ULB UNIVERSITÉ LIBRE DE BRUXELLES

Faculty of Sciences

Department of Computer Science

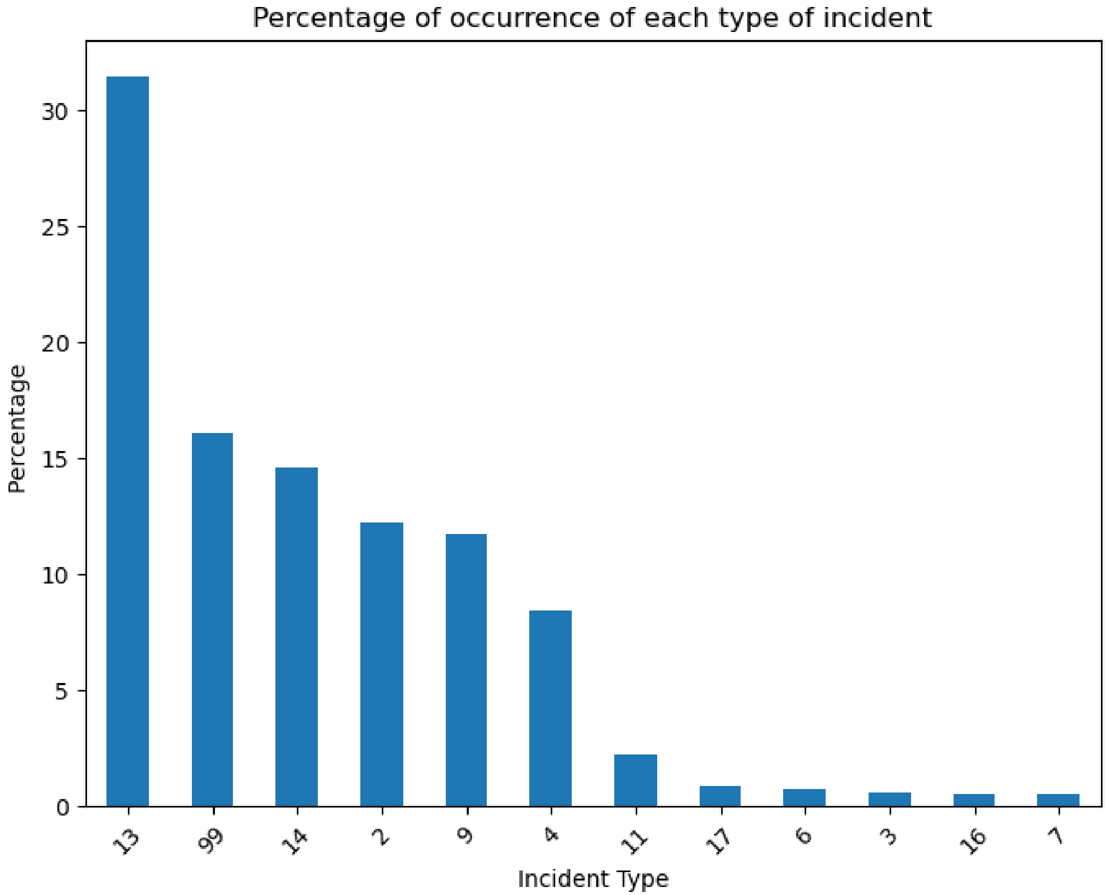
Augmenting train maintenance technicians with automated incident diagnostic suggestions

INFO-H423 : Data Mining

Authors : ARFANI Abdessamad, EHLALOUCH Safouan , MATIN Jordan

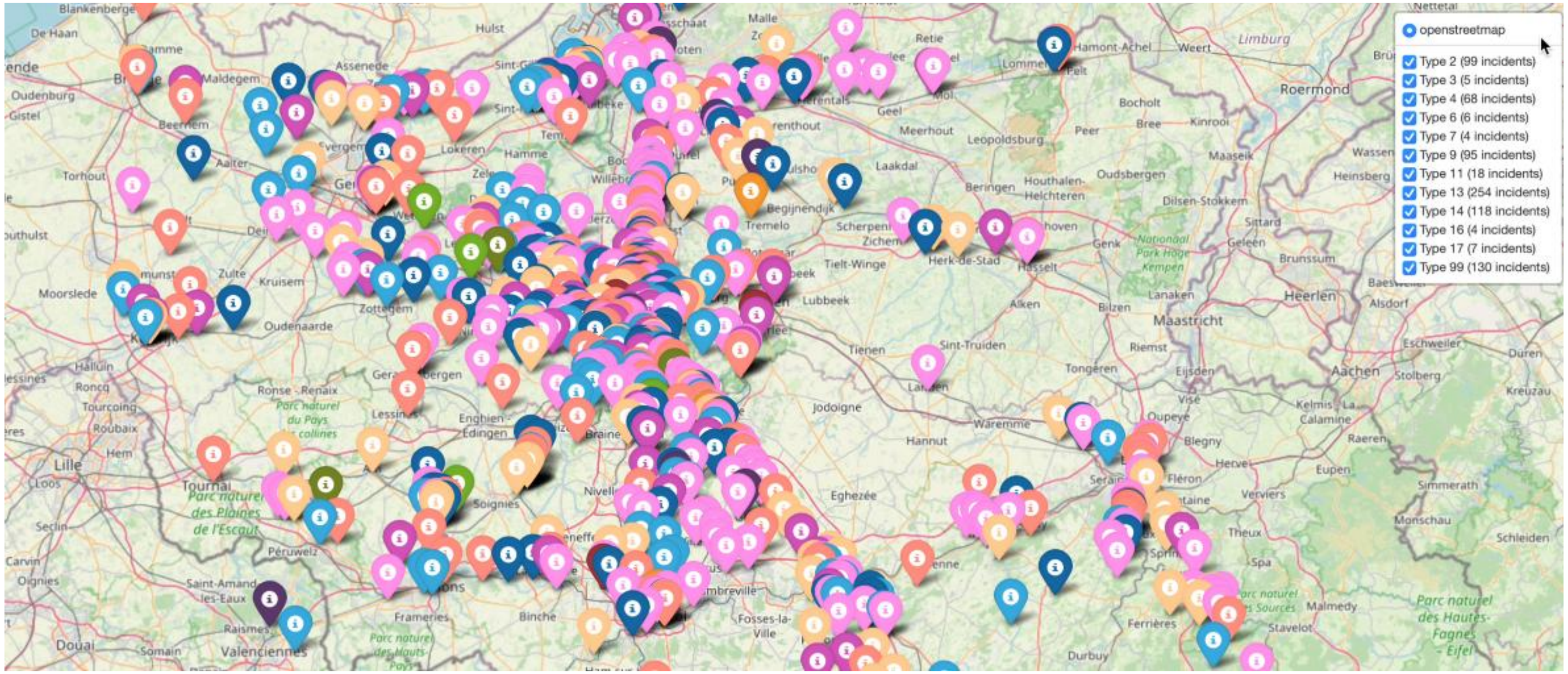
Academic year 2024–2025

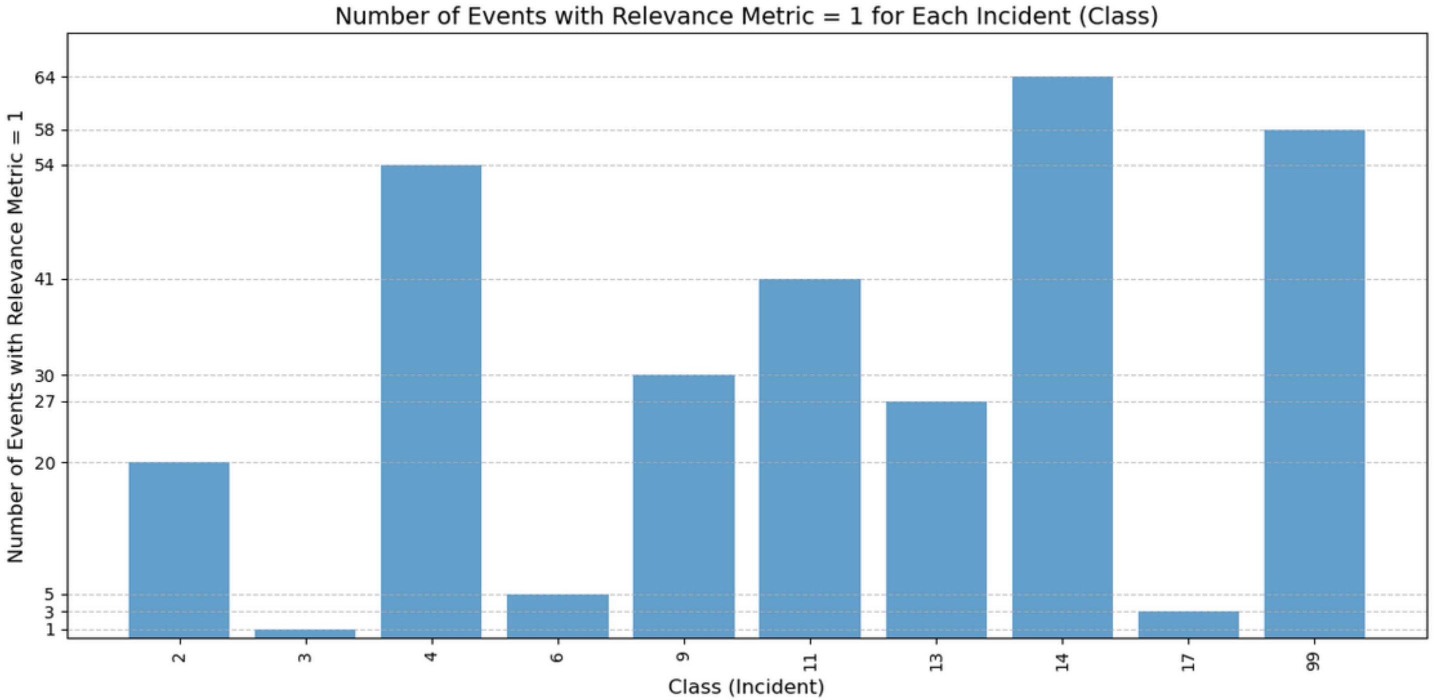
Data management and exploration



- **Unbalanced dataset**
- **Some classes are extremely rare**
- **Yield to the hard classification of minor classes**

