



Développement de modèles de prévision en R pour l'identification de bris sur les navires de la Marine royale canadienne

J.-D. Caron, M. Fee et B. Khaddaj

Recherche et développement pour la défense Canada (RDDC)

Centre d'analyse et de recherche opérationnelle (CARO)

Équipe de recherche opérationnelle de la Marine (MORT)

R à Québec, 15 mai 2019

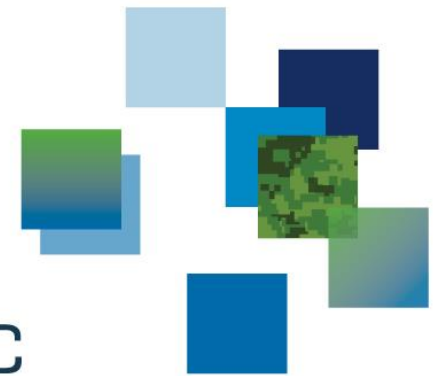
Campus de l'Université Laval




NOTICE (U)

(U) This document has been reviewed and
DOES NOT CONTAIN controlled goods.

DRDC | RDDC



Plan de la présentation

- Qui sommes-nous?
- Le problème
- Objectif du projet
- Le processus 
 - Analyse exploratoire des données
 - Préparation des données
 - Modélisation
- Conclusion

Qui sommes-nous?

- **RDDC**: Recherche et développement pour la défense Canada

- Fournir les connaissances et les technologies nécessaires pour défendre et protéger les intérêts du Canada au pays et à l'étranger



- **RDDC / CARO**: Centre d'analyse et de recherche opérationnelle

- Fournir des analyses de recherche opérationnelle et stratégiques pour supporter la prise de décisions en matière de défense (e.g. Force armée canadienne) et de sécurité

- **RDDC / CARO / MORT**: Équipe de recherche opérationnelle de la Marine

- Support la Marine royale canadienne avec ses besoins en recherche opérationnelle

Le problème

- Marine royale canadienne (MRC)

- 12 frégates de la classe Halifax

- Types de maintenance en vigueur:

- **Préventive**: Maintenance faite à intervalles prédéterminés afin de réduire la chance de défaillance et de dégradation
- **Corrective**: Maintenance exécutée après détection d'une panne

- Futur:

- **Maintenance selon l'état** ("Condition Based Maintenance"): Stratégie qui vise à vérifier l'état actuel de l'équipement et décider quelle maintenance doit être faite en fonction de la dégradation

Avantages: Moins de perturbation sur les opérations, minimise le temps consacré à la maintenance, réduit le coût des défaillances, etc.



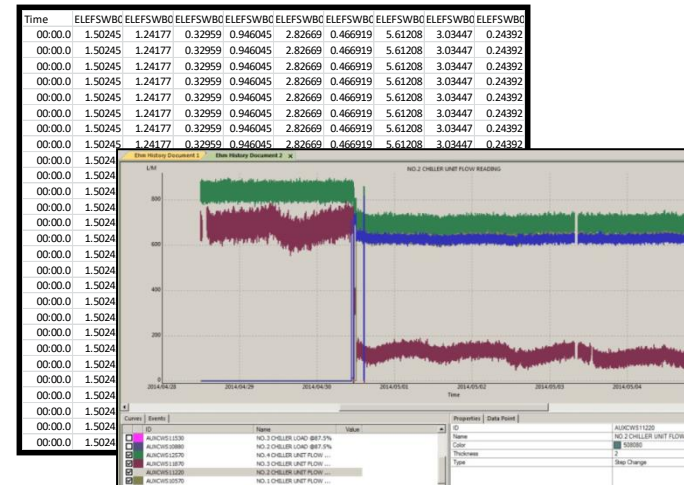
HMCS Halifax

Objectif du projet

- Exploiter les données de senseurs à bord des frégates (i.e. Integrated Platform Management System (IPMS)) pour l'identification de défaillances
 - Est-ce que les données IPMS peuvent être utile? Pouvoir de prédiction?
 - Identification de **quick wins** et de **value cases**

Focus sur 5 systèmes:

1. Propulsion Diesel Engine (PDE)
2. Diesel Generator (DG)
3. Gear Box (GB)
4. Shaft Line (SL)
5. Controllable Reversible-Pitch Propeller (CRPP)

