

$$\frac{\partial \rho}{\partial t} + (1-2\rho)\frac{\partial \rho}{\partial x} = 0, \qquad \rho(x,0) = \begin{cases} 1, & 1 \leq x \leq 2, \\ 0, & \text{otherwise}, \end{cases} \tag{5.12}$$

where the density ρ verifies $0 \leq \rho \leq 1$. The solution breaks into two regimes. For $0 \leq t < 1$ the discontinuity originally at $x = 2$ becomes a vertex of a rarefaction fan that spreads out and the discontinuity at $x = 1$ remains in this location: that is, for $0 \leq t < 1$, we have

$$\rho(x,t) = \begin{cases} 0, & x < 1, x > 2 + t, \\ 1, & 1 \leq x < 2 - t, \\ \frac{1}{2}\left(1 - \frac{x-2}{t}\right), & 2 - t \leq x \leq 2 + t. \end{cases} \tag{5.13}$$

Beyond this time, $t \geq 1$, the shock at $x = 1$ interacts with the rarefaction fan causing it to bend to the right, giving a density of

$$\rho(x,t) = \begin{cases} 0, & x < \sigma(t), x > 2 + t, \\ \frac{1}{2}\left(1 - \frac{x-2}{t}\right), & \sigma(t) \leq x \leq 2 + t, \end{cases} \tag{5.14}$$

with $\sigma(t) = 2 + t - 2t^{1/2}$. The discontinuity weakens as the rarefaction fan continues to spread with $\rho(\sigma(t),t) = t^{-1/2}$ for $t \geq 1$.

The initial density can be thought of as a uniform distribution with initial support on the interval $1 \leq x \leq 2$. This means that the effective location and spread of the distribution can be characterized with mean

$$\mathbb{E}(x(t)) = \int_{\mathbb{R}} x\rho(x)\,\mathrm{d}x = \begin{cases} \dfrac{3}{2} + \dfrac{t^2}{6}, & 0 \leq t \leq 1 \\ 2 + t - \dfrac{4t^{1/2}}{3}, & t > 1, \end{cases} \tag{5.15}$$

and spread given by the variance

$$\mathrm{Var}(x(t)) = \mathbb{E}(x^2(t)) - (\mathbb{E}(x(t)))^2 = \begin{cases} \dfrac{1}{12} + \dfrac{t^2}{6} - \dfrac{t^4}{36}, & 0 \leq t \leq 1, \\ \dfrac{2t}{9}, & t > 1. \end{cases} \tag{5.16}$$

Figure 5.23 displays the solution with the corresponding effective location of the 'turbulence' as given by this expected position and variance.

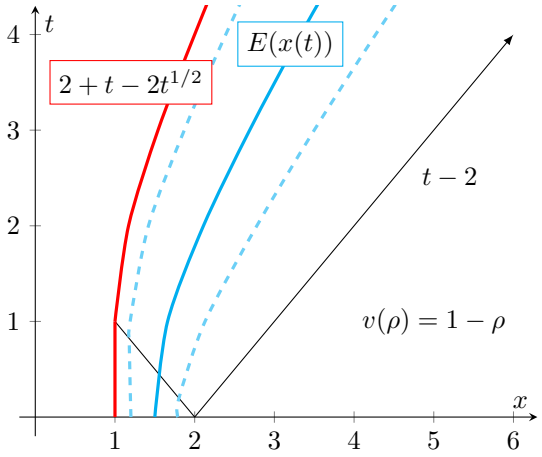


Figure 5.23: The evolution of the base situation with $v(\rho) = 1 - \rho$ and an initial uniform density on $1 \leq x \leq 2$. The red curve shows the location of the shock and the solid cyan curve shows $E(x(t))$ whereas the dashed cyan curves are separated by a distance of $\pm V(x(t))^{1/2}$.