Map indicating the locations of all incidents





Some events are exclusively associated with a single class

	Class	Event	Relevance Metric
294	6	2320	1.0
295	6	4248	1.0
296	6	4382	1.0
297	6	4368	1.0
298	6	2714	1.0
	Class	Event	Relevance Metric
299	17	992	1.0
300	17	2514	1.0
301	17	2516	1.0
	Class	Event	Relevance Metric
302	3	1828	1.0

Examples of events that are exclusively associated with a single class

ULB UNIVERSITÉ LIBRE DE BRUXELLES

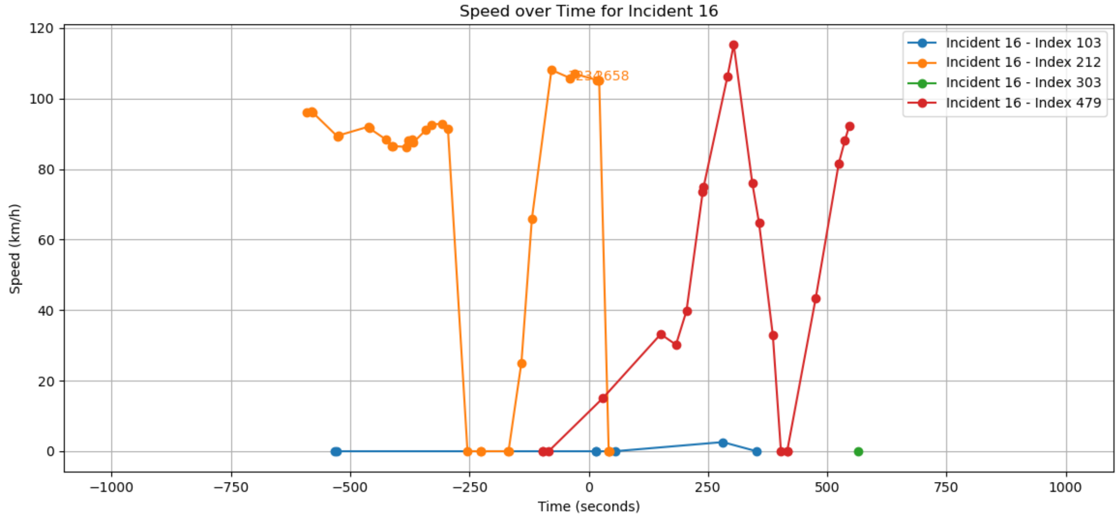
Limitation of the paper's approach

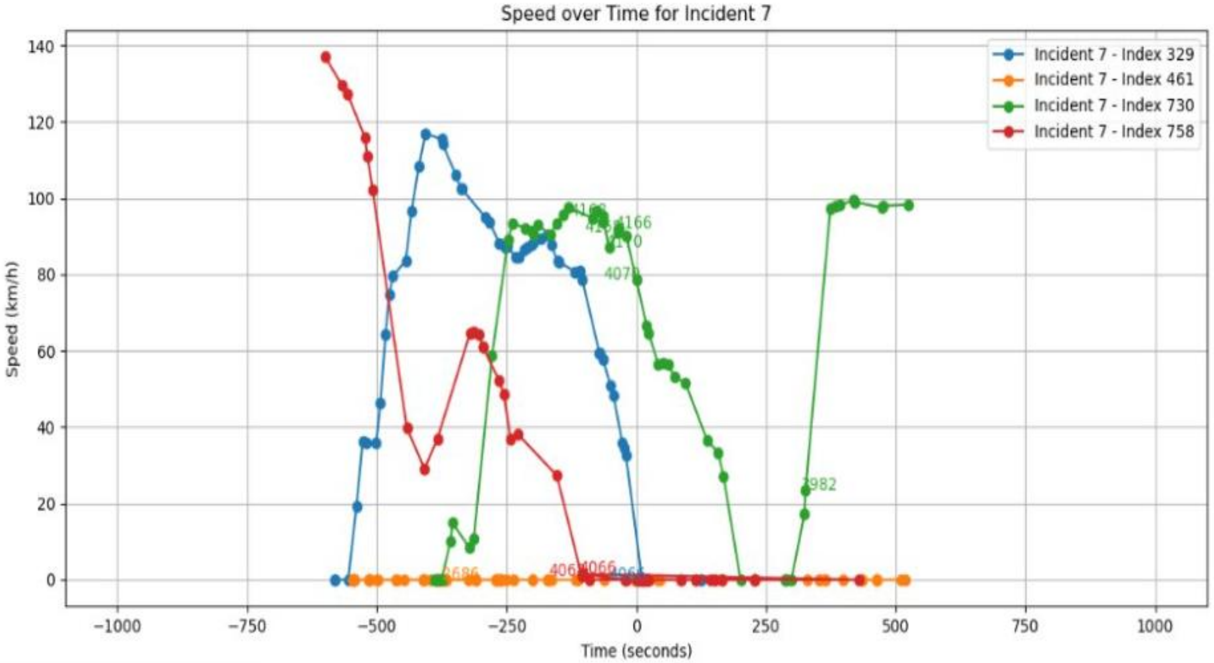
- The context of events is **not** taken into account: train speed, catenary tensions, etc.
- **Understanding Incident Scenarios**: to create remote diagnostic alerts with ***deterministic rules***

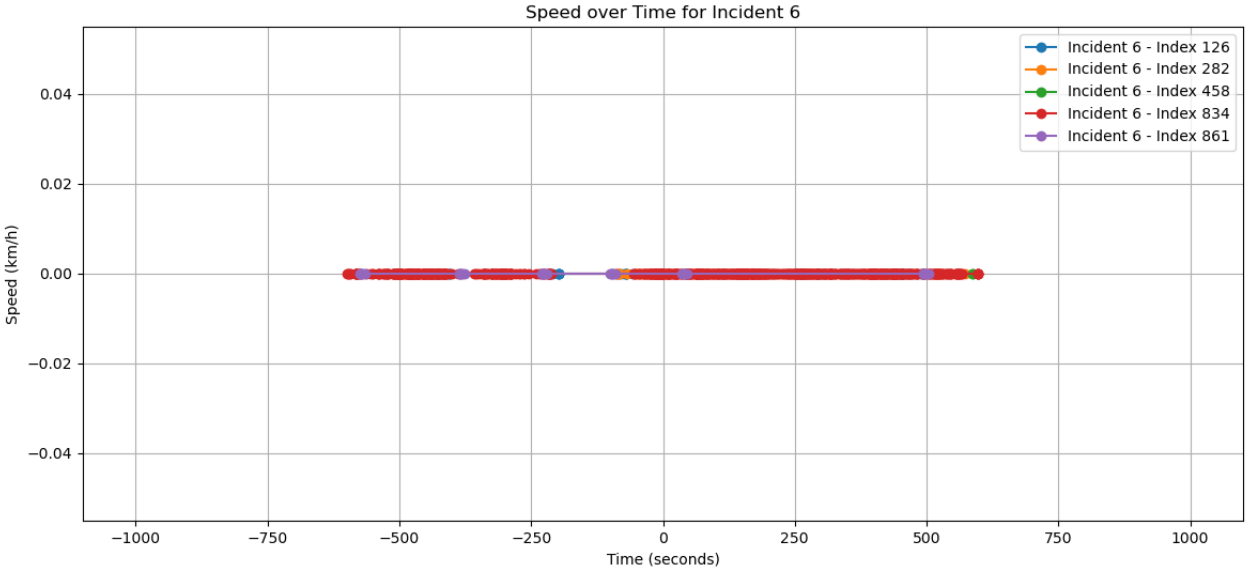
- => Investigation of others data in the dataset.

- **Goals:**
  - (i) identify patterns that could help predict incidents
  - (ii) support the creation of more effective diagnostic alerts.