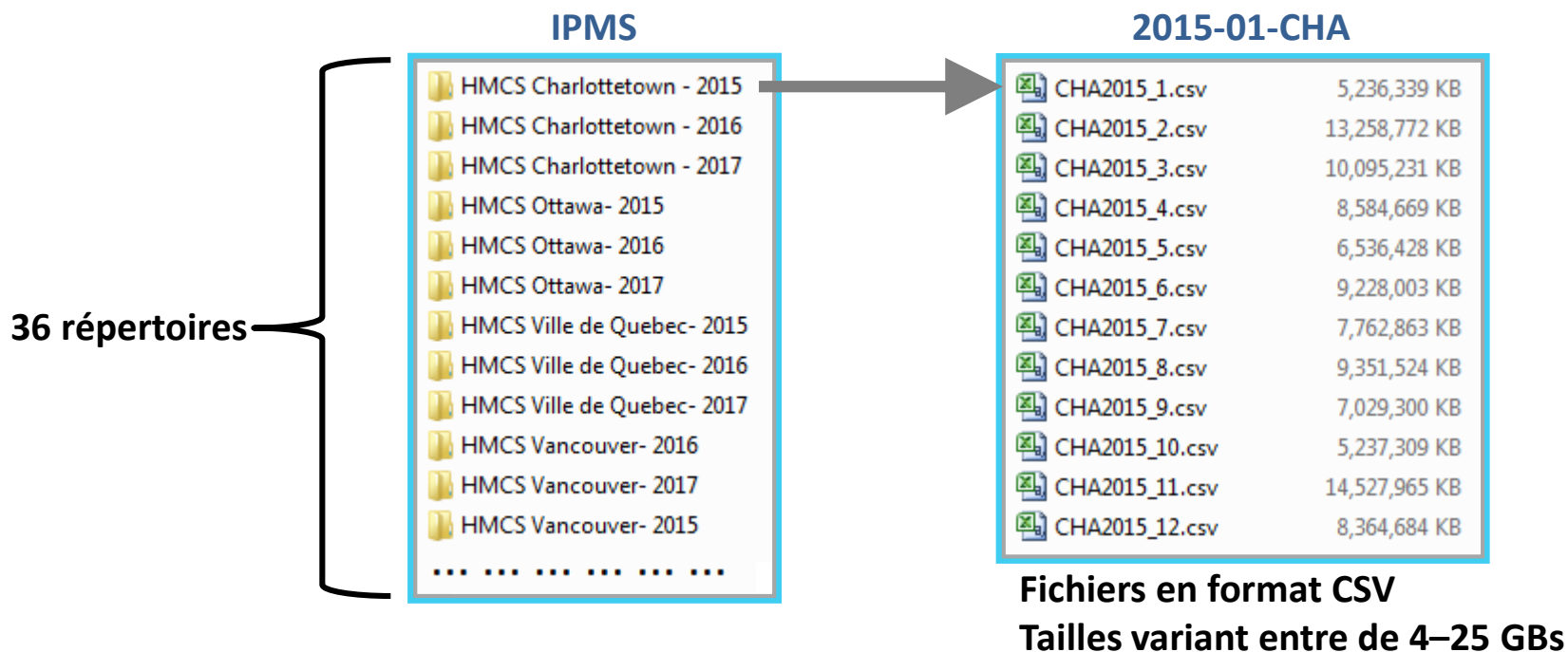


- Utilisation de techniques d'apprentissage supervisé

Analyse exploratoire des données

Aperçu des données obtenues (métadonnées)

- 12 frégates
- 8000+ senseurs capturés à toutes les 0.5 sec, si changement
- De janvier 2015 à décembre 2017





Librairies

- ggplot2

Fonctions

- geom_tile

Analyse exploratoire des données

Aperçu des données obtenues (métadonnées)

- Au total, 2.4 téraoctets (TBs) de données IPMS





Librairies

- ff

Fonctions

- read.csv.ffdf

Analyse exploratoire des données

Aperçu des données

- Problèmes (mémoire et temps) avec **read.csv**

```
> a<-read.csv(file = paste0(pathLocation,"\\CHA2015_1.csv"), header=FALSE)
Error: cannot allocate vector of size 2.0 Gb
Timing stopped at: 529.9 56.58 607.2
```

- Initialement, pour manipuler les fichiers CSV – Utilisation de **read.csv.ffdf**