Now that the base case is described, consider an air mass with a different velocity dependence of $v_2(\rho) = 1 - \rho^2$ that approaches with unit speed. Figure 5.24 displays the resulting situation. In this scenario, in front of the weather system, $x > t$, the density $\rho(x,t)$ is unchanged with

$$\frac{\partial \rho}{\partial t} + (1 - 2\rho)\frac{\partial \rho}{\partial x} = 0, \qquad \rho(x,0) = \begin{cases} 1, & 0 \leq x \leq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{5.17}$$

For $t > 1$ the air mass overtakes the rarefaction fan. The density in this region satisfies the PDE

$$\frac{\partial \rho}{\partial t} + (1 - 3\rho^2)\frac{\partial \rho}{\partial x} = 0, \qquad \rho(x = t, t) = \begin{cases} 0, & t < 1, \\ \dfrac{1}{t}, & t \geq 1. \end{cases} \tag{5.18}$$

Solving for the density using the method of characteristics we obtain the formula

$$\rho(x,t) = \frac{1}{2t}\left(1 + \left(1 + \frac{4}{3}(x - t)\right)^{1/2}\right). \tag{5.19}$$

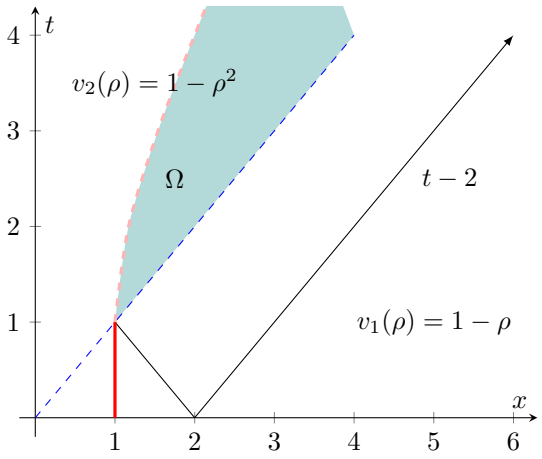


Figure 5.24: With an approaching front for $x \leq t$ the density is $v_2(\rho) = 1 - \rho^2$. The region Ω density needs to be consistent with v_2. In addition the initial condition must be consistent with the rarefaction fan generated within the region with v_1.

With the density evolved beyond $t = x$, the location of the shock initially located at $x = 1$, $t = 1$ can be computed. This 'bounding' shock, $x = \sigma_1(t)$, is determined by the Rankine-Hugoniot condition

$$\frac{\mathrm{d}\sigma_1}{\mathrm{d}t} = 1 - \left(\frac{1}{2t}\left(1 + \left(1 + \frac{4}{3}(\sigma_1 - t)\right)^{1/2}\right)\right)^2, \qquad \sigma_1(1) = 1. \tag{5.20}$$

Note that asymptotically, as $t \to \infty$, $\sigma_1(t) = t + o(t)$ holds, so that $\sigma_1(t)$ becomes parallel to the line $x = t - 2$. In essence, for $x < t$, there is a region, $\Omega = \{x \,|\, \sigma_1(t) < x < t\}$, with a different PDE to reflect the different air mass (see Figure 5.24).

In this sense, the trajectory of the turbulence gives an indication of the density dependence of the velocity of the incident air mass. This toy model shows that by viewing 'turbulence' as an interface between air masses, theories based on the conservation of mass can predict the motion of these interfaces. This provides a way to connect the climatology along a flight path with the observations of regions of persistent EDR values from clear air turbulence.

5.6 Conclusions and future work

Our methods of analysis have given useful insights into the resolution of Problems 1 and 2 posted by IATA at the IPSW. Due to limitations on time, we did not have the chance to explore the incorporation of meteorological data, such as cloud coverage, to build models for thermal-based events, but that investigation would be an interesting next step.

Bibliography

[1] M. C. Prosser, P. D. Williams, G. J. Marlton, R. G. Harrison, Evidence for large increases in clear-air turbulence over the past four decades, Geophysical Research Letters 50 (11) (2023) e2023GL103814. doi:https://doi.org/10.1029/2023GL103814.
URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2023GL103814

[2] P. Williams, Increased light, moderate, and severe clear-air turbulence in response to climate change, Advances in Atmostpheric Sciences 34 (2017) 576–58.

[3] 14th annual industrial problem solving workshop.
URL https://www.crmath.ca/en/activities/#/type/activity/id/3955

[4] K. R. Shahapure, C. Nicholas, Cluster quality analysis using silhouette score, in: 2020 IEEE 7th international conference on data science and advanced analytics (DSAA), IEEE, 2020, pp. 747–748.

[5] D. F. Specht, Probabilistic neural networks, Neural Networks 3 (1) (1990) 109–118. doi:10.1016/0893-6080(90)90049-Q.

[6] S. Haykin, Neural Networks and Learning Machines, 3rd Edition, Pearson, 2009.

[7] J. Lewin, Probabilistic neural networks for predicting energy dissipation rates in geophysical turbulent flows, Journal of Computational Physics 45 (4) (2021) 521–537. doi:10.1016/j.jcp.2021.04.023.