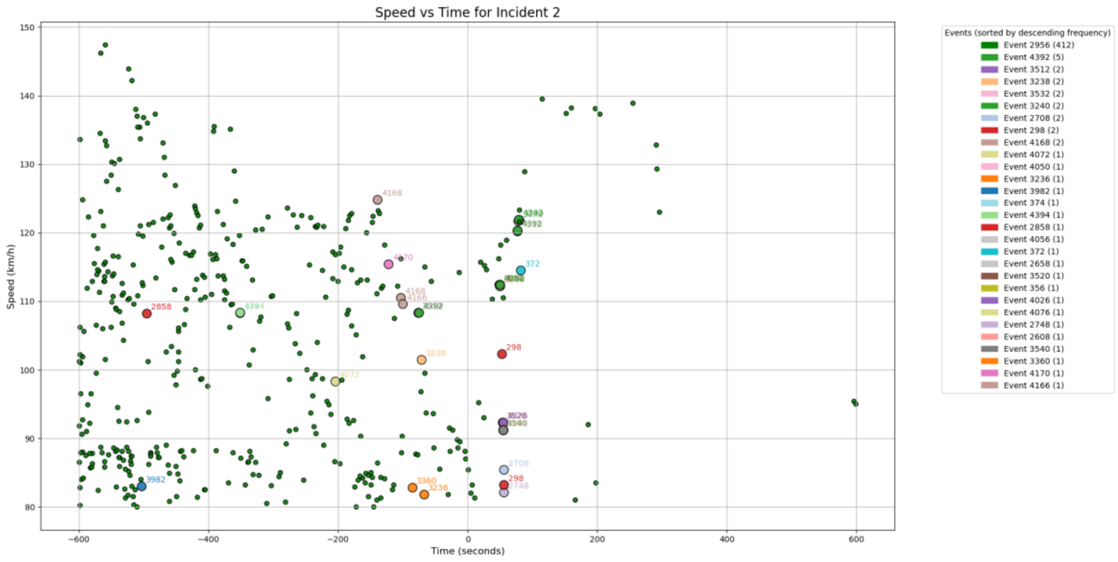
ULB UNIVERSITÉ LIBRE DE BRUXELLES



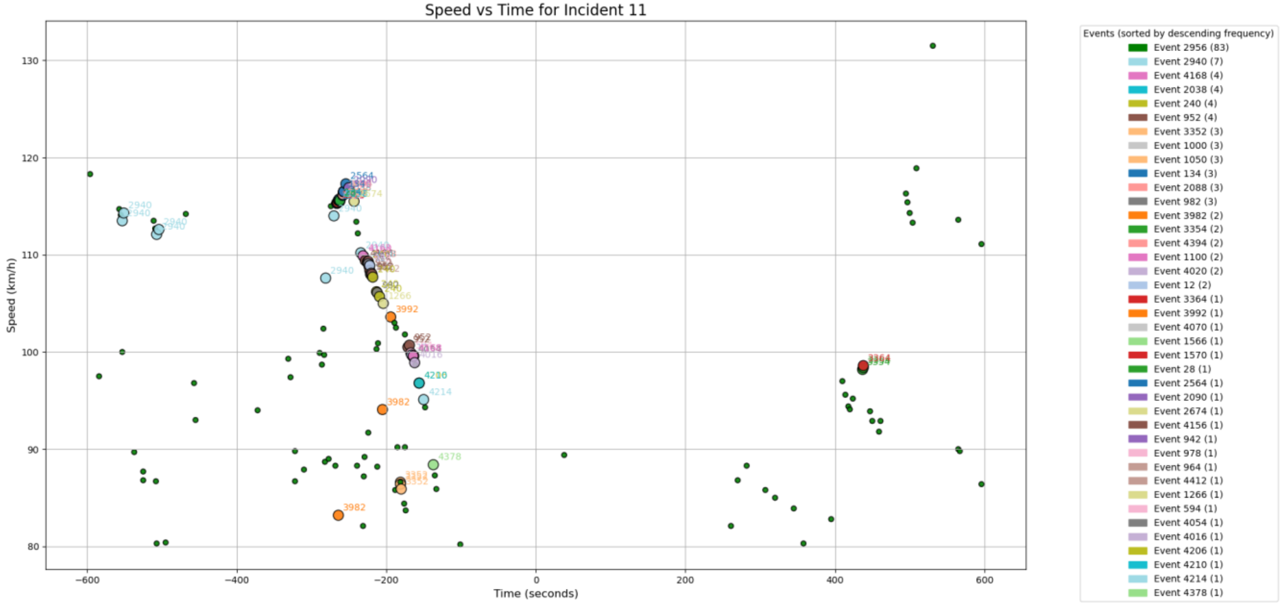




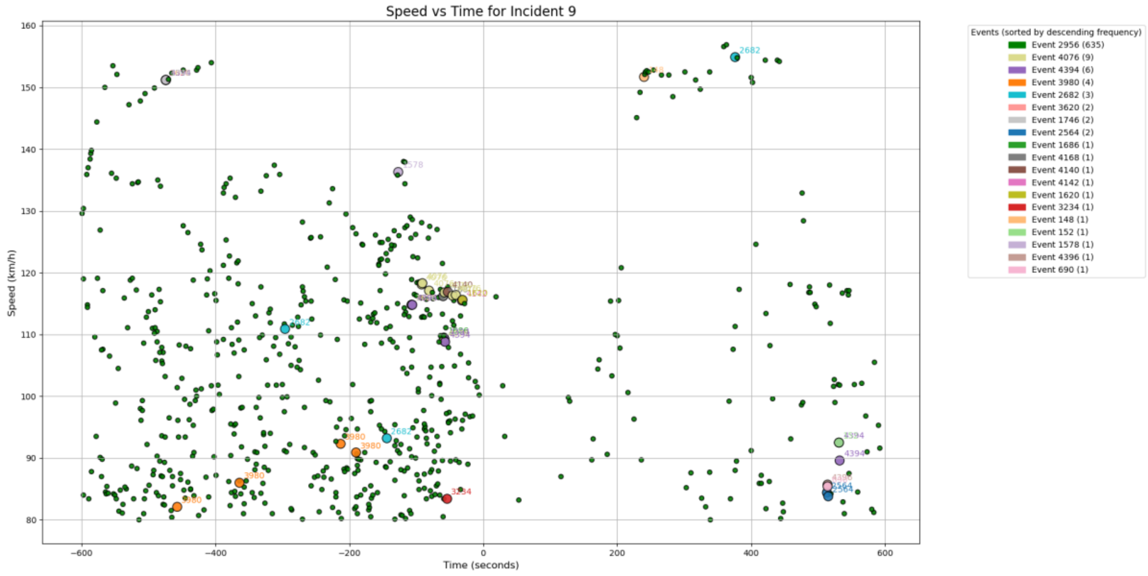
Graphs showing the speed of railways at the time of rare incidents



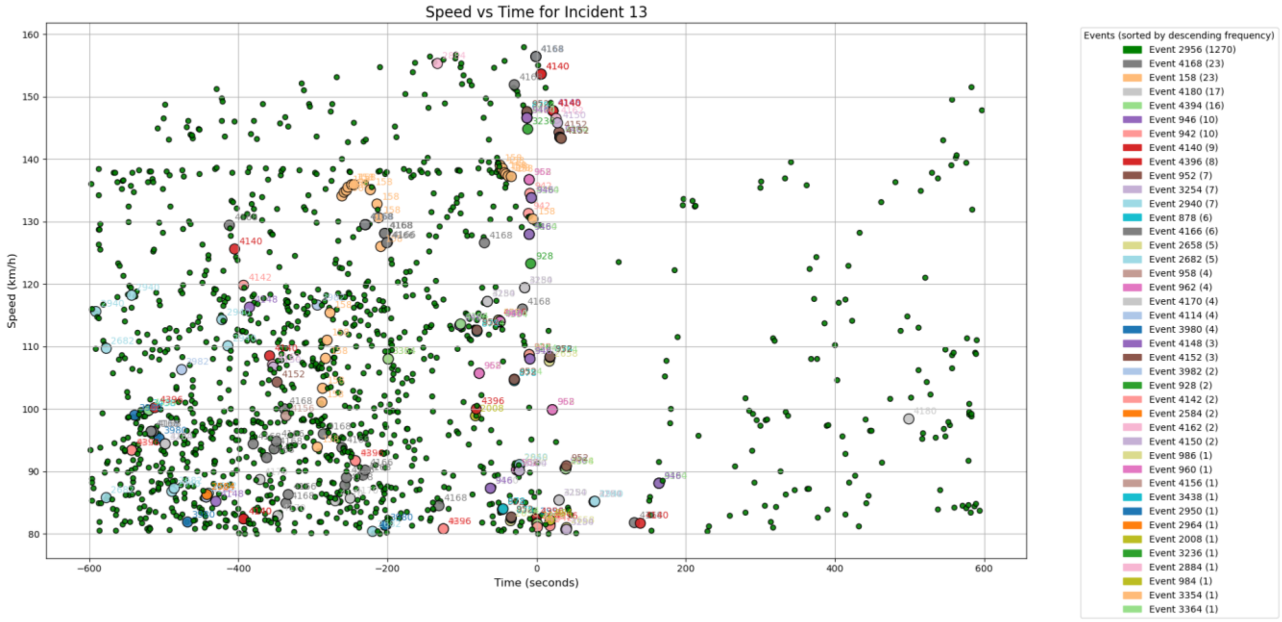
Speed vs Time for Incident 2
Speed (km/h)
Time (seconds)
Events (sorted by descending frequency)
Event 2956 (412)
Event 4392 (5)
Event 3512 (2)
Event 3238 (2)
Event 3532 (2)
Event 3240 (2)
Event 2708 (2)
Event 298 (2)
Event 4168 (2)
Event 4072 (1)
Event 4050 (1)
Event 3236 (1)
Event 3982 (1)
Event 374 (1)
Event 4394 (1)
Event 2858 (1)
Event 4056 (1)
Event 372 (1)
Event 2658 (1)
Event 3520 (1)
Event 356 (1)
Event 4026 (1)
Event 4076 (1)
Event 2748 (1)
Event 2608 (1)
Event 3540 (1)
Event 3360 (1)
Event 4170 (1)
Event 4166 (1)