Champs importants

	Ship	Sys. ID	System Description	Sensor Code	Sensor Description	Time	Unit	Sens. ID	Value	
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
1	CHA	10300	GTS - STBD GT	DPROGTS15200_RH	DPROGTS15200_RH RUNNING HOURS	57:20.0	min	7118000	636317.00000	16
2	CHA	10200	PDE - PROP DIESEL ENG	DPROPDE11550_RH	DPROPDE11550_RH RUNNING HOURS	57:20.0	min	7133000	575523.00000	16
3	CHA	10200	PDE - PROP DIESEL ENG	DPROPDE12000_RH	DPROPDE12000_RH RUNNING HOURS	57:20.0	min	7142000	182972.00000	16
4	CHA	40100	ASC - AUX SALT WATER CIR	AUXASC12800	FER AUX SWC PRESSURE	57:20.6	BARS	1445000	1.33921	16
5	CHA	201800	FWD Switchboard	ELEFSWB0112	FWD SWB Voltage on shore (0 to 600 V) VT5	57:20.6	VOLTS	7680000	441.80701	16
6	CHA	201800	FWD Switchboard	ELEFSWB0114	FWD SWB Voltage on Bars (0 to 600 V) BVT	57:20.6	VOLTS	7682000	445.87399	16
7	CHA	10200	PDE - PROP DIESEL ENG	PROPDE16501	MAIN BEARING TEMP. NO.3	57:20.6	°C	8555000	34.00000	16
8	CHA	10200	PDE - PROP DIESEL ENG	PROPDE21101	EXHAUST GAS TEMP. CYL. A7	57:20.6	°C	8581000	36.00000	16
9	CHA	40500	CWS - CHILLED WATER	AUXCWS14340	PAC 1 CW FLOW	57:21.5	L/M	1624000	185.37500	16
10	CHA	41000	HPA - HIGH PRESS. AIR	AUXHPA11000	AFT MD COMPRESSOR DISCHARGE PRESSURE	57:21.5	BARS	1789000	198.62000	16
11	CHA	201700	AFT Switchboard	ELEASWB0107	AFT SWB Voltage on BT1 (0 to 600 V) VT8	57:21.5	VOLTS	7270000	440.01199	16

Redondances



Préparation des données

Manipulation des données

- Même avec la librairie "ff", beaucoup de défis:

1. Lent à lire un fichier CSV
2. Manipulation un fichier à la fois
3. Incapable que faire des requêtes du type:
 - Records pour une combinaison de bateau / senseur
 - Records pour un système

~18 min pour un fichier CSV 25 GBs

user	system	elapsed
993.57	64.65	1065.07

SOLUTION:

Utilisation de *sparklyr* – Spark "dataframe"



Interface en R à Apache Spark (<http://spark.apache.org>).
Supporte la connection a un "cluster" local ou distant et
est compatible avec la librairie "dplyr".

- ff

- read.csv.ffdf
- write.csv.ffdf

Préparation des données

Réduction des données (avant de créer le Spark "dataframe")

- Élimination de redondance
- Colonnes importants des données IPMS ("Time", "Sensor ID", "Value")

Error in UseMethod("spark_write_parquet") :
no applicable method for 'spark_write_parquet'
applied to an object of class "ffdf"