[8] B. D. Flury, E. P. Levri, Periodic logistic regression, Ecology 80 (7) (1999) 2254–2260.
URL https://doi.org/10.2307/176907

[9] A. Menafoglio, G. Petris, Kriging for Hilbert-space valued random fields: The operatorial point of view, Journal of Multivariate Analysis 146 (2016) 84–94, special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces. doi:https://doi.org/10.1016/j.jmva.2015.06.012.
URL https://www.sciencedirect.com/science/article/pii/S0047259X15001578

[10] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).

[11] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv: 1412.6980 (2017).

[12] E. Audusse, B. Perthame, Uniqueness for scalar conservation laws with discontinuous flux via adapted entropies, Proceedings of the Royal Society of Edinburgh Section A: Mathematics 135 (2) (2005) 253–265.

6 Revenu Québec: Detecting fraudulent patterns in a real estate transactional database

Gilles Caporossi [a,b]
Karine Dufresne [c]
Mathieu Gervais-Dubé [b,d]
Nicolas Goulet [e]
Kiyan Karimi Nemch [f]
Hugues-Etienne Moisan-Plante [c]
Abdelmouksit Sagueni [g]

[a] *HEC Montréal*
[b] *GERAD*
[c] *Revenu Québec*
[d] *Polytechnique Montréal*
[e] *UQAM*
[f] *Concordia University*
[f] *Université Claude Bernard Lyon 1*

November 2024
Les Cahiers du GERAD

6.1 Introduction

Tax evasion is a significant issue for governments across the world. As data becomes more accessible in our information age, governments are increasingly turning to advanced data analytics tools to assess individuals and companies. These numerical tools allow for faster analysis of large data sets. Revenu Québec, the entity responsible for managing the Government of Québec's tax collection, has begun using databases to detect tax evasion patterns.

Different methods have been proposed in the literature to solve tax risk and tax evasion related problems. Approaches proposed can be found in a review by Zheng and al. [1]. Among them are association rules mining methods, random forests and support vector machine classifiers applied on companies, transactions or reports, neural networks to predict companies behaviour, and agent-based models to simulate and understand these behaviours. Graph learning methods are also employed and used on the graph representation of datasets. This representation puts forward structural and relational information. Graph neural networks, topological feature extraction, and graph pattern matching have been used on them.

For the Fourteenth Montreal ISPW, Revenu Québec has made an anonymized real estate transactional database available for participants to find potential novel patterns of tax evasion. Normally, detection of tax evasion patterns relies on the experience of tax auditors combined with transactions sampling. As auditors permit a deeper analysis of transactions cases, choosing promising samples of transactions or entities for the auditors to audit is tedious and time consuming and if it is not supported by data, it is left to luck.

In an effort to automatize and formalize this sampling problem, Revenu Québec's representatives wish to develop a support tool to detect potential novel tax fraud patterns. A key point of this problem is its unsupervised nature: the goal was not to find labeled patterns in a graph but to assign the label *suspicious* or *fraudulent* to certain patterns. Therefore we could not rely on labeled data.

To tackle this problem, we propose graph-modelling approaches of the database and feature engineering methods. Notably, we develop a support tool to detect suspicious communities of actors through maximal biclique enumeration. Although the tool is not complete, we hope it will be a starting point for future explorations.

6.2 The data set

In this section the data set provided by Revenu Québec is described. It begins with a brief overview of the data before providing a graph theoretical approach to modelling the data set.

6.2.1 Transactional real estate database

The data set consisted of all (anonymized) real estate transactions that occurred in Québec from 2011 to 2022. Each transaction was associated with a single property and a set of actors (individuals or legal entities involved in a transaction), and various features about each of these three element type were provided. Some examples of these features are the estimated price of a property, its postal code, the type of actors or property involved in a transaction, etc... In total, 3.6 million transactions occurred in Québec for that time period, involving around 2.1 million properties and 11.2 million actors. Furthermore details regarding the relationships between actors were provided.

6.2.2 Graph representation

The entire database can be modelled as a network or an undirected *partial* tripartite graph $N = (A, T, P, E)$, where A is the set of actors, T the set of transactions, P the set of real estate properties, and E the set of edges. The network is *partially* tripartite since there are edges between actors ($e_{a_1 a_2} \in E$) but not between properties or between transactions ($e_{t_1 t_2} \notin E$ and $e_{p_1 p_2} \notin E$). Actors and properties may belong to edges including the transactions they are involved with. Actors belong to edges including other actors if they are involved in a transaction together or if they have relationships apart from transactions. These relationships might be personal or business-related. Figure 6.1 shows an example of such a graph.