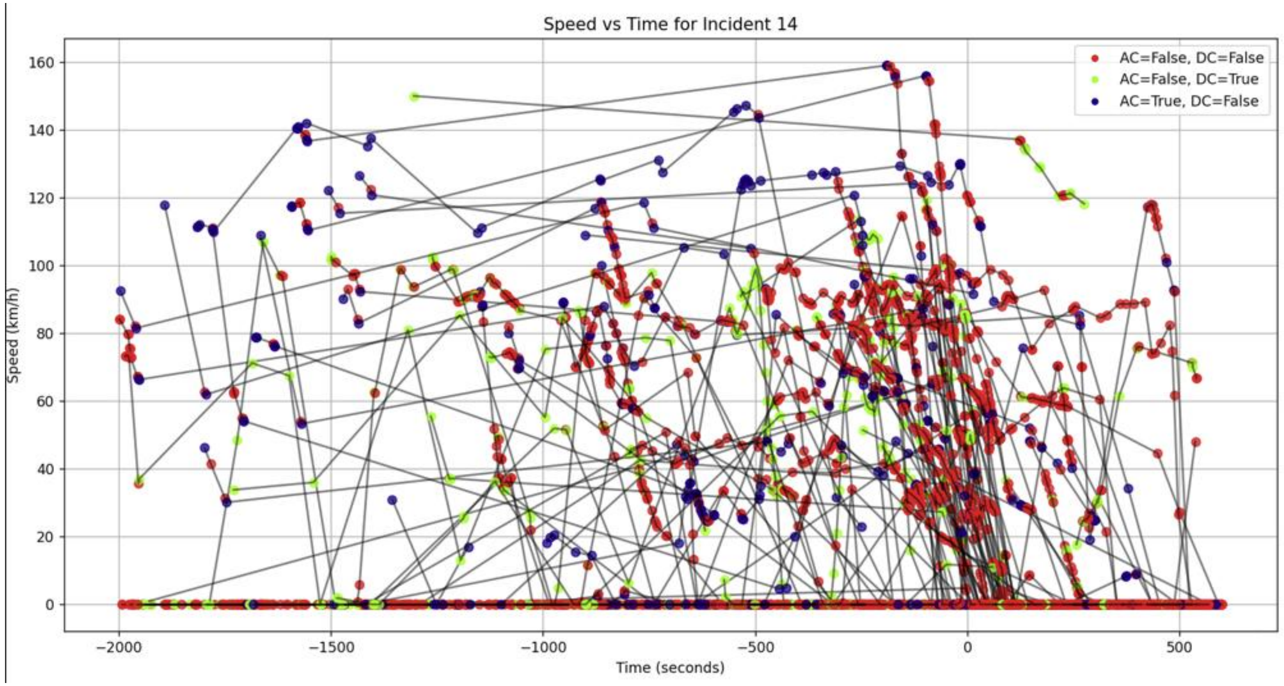
Speed vs Time for Incident 11
Speed (km/h)
Time (seconds)
Events (sorted by descending frequency)
Event 2956 (83)
Event 2940 (7)
Event 4168 (4)
Event 2038 (4)
Event 240 (4)
Event 952 (4)
Event 3352 (3)
Event 1000 (3)
Event 1050 (3)
Event 134 (3)
Event 2088 (3)
Event 982 (3)
Event 3982 (2)
Event 3354 (2)
Event 4394 (2)
Event 1100 (2)
Event 4020 (2)
Event 12 (2)
Event 3364 (1)
Event 3992 (1)
Event 4070 (1)
Event 1566 (1)
Event 1570 (1)
Event 28 (1)
Event 2564 (1)
Event 2090 (1)
Event 2674 (1)
Event 4156 (1)
Event 942 (1)
Event 978 (1)
Event 964 (1)
Event 4412 (1)
Event 1266 (1)
Event 594 (1)
Event 4054 (1)
Event 4016 (1)
Event 4206 (1)
Event 4210 (1)
Event 4214 (1)
Event 4378 (1)

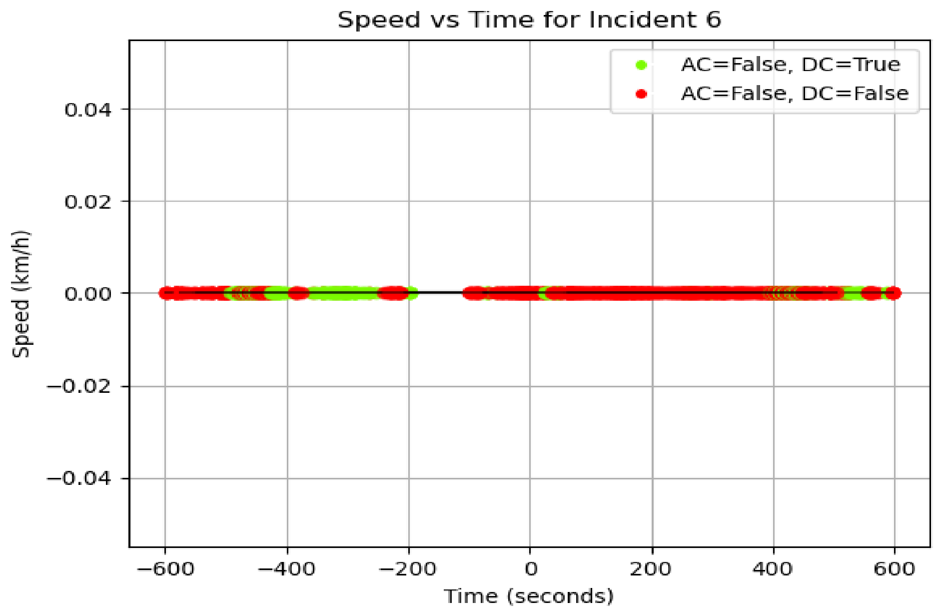


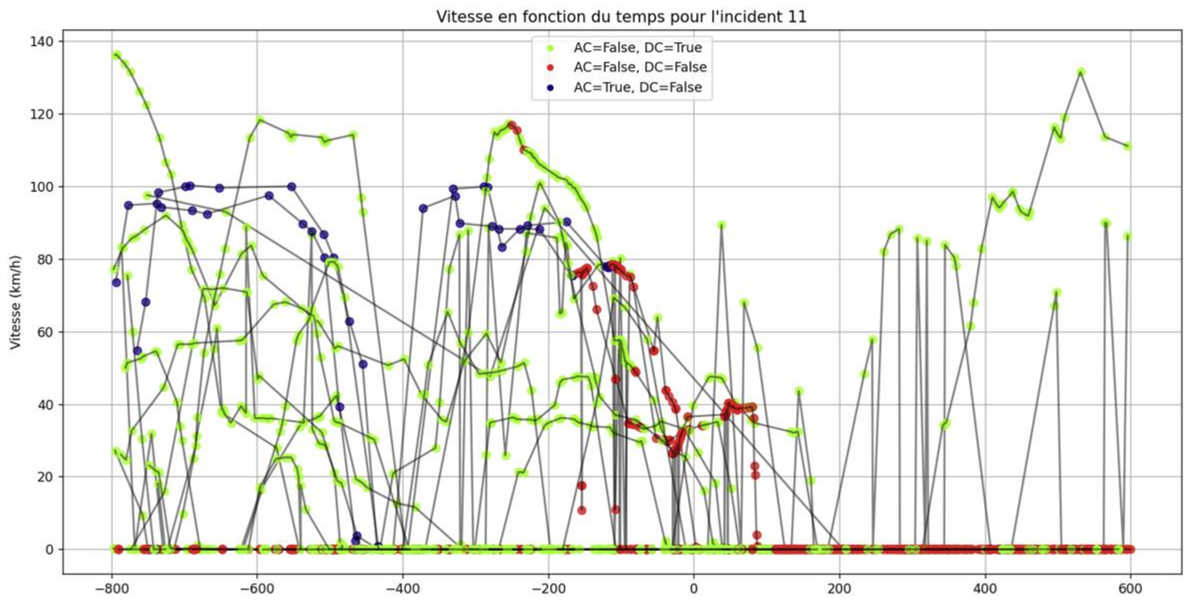
Speed vs Time for Incident 9
Speed (km/h)
Time (seconds)
Events (sorted by descending frequency)
Event 2956 (635)
Event 4076 (9)
Event 4394 (6)
Event 3980 (4)
Event 2682 (3)
Event 3620 (2)
Event 1746 (2)
Event 2564 (2)
Event 1686 (1)
Event 4168 (1)
Event 4140 (1)
Event 4142 (1)
Event 1620 (1)
Event 3234 (1)
Event 148 (1)
Event 152 (1)
Event 1578 (1)
Event 4396 (1)
Event 690 (1)