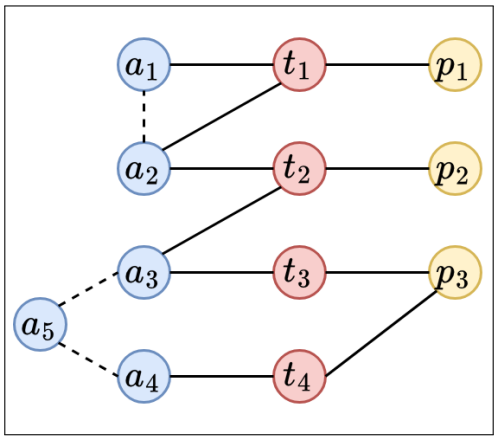


Figure 6.1: A partial tripartite graph. Dashed lines represent edges that can be omitted to model the database without actor-actor edges, resulting in a tripartite graph.

The network, however, can be altered by not including the edges between actors. We consider two different networks: one with the a-a edges and one without them. We observed that 7 671 848 actors were not directly related to a transaction and that the a-a edges made up the bulk of the edges in the network. Table 6.1 displays the cardinalities of these various sets according to the modelling of the network. Processing these networks necessitated significant computing resources, with the first version requiring around 40Gb of RAM simply to be loaded in memory - without labels or metadata about edges and nodes.

Table 6.1: Cardinalities of networks according to the presence of a-a edges.

	Network with a-a edges	Network without a-a edges
Nb. of nodes	17 096 773	9 424 925
Nb. of edges	63 332 237	15 099 371
Nb. of actors	11 253 698	3 581 850
Nb. of a-a edges	48 226 866	NA
Nb. of properties	2 179 427	
Nb of transactions	3 663 648	
Nb. of a-t edges	11 435 723	
Nb. of p-t edges	3 663 648	

We then looked at the connected components (CCs) in the hope of reducing the complexity of the problem. For the first network (with a-a edges), 99% of its nodes and edges are found in the largest connected component; for the second one, 93% of its nodes and 99% of its edges are found in the largest connected component. Other CCs in the networks are not comparable in terms of size and are not reported on; however, adjacency lists for each CC are still made available to the Revenu Québec representatives. Table 6.2 displays the contents of the largest connected component for each network.

Table 6.2: Cardinalities of the biggest CC for each network.

	CC with a-a edges	CC without a-a edges
Nb. of nodes	17 007 724	8 773 121
Nb. of edges	63 327 521	14 498 213
Nb. of actors	11 203 385	3 259 578
Nb. transactions	3 642 175	3 486 526
Nb. of properties	2 162 164	2 027 017
Nb. of a-t edges	11 389 110	11 011 687
Nb. of p-t edges	3 642 175	3 486 526
Nb. of a-a edges	48 206 236	NA

6.3 The proposed tool

In this section, the tool developed to detect suspicious communities is described. First anomalies are detected in the transaction database through feature engineering and statistical analysis. In the second step these abnormal transactions are used as a way to detect suspicious communities using a distance criterion.

6.3.1 Statistical detection of abnormal transactions using feature engineering

This subsection describes the use of statistics to define and detect abnormal transactions.
Large data sets often contain mistakes and missing parts. The first step of the work consisted of cleaning the data: for instance, by removing the transactions for which the value or the associated property is missing. The second step consisted of feature selection, where we determined the important features we would work with and gathered them in a single table. Amongst these features was the region of a property. By studying the regions independently, we established that the data of a certain region of the city of Montréal was so poor that it was better to analyze it independently.

For the rest of this subsection, we consider the data as a table containing the following features: category, year, region, value (of the transaction), estimated value (of the property).

A first step was to measure the discrepancy (or error) between the value of the property and its estimated value and to define a suspicious transaction as a transaction with very high/low error.

We quickly found out about the inefficiency of our first approach and suggested a new error definition. Instead, we computed the ratio between the value of the property and its estimated value, *we consider those transactions with high/low ratio to be suspicious.* The ratio is defined in the following Equation 6.1.

$$\text{Value Ratio} = \frac{\text{Transaction value}}{\text{Municipal Evaluation}} \tag{6.1}$$

The histograms in Figure 6.2 show the robustness of median ratio per year and region.

Given the previous figure, one might think of considering the $0.2\% \sim 0.5\%$ quantile to be suspicious. By investigating the median ratio by category (Figure 6.3), however, we found out that this criterion makes sense except for one category, which is "parking lots." The reason this category has a very large median ratio is because its value is often underestimated. We should then compute the $0.2\% \sim 0.5\%$ quantile of the "parking lots" category separately.

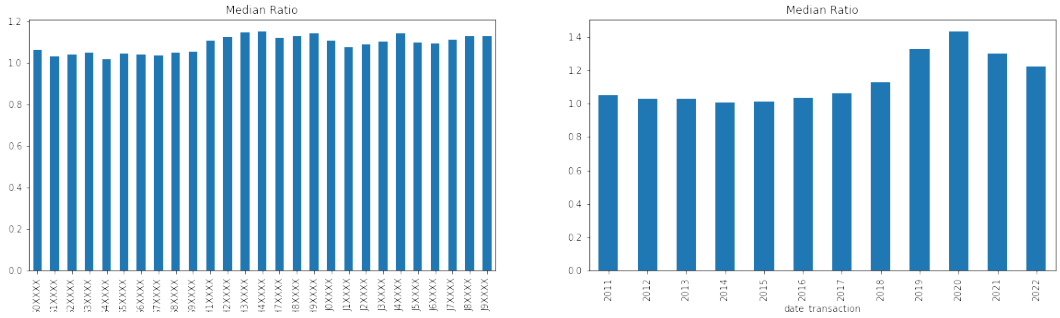


Figure 6.2: Median ratio by region and by year.

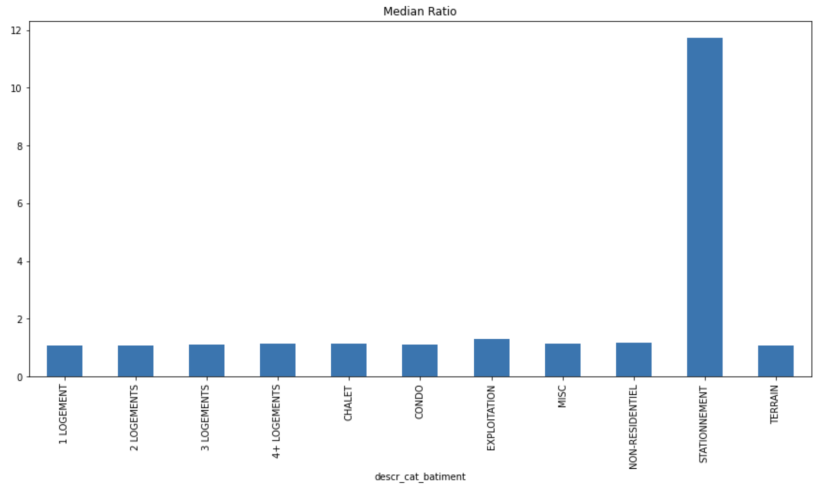


Figure 6.3: Median ratio by category.

Our conclusion was that a suspicious transaction is a transaction of ratio within the $0.2\% \sim 0.5\%$ quantile for a restricted category. Moreover we conclude that one can refine our analysis by looking at specific years and regions in order to gain more accuracy.

6.3.2 Detecting suspicious communities by maximal biclique enumeration

In this subsection the approach to detect suspicious communities of actors in the transactional database is described.

Actor - Actor graph

In the first step, a multigraph $G = (V, E)$ is created where the set of vertices V consists of all the actors. For each pair of actors in V, an edge is present in G if both are involved in a given transaction. Therefore, for each transaction in the transactional database involving a subset of actors $P \subseteq V$, a clique is formed in G. The following Figure 6.4 gives an example of such a graph.

Actors - Suspicious transactions bipartite graph

In the second step a bipartite graph $B = (V', T, L)$ is created with $V' \subseteq V$ a subset of the actors involved in abnormal transactions, T the set of abnormal transactions, and L the set of edges (links) between V' and T.

The following criterion is used to link participants to abnormal transactions.