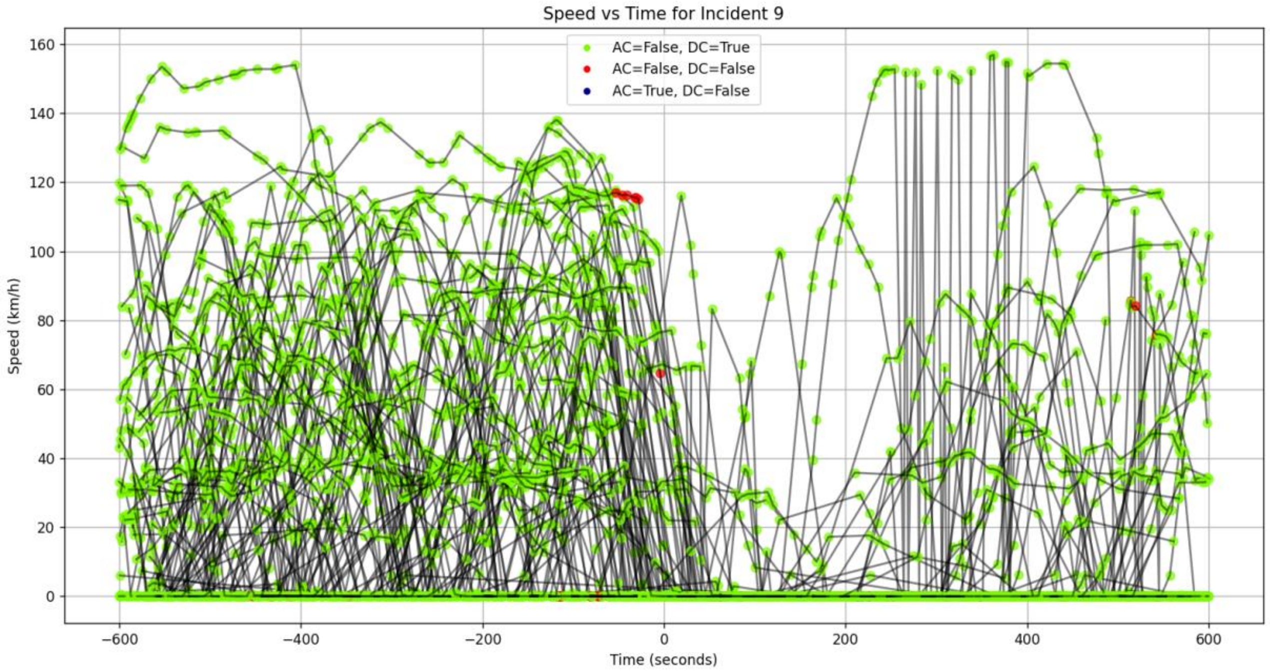


Speed vs Time for Incident 13
Speed (km/h)
Time (seconds)
Events (sorted by descending frequency)
Event 2956 (1270)
Event 4168 (23)
Event 158 (23)
Event 4180 (17)
Event 4394 (16)
Event 946 (10)
Event 942 (10)
Event 4140 (9)
Event 4396 (8)
Event 952 (7)
Event 3254 (7)
Event 2940 (7)
Event 878 (6)
Event 4166 (6)
Event 2658 (5)
Event 2682 (5)
Event 958 (4)
Event 962 (4)
Event 4170 (4)
Event 4114 (4)
Event 3980 (4)
Event 4148 (3)
Event 4152 (3)
Event 3982 (2)
Event 928 (2)
Event 4142 (2)
Event 2584 (2)
Event 4162 (2)
Event 4150 (2)
Event 986 (1)
Event 960 (1)
Event 4156 (1)
Event 3438 (1)
Event 2950 (1)
Event 2964 (1)
Event 2008 (1)
Event 3236 (1)
Event 2884 (1)
Event 984 (1)
Event 3354 (1)
Event 3364 (1)


Speed vs Time for Incident 14
AC=False, DC=False
AC=False, DC=True
AC=True, DC=False
Speed (km/h)
Time (seconds)


Speed vs Time for Incident 6
AC=False, DC=True
AC=False, DC=False
Speed (km/h)
Time (seconds)


Vitesse en fonction du temps pour l'incident 11
AC=False, DC=True
AC=False, DC=False
AC=True, DC=False
Vitesse (km/h)


Speed vs Time for Incident 9
AC=False, DC=True
AC=False, DC=False
AC=True, DC=False
Speed (km/h)
Time (seconds)

Model One: Random Forest Classifier

- The goal is to find the optimal threshold that maximizes the **F1-score** for incident classification.

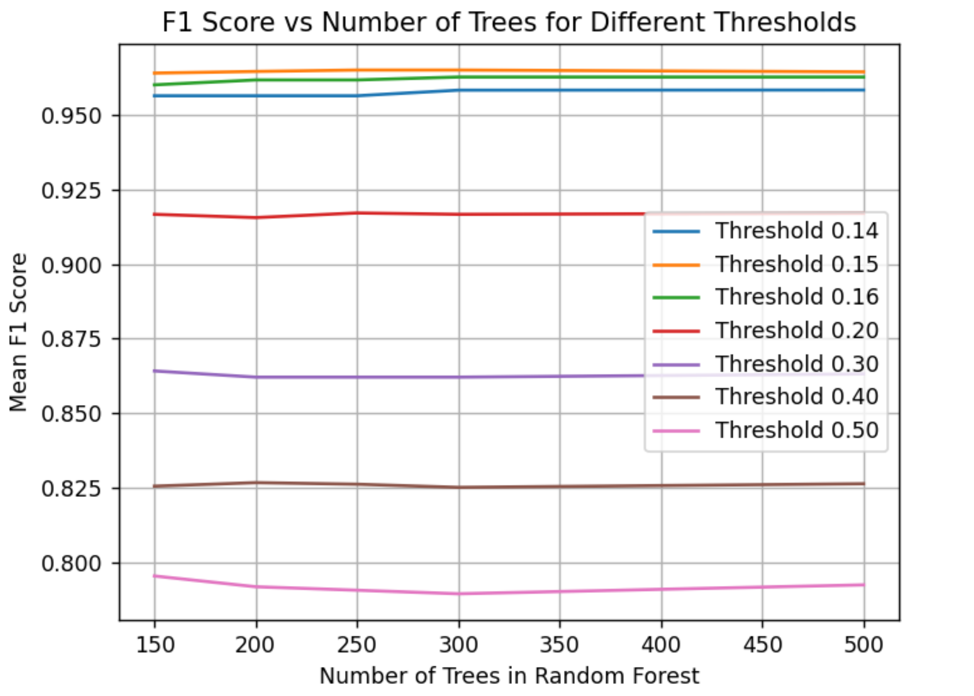
- The model involves:
  - Stratified cross validation with 10-folds.
  - **Data preparation:** event filtering based on a relevance metric and time (4 hours before / 10 min after).
  - Threshold tuning for optimal classification performance.

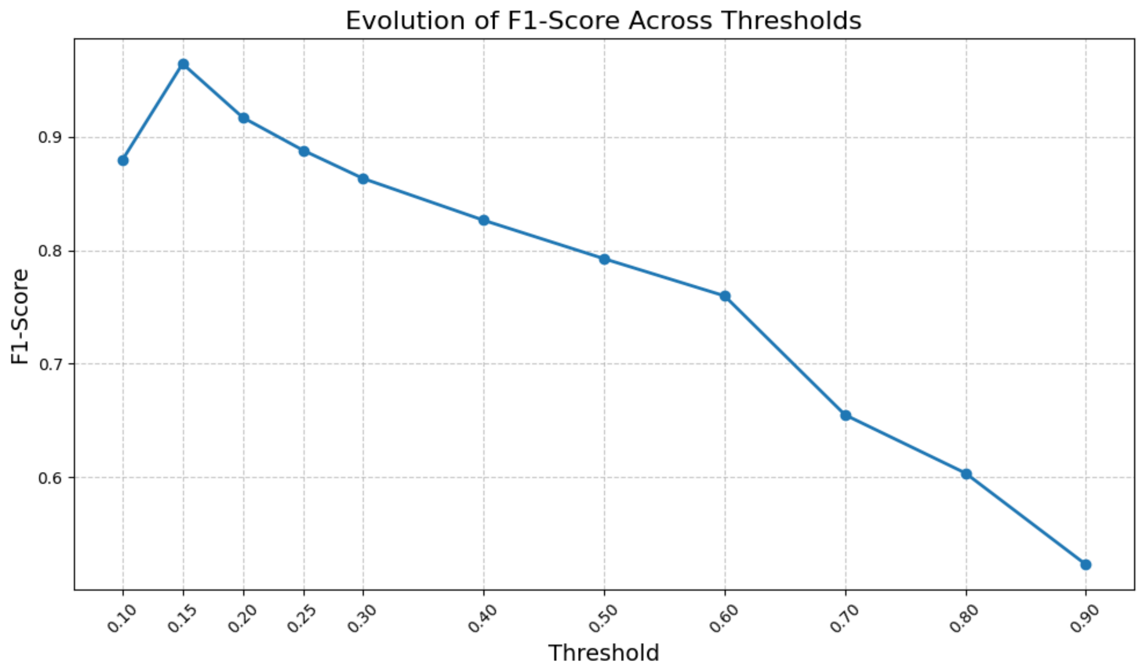
- **Balanced class weight**: ensure fairness across imbalanced classes (weights inversely proportional to class frequencies).