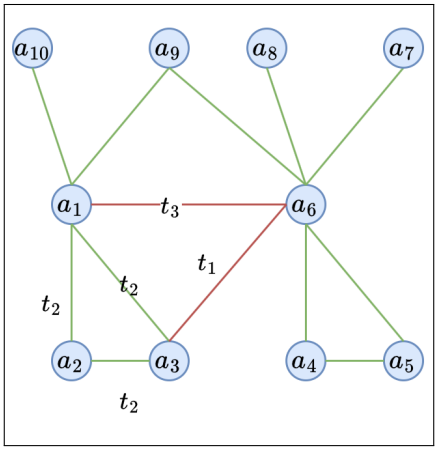
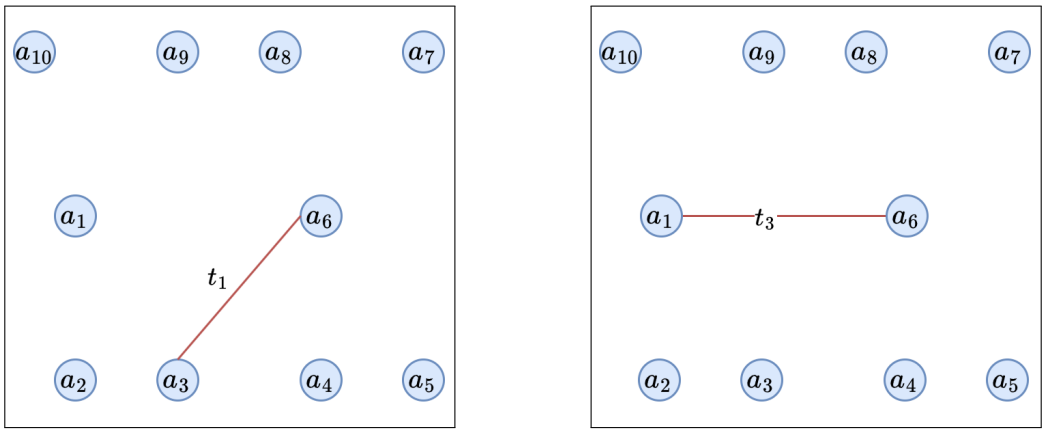


Figure 6.4: A graph linking actors by their common transactions.

Definition 6.1 (Distance zero criterion) *Let $t \in T$ be an abnormal transaction and let $N_G^0(t)$ be all of the actors which are involved in this transaction. An actor $a \in V$ is linked to an abnormal transaction $t \in T$ according to the distance zero criterion if $a \in N_G^0(t)$.*

Figures 6.5a and 6.5b give an example of two transactions t_1 and t_3 with their sets of involved actors, respectively $\{a_3, a_6\}$ and $\{a_1, a_6\}$.



(a) Actors at distance zero from transaction t_1 are a_3 and a_6 **(b) Actors at distance zero from transaction t_3 are a_1 and a_6.**

Figure 6.5: Actors linked to abnormal transactions.

We construct the bipartite graph $B = (V', T, L)$ as follows. A pair $\{v, t\} \ : \ t \in T, v \in V$ is an edge of B if $v \in N_G^0(t)$. Figure 6.6 gives an example of a bipartite graph created using this criterion on the graph in Figure 6.4.

Finding suspicious communities

In this subsection the third step is detailed. The use of a maximal biclique enumeration algorithm to find suspicious communities in the transaction database is explained in detail.

We start by giving the definition of maximal biclique in a bipartite graph.

Definition 6.2 (Biclique) *A complete bipartite graph is a bipartite graph $G = (X, Y, E)$ such that $E = \{\{x, y\} \ : x \in X \wedge y \in Y\}$: that is, all possible edges are included in G. A **biclique** in G (where G is not necessarily complete bipartite) is a complete bipartite subgraph $B = (X', Y', E_B)$ of G.*

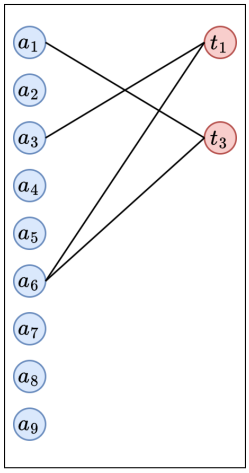


Figure 6.6: The bipartite graph linking actors to abnormal transactions.

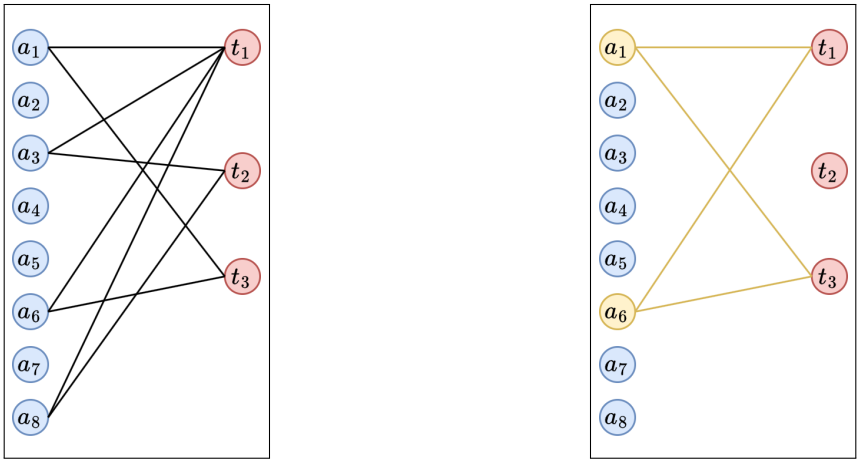
Definition 6.3 (Maximal biclique) *A complete bipartite graph $B = (X', Y', E_B)$ is an inclusion-wise maximal biclique in $G = (X, Y, E_G)$ if there does not exist a vertex $v \in (X \cup Y) \setminus (X' \cup Y')$ such that the induced subgraph $G[X' \cup Y' \cup \{v\}]$ is a biclique.*

For this workshop, suspicious communities are detected using a maximal biclique enumeration algorithm. In a bipartite graph of people and items, it is possible to detect communities by finding maximal bicliques since each community represents a group of people interacting with the same set of items.

One example of this is detecting web communities by creating a bipartite graph where one set of vertices represents web users and one set of vertices represents visited url [2]. A maximal biclique in this graph corresponds to a group of users visiting the same group of urls, meaning that the users have the same interest.

In this workshop the same idea is used to detect suspicious communities of actors in the real estate transactions database. The criterion in Definition 6.1, used to link actors and abnormal transactions in the bipartite graph, implies that a maximal biclique in the bipartite graphs represents a group of actors that are all involved in the same abnormal transactions. Furthermore we enumerate only the bicliques in which each side has more than one vertex: indeed multiple actors linked to the same transaction do not arouse suspicion and neither do multiple transactions linked to the same actor.

Figure 6.7 shows a bipartite graph and one of its maximal bicliques.



(a) A bipartite graph G. **(b) A maximal biclique in G.**

Figure 6.7: Actors linked to abnormal transactions.

An in-house maximal biclique enumeration algorithm was applied to find all the maximal bicliques. To detect abnormal transactions, transactions were clustered geographically, by type, and by year. Quantiles were calculated on the ratio equation (6.1) for each cluster.

Table 6.3 shows the number of maximal bicliques found using the biclique enumeration algorithm when applied to the actors-abnormal transactions bipartite graph. Transactions are considered as suspicious if their ratio is greater than the calculated quantile. For each cluster the two quantiles 0.95 and 0.98 were calculated using the quantile function in the Statistics package of the Julia programming language.

Table 6.3: Number of suspicious communities detected for the two quantile thresholds.

Quantile threshold	Number of Communities
0.95	11 174
0.98	4 926

6.4 Future work

We conclude by outlining further directions for the detection of fraudulent patterns. All implemented pipelines and proofs-of-concept described here are privately made available to the Revenu Québec representatives.

6.4.1 Further exploration of the networks

To facilitate further study of the networks, scripts are provided to generate, from the original CSV files, the adjacency matrices of both networks and their connected components (to separate files). Also recent developments in the CuGraph library [3] allow for the parallelization of multiple graph-related operations. This gain in performance allowed for the testing of more hypotheses regarding the structure of the networks. While we could not achieve useful information extraction during the workshop - outside of generating hypotheses for other approaches - we hope that this proof-of-concept will allow Revenu Québec representatives that have more domain-specific knowledge to find relevant graph metrics in the future.

6.4.2 A graph neural network approach

Graph neural networks are a burgeoning field of study [4], with a wild landscape of various techniques. A particularly interesting technique for detecting statistical outliers in an unsupervised context is the use of Graph Neural Network Autoencoders (GAE) [5].

GAEs are used like convolutional autoencoders to recreate the inputs fed to a model. Instead of recreating an image, however, GAEs can be used to recreate the topological structure of a graph. The differences between the training graph and the resulting recreated graph point to statistical outliers in the former. Indeed a GAE can be thought of as recreating the *true* structure of a graph, where errors in recreation could point to potentially suspicious patterns. Although the main weakness of GAEs is the loss of non-topological data, past [6] and ongoing efforts are made to circumvent this weakness. Figure 6.8 shows an example of outlier detection from a GAE.