- **Feature matrix** constructed by one-hot encoding:
  - Each vector has a length equal to the number of unique events
  - `1` indicates that the event is present for that incident.
  - `0` indicates that the event is absent.

Tuning threshold and number of trees parameter

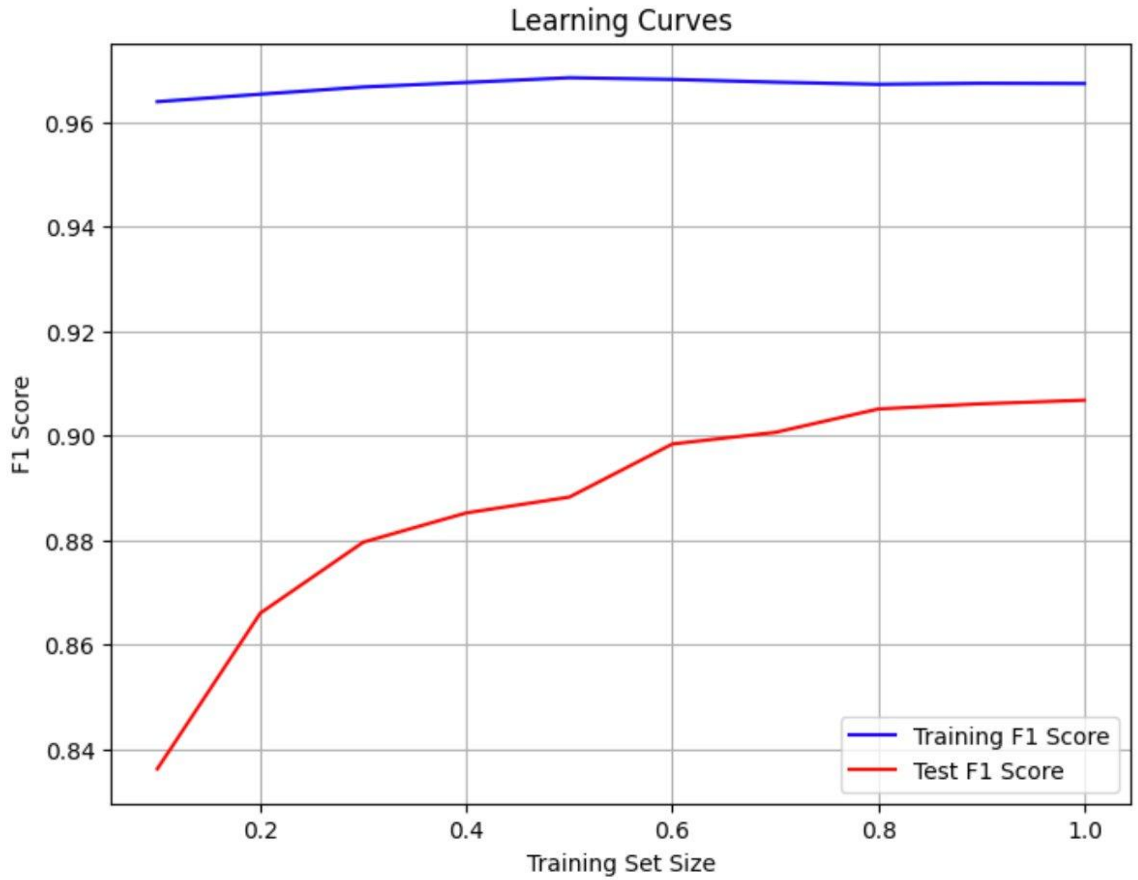
- The best F1-score found is **0.96** with threshold **0.15**
- Threshold set to **0.2** (F1-score greater than **0.90**, which maximizes the number of events retained).





ULB UNIVERSITÉ LIBRE DE BRUXELLES

Is the Random Forest Classifier overfitting ?



- **The training F1 score is consistently high, showing strong performance on seen data**
- **The test F1 score improves as the training size increases, indicating better generalization**
- **The difference in F1-score between the train and test sets is small, indicating good generalization**

ULB UNIVERSITÉ LIBRE DE BRUXELLES

Classification Report:

	precision	recall	f1-score	support
2	1.00	1.00	1.00	12
3	0.11	1.00	0.20	1
4	1.00	1.00	1.00	8
6	0.00	0.00	0.00	1
9	1.00	0.75	0.86	12
11	1.00	0.67	0.80	3
13	1.00	1.00	1.00	32
14	0.93	0.93	0.93	14
17	0.00	0.00	0.00	1
99	1.00	0.88	0.94	17
accuracy			0.91	101
macro avg	0.70	0.72	0.67	101
weighted avg	0.96	0.91	0.93	101

- **Not all types of incidents are predicted with the same precision**
- **Some classes, such as '17', are not predicted at all (likely due to their extreme rarity)**
- **'3' have very low precision despite being detected**
- **Additionally, some classes, such as '7', are not reported at all**

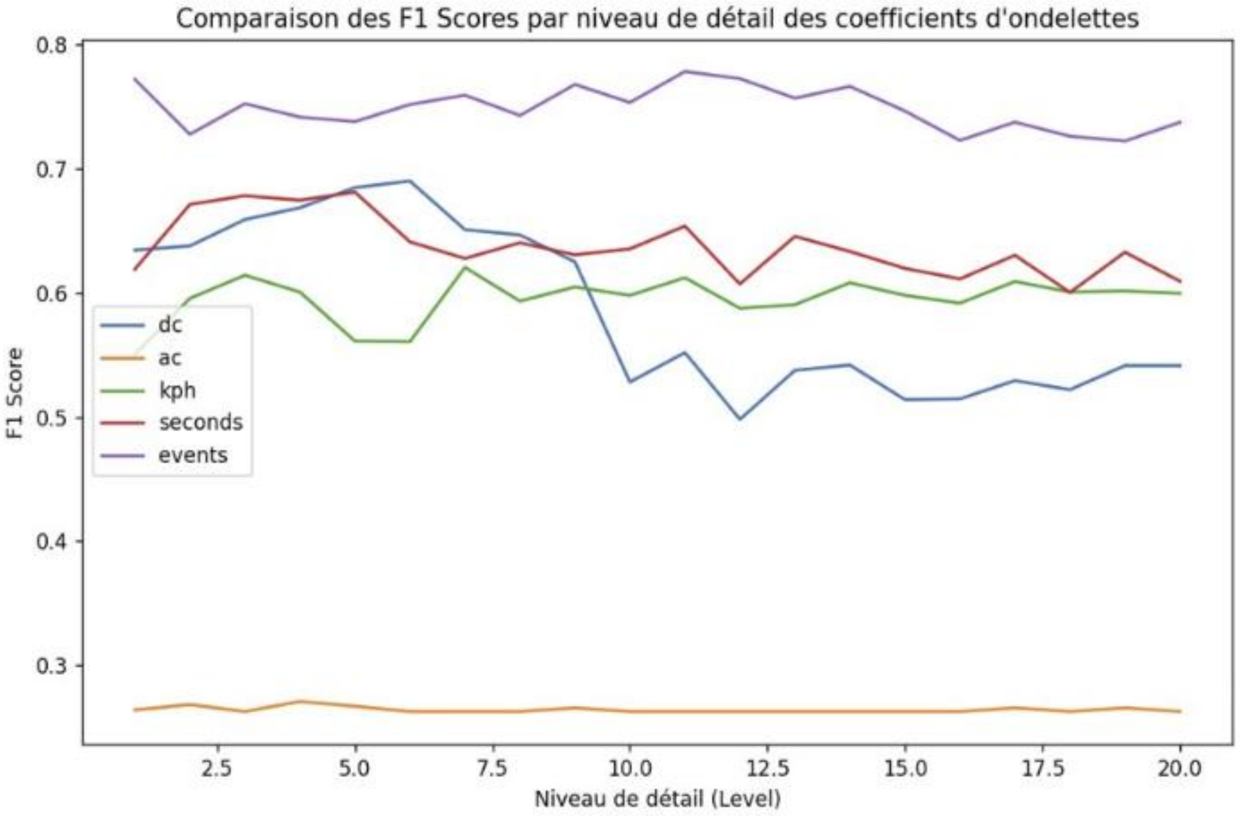
Why Choose Random Forest Classifier?