6.4.3 Statistical detection of abnormal transactions using machine learning techniques

Following our definition of *suspicious transaction* in Section 6.3.1, we explored more advanced approaches relying on machine learning techniques to detect suspicious actors. We first put aside the transactions with very high/low ratio (ratio=value of the transaction / estimated value of the property); we then used Isolation Forest and K-means clustering (separately) to extract more information from the data. The previous techniques provide a definition, hence detection, of what is abnormal.

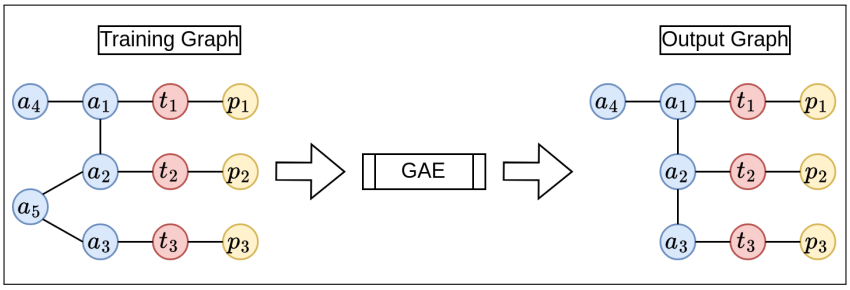


Figure 6.8: An example recreation of a graph by a GAE. It suggests the actor a_5 should not have played a role in transactions between a_2 and a_3. It should therefore be replaced with an edge $e_{a_2a_3}$.

6.4.4 Complete proposed pipeline

Considering the unsupervised approach, maximizing the number of features - and therefore of data points - should be kept in mind. Luckily, there are many possible approaches for feature engineering described in this document that should provide good food for thought for future work. Figure 6.9 shows how all our various approaches can be combined for creating a new enriched database for Revenu Québec representatives.

The idea is that while we proposed metrics for detection of fraudulent patterns, these metrics can still be used as new features for the data set and provide insights for further metrics to analyze or features to define. Furthermore, since the network structures are part of this new database, all these various features and metrics can be used as labels for nodes and edges. This provides the basis for a positive feedback loop, where new metrics can help improve the precision of prior metrics or used methods.

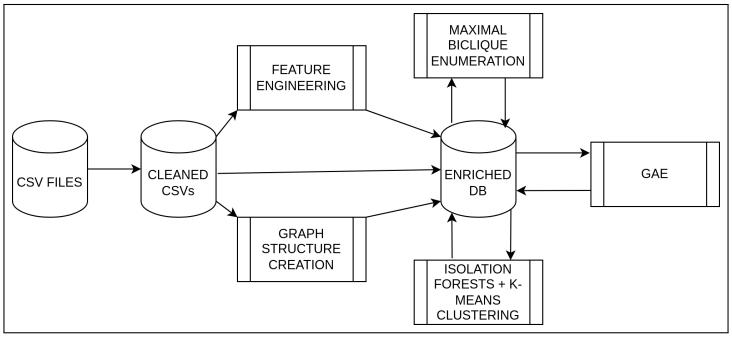


Figure 6.9: A simplified version of the proposed pipeline.

Bibliography

[1] Q. Zheng, Y. Xu, H. Liu, B. Shi, J. Wang, B. Dong, A survey of tax risk detection using data mining techniques, Engineering 34 (2024) 43–59. doi:https://doi.org/10.1016/j.eng.2023.07.014.
URL https://www.sciencedirect.com/science/article/pii/S2095809923003867

[2] T. Murata, Discovery of user communities from web audience measurement data, in: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, WI '04, IEEE Computer Society, USA, 2004, p. 673–676.

[3] A. Fender, B. Rees, J. Eaton, RAPIDS cuGraph, in: Massive Graph Analytics, Chapman and Hall/CRC, 2022, pp. 483–493.

[4] B. Khemani, S. Patil, K. Kotecha, S. Tanwar, A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions, Journal of Big Data 11 (1) (2024) 18.

[5] X. Du, J. Yu, Z. Chu, L. Jin, J. Chen, Graph autoencoder-based unsupervised outlier detection, Information Sciences 608 (2022) 532–550.

[6] M. Lin, K. Wen, X. Zhu, H. Zhao, X. Sun, Graph autoencoder with preserving node attribute similarity, Entropy 25 (4) (2023) 567.