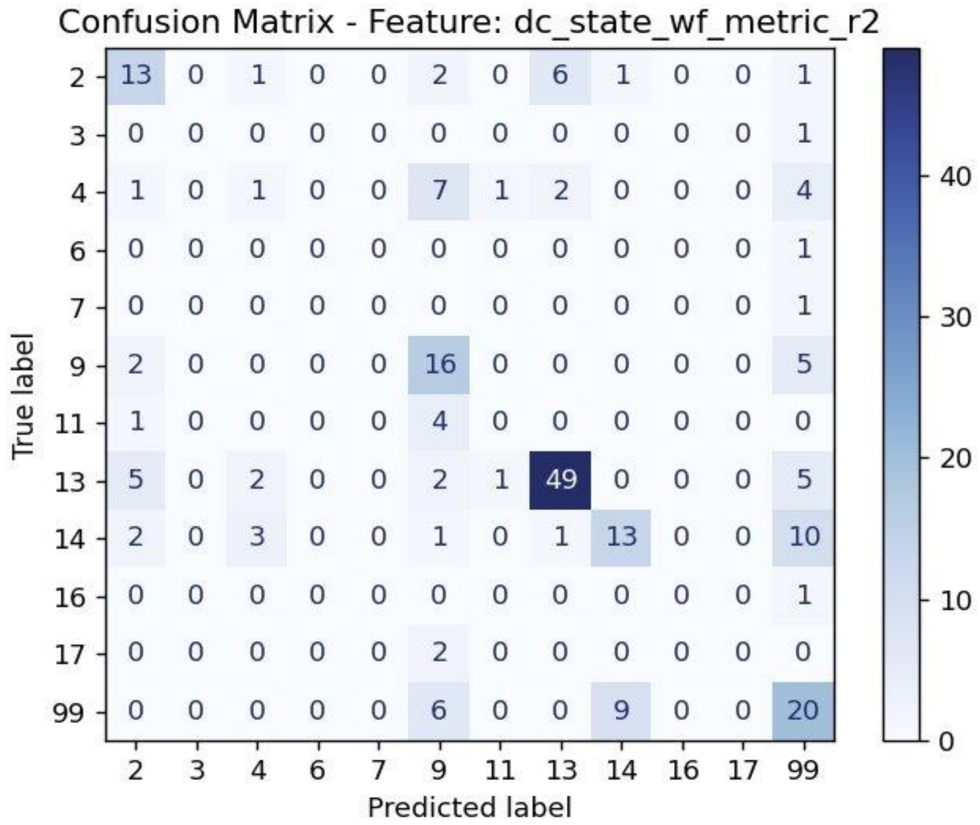
- Robust to overfitting due to (i) ensemble learning and (ii) bias-variance tradeoff.

- Handles imbalanced datasets
  - Class Weight Adjustment

- Scalable and computationally efficient
  - Handles high-dimensional data: one-hot encoding increases the number of features
  - Parallelizable
- **Major drawback:** not suited for time-series data
- **Improvement:** Integrating **discrete wavelet transforms** with **Random Forests (RF)**

Wavelet transforms with RF





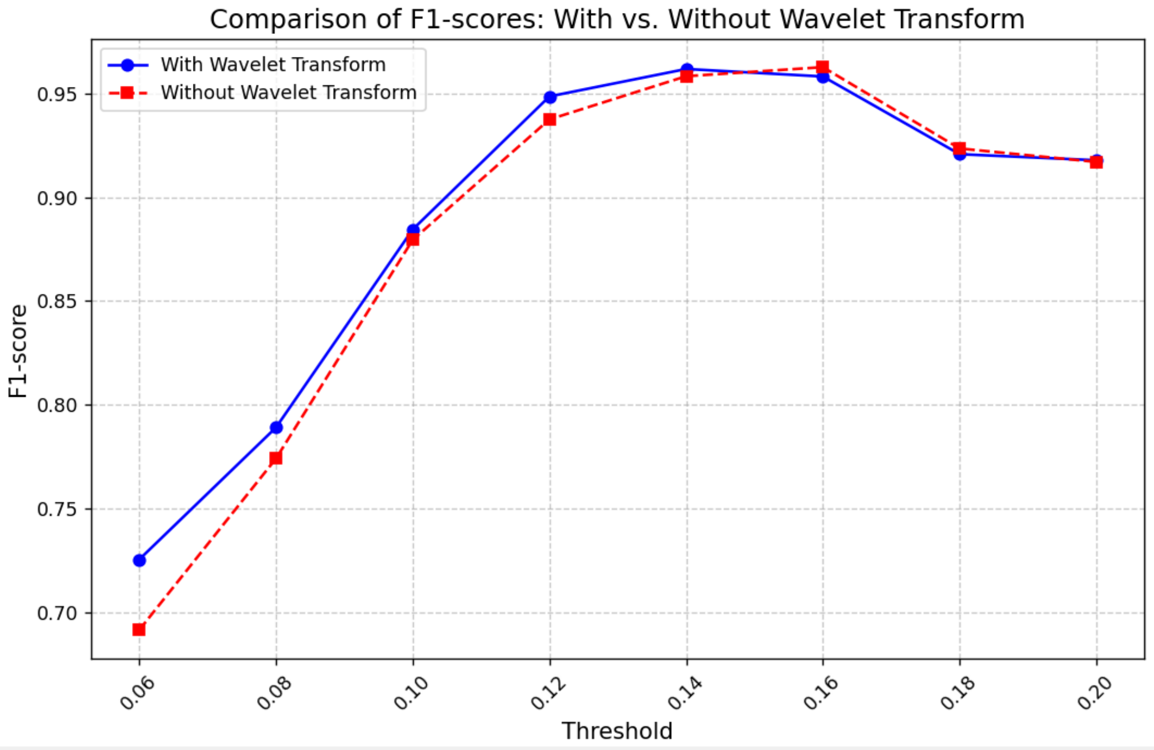
Wavelets transforms with RF

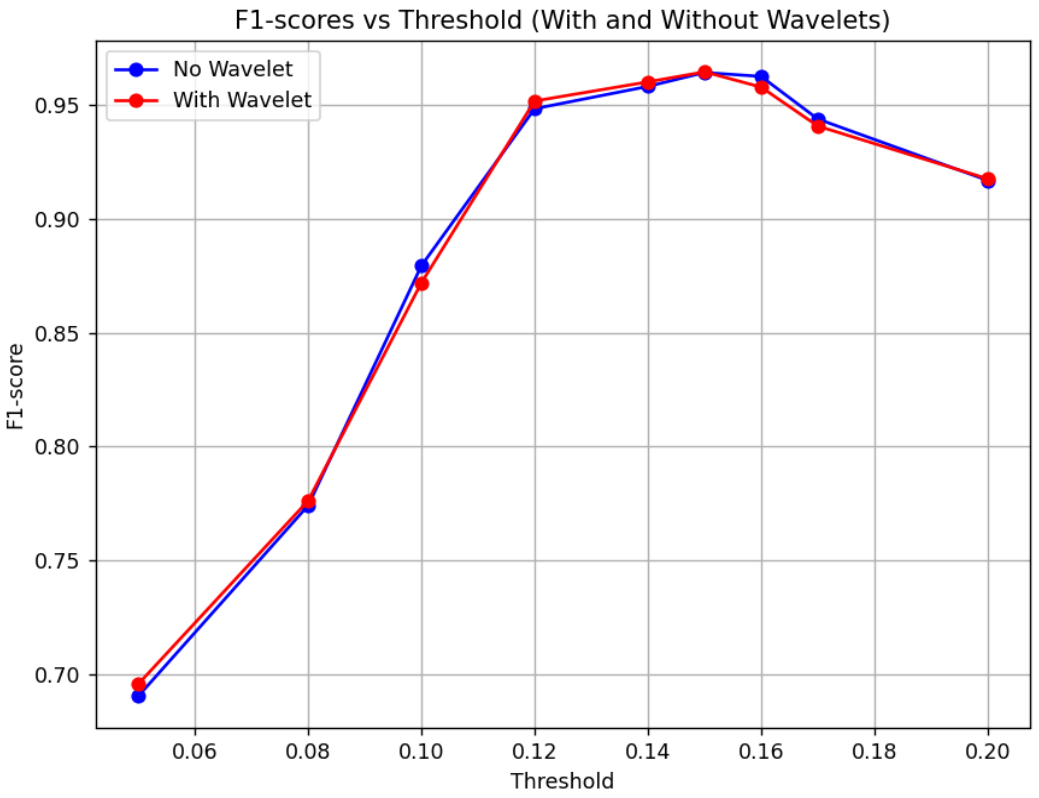
- **Comparison of F1-score**: KPH, seconds, DC-AC state, and events.
- **Replaced events with relevance metric:** Event occurrence in a class as a measure of frequency. F1 score remains high, but not as high as with the one-hot encoding.
- **DC-state matters** as much (if not more) than speed and time before incident.
- **DWT on one-hot encoded events**: Slightly improved F1 score depending on wavelet, threshold, and level chosen.
- **Wavelet performance:** Haar wavelet generally outperforms Daubechies (DB1, DB4, etc.) in average F1 score.

ULB UNIVERSITÉ LIBRE DE BRUXELLES

Haar Wavelet

Daubechie Wavelet (Daub4)





Model Two: Original Naïve Bayes + Single Class occurence

- **Some events are only occuring in only one class (more after the filtering)**
  => use this fact to build our second model and incorporate it in the Naïve Bayes classifier

- **Pseudo code of our proposed model:**

```
def single_occurence_event() :
    single_occurence = []
    for i in range(n) :
        find events occurring in only one class ;
        add to single_occurence
def recognize() :
    if event is recognized :
        return the class
    else :
        Do Naive Bayes Classifier paper
```

Evolution of the F1-score with increasing cascaded windows size

- From a certain number of windows (ID 5), performance stagnates and decreases.
- The windows ID 5 correponds to the configuration **([(-150.0, 150.0), (-300.0, 300.0), (-450.0, 450.0), (-600.0, 600.0), (-900, 600)]**
- Consistent with the paper (the larger the windows are, the more the F1-score drops)

Problem: Computation efficiency

- **Computation time**: 30 mins to 1h to calculate all LLCSS in a training set for **one** fold
  => Requires powerful computational resources.

- **Another (faster) approach:** Focused on events that occur only once in the training set.
  - If an event is recognized in the test set, its class is returned with a probability of 1 (no calculation needed).

- **Results:**
  - **Speed**: 6 to 15 times faster using this approach.
  - **Performance**: F1 score increased by 5.5%.
  - Approximately **50% of events** are categorized with this method.

F1-score comparison between models

	F1-Score	Rare events
Random Forest	96 % (0.15) 91 % (0.2)	Poorly classified
Original naïve bayes (our modified version)	83%(conf. 1) 84% (conf. 5)	Poorly classified
Original ensemble naïve bayes (paper)	65% (M7) 85% (AM08)	Poorly classified
Random Forest with Wavelet	96.8 (0.15) 91.1 (0.2)	Poorly classified

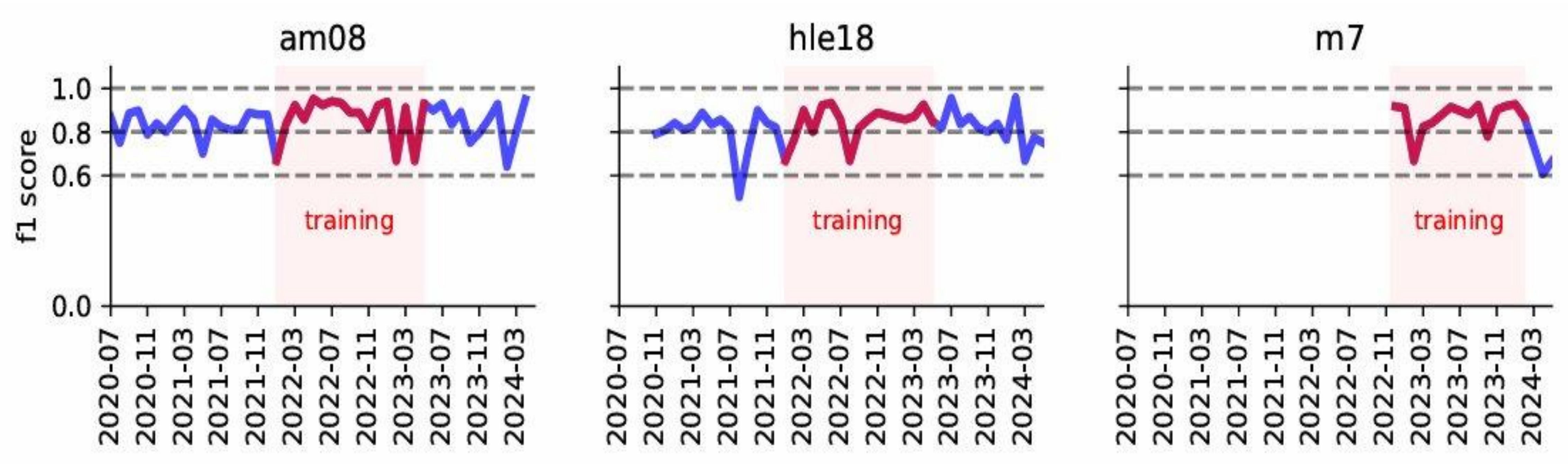


Figure 5: Learning machine performance: descriptive (red) and predictive (blue) performances across three different fleets (AM08, HLE18 and M7): typically the F_1-score is high. Nevertheless some incidents are poorly classified even during training. The M7 is a very recent fleet which explains why there is less data.

Analysis of the formula provided in the paper for the Bayesian classifier

1 Introduction

The classifier formula is given by:

$$c_k = \arg\max_j \left(p(c_j) \prod_{i=1}^{n_{x_k}} p(x_{k_i} \mid c_j) \right)$$

where:

$$p(x_{k_i} \mid c_j) = \frac{\mathrm{card}(x_{k_i} \mid c_j) + \beta}{\beta \cdot n_{x_k} + \sum_i \mathrm{card}(x_{k_i} \mid c_j)}$$

2 Example

Consider the following window x_k:

$$[1802, 1808, 1826, 1828, 1802, 1826, 1828, 1802, 1808, 1826, 1828]$$

And the dictionary containing events that occur only in one class:

$$3 : \{1802 : 37, 1808 : 36, 1822 : 39, 1828 : 38\}$$

Here, $n_{x_k} = 4$.

2.1 Case for $c_j \neq 3$

For $c_j \neq 3$, since the events $1802, 1808, 1826, 1828$ do not occur in c_j, their probabilities are:

$$p(1802 \mid c_j) = p(1808 \mid c_j) = p(1826 \mid c_j) = p(1828 \mid c_j) = \frac{\beta}{4\beta}$$

2.2 Case for $c_j = 3$

For $c_j = 3$, using the values from the dictionary, we compute:

$$p(1802 \mid 3) = \frac{\beta + 37}{4\beta + 36 + 37 + 38 + 39} = 0.2466$$

Similarly:

$$p(1808 \mid 3) = \frac{\beta + 36}{4\beta + 36 + 37 + 38 + 39} = 0.2400$$

$$p(1826 \mid 3) = \frac{\beta + 38}{4\beta + 36 + 37 + 38 + 39} = 0.2533$$

$$p(1828 \mid 3) = \frac{\beta + 39}{4\beta + 36 + 37 + 38 + 39} = 0.2600$$

2.3 Classifier Calculation

Plugging the probabilities into the classifier formula:

For $c_j = 3$:

$$p(c_j)\prod_{i=1}^{n_{x_k}} p(x_{k_i} \mid c_j) = 0.0049456 \cdot 0.2400 \cdot 0.2466 \cdot 0.2533 \cdot 0.2600 = 0.00001928$$

For $c_j \neq 3$:

$$p(c_j)\prod_{i=1}^{n_{x_k}} p(x_{k_i} \mid c_j) = p(c_j) \cdot \left(\frac{1}{4}\right)^4$$

Since the class 13 has the largest probability with $p(13) = 0.33$, we calculate:

$$p(13) \cdot \left(\frac{1}{4}\right)^4 = 0.33 \cdot \frac{1}{256} = 0.00128906$$

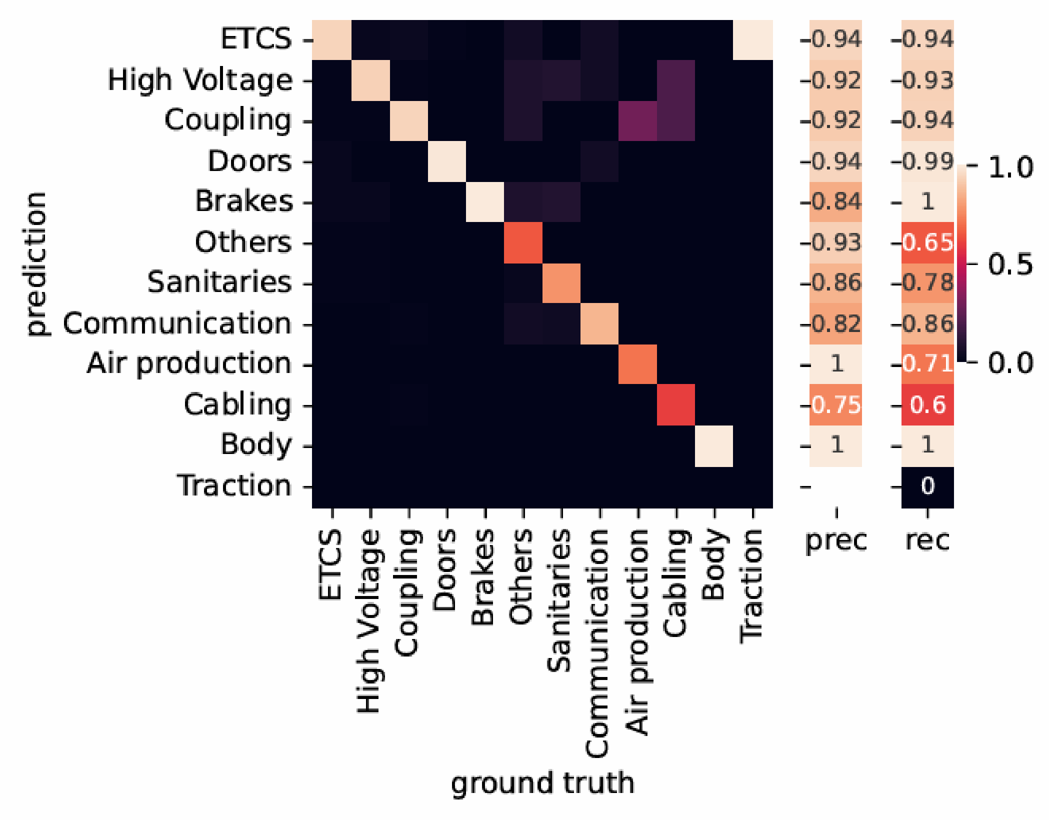
Thus, the highest probability among the classes is achieved for $c_j = 13$ with $p(13) = 0.33$.

3 Conclusion

Based on the calculations, the classifier would select the class with the highest probability, which in this case is $c_j \neq 3$.

Bonus: match incident ID to real labels

2
14
4
5
13
99
11
3
7
16
17
6



Any questions ?

Reference

- **Augmenting train maintenance technicians with automated incident diagnostic suggestions** ; Georges Tod, Jean Bruggeman, Evert Bevernage, Pieter Moelans, Walter Eeckhout and Jean-Luc Glineur.