	A	B	C	D	E	F	G	H	I	J
1	CHA		DCSL222036	JP-5 SUPPLY PUMP OUTLET PORT PRESSURE	08-04-2015 0:00:00.000 BAR	4315100	6.87E-02	16		
2	CHA	40100 ASC - AUX SALT WATER CIR	AUXASC12900	FAMR AUX SWC PRESSURE	08-04-2015 0:00:00.400 BARS	1446000	1.5683399	16		
3	CHA	40100 ASC - AUX SALT WATER CIR	AUXASC13000	AER AUX SWC PRESSURE	08-04-2015 0:00:00.400 BARS	1447000	1.45338	16		
4	CHA	100600 STBD POWER TURBINE	PROGTS10600	COMPRESSOR INLET TEMPERATURE	08-04-2015 0:00:00.400 °C	8397000	13.1685	16		
5	CHA	40100 ASC - AUX SALT WATER CIR	AUXASC12800	FER AUX SWC PRESSURE	08-04-2015 0:00:00.500 BARS	1445000	1.3390599	16		
6	CHA	201700 AFT Switchboard	ELEASWB0107	AFT SWB Voltage on BT1 (0 to 600 V) VT8	08-04-2015 0:00:00.500 VOLTS	7270000	453.767	16		
7	CHA	201700 AFT Switchboard	ELEASWB0114	AFT SWB Voltage on Bars (0 to 600 V) BVT	08-04-2015 0:00:00.500 VOLTS	7277000	450.767	16		
8	CHA	202400 Diesel Generator no. 4	ELEEPG19200	GENERATOR 4 AFT BEARING TEMP	08-04-2015 0:00:00.500 °C	7371000	38.90919	16		
9	CHA	201800 FWD Switchboard	ELEFSWB0107	FWD SWB Voltage on BT2 (0 to 600 V) VT10	08-04-2015 0:00:00.500 VOLTS	7675000	455.1409	16		
10	CHA	201800 FWD Switchboard	ELEFSWB0112	FWD SWB Voltage on shore (0 to 600 V) VTS	08-04-2015 0:00:00.500 VOLTS	7680000	456.3769	16		
11	CHA	201800 FWD Switchboard	ELEFSWB0114	FWD SWB Voltage on Bars (0 to 600 V) BVT	08-04-2015 0:00:00.500 VOLTS	7682000	448.51	16		
12	CHA	202100 Diesel Generator no. 1	ELLCPDG1009	DG #1 Fresh Cooling water Temperature No. 1	08-04-2015 0:00:00.500 °C	7694000	65.5	16		
13	CHA	202400 Diesel Generator no. 4	ELLCPDG4050	DG #4 Cylinder Exhaust Temperature Monitoring - B3	08-04-2015 0:00:00.500 °C	7924000	68	16		
14	CHA	40500 CWS - CHILLED WATER	AUXCWS14380	OPS/CSU CW FLOW	08-04-2015 0:00:01.597 L/M	1626000	17.624701	16		
15	CHA		DCSL217036	JP-5 SUPPLY PUMP INLET STBD PRESSURE	08-04-2015 0:00:01.300 BAR	4314800	2.08E-02	16		
16	CHA		DCSL221036	JP-5 SUPPLY PUMP INLET PORT PRESSURE	08-04-2015 0:00:01.300 BAR	4315000	0.0147	16		
17	CHA	201700 AFT Switchboard	ELEASWB0107	AFT SWB Voltage on BT1 (0 to 600 V) VT8	08-04-2015 0:00:01.497 VOLTS	7270000	453.745	16		
18	CHA	201700 AFT Switchboard	ELEASWB0114	AFT SWB Voltage on Bars (0 to 600 V) BVT	08-04-2015 0:00:01.497 VOLTS	7277000	450.608	16		
19	CHA	201800 FWD Switchboard	ELEFSWB0107	FWD SWB Voltage on BT2 (0 to 600 V) VT10	08-04-2015 0:00:01.497 VOLTS	7675000	455.164	16		
20	CHA	201800 FWD Switchboard	ELEFSWB0110	FWD SWB Current on shore (0 to 2000 A) ITS	08-04-2015 0:00:01.497 AMPS	7678000	880.492	16		
21	CHA	201800 FWD Switchboard	ELEFSWB0113	FWD SWB frequency on shore (55 to 65 Hz) FTS	08-04-2015 0:00:01.497 Hz	7681000	59.951302	16		

Fichiers CSV originaux de 4–25 GBs

Fichiers CSV plus petits (~100 MBs)

	A	B	C
1	04/Jan/15 20:08:50.5	7682000	451.987
2	04/Jan/15 20:08:50.5	7744000	57
3	04/Jan/15 20:08:51.4	4315100	-0.02302
4	04/Jan/15 20:08:51.5	1626000	14.0384
5	04/Jan/15 20:08:51.5	4315000	-6.72E-03
6	04/Jan/15 20:08:51.5	7277000	454.12399
7	04/Jan/15 20:08:51.5	7335000	50.560501
8	04/Jan/15 20:08:51.5	7675000	449.991
9	04/Jan/15 20:08:51.5	7678000	744.60199
10	04/Jan/15 20:08:51.5	7680000	450.884
11	04/Jan/15 20:08:51.5	7682000	452.37701
12	04/Jan/15 20:08:51.5	8575000	51
13	04/Jan/15 20:08:52.4	8979400	158.774
14	04/Jan/15 20:08:52.5	7270000	448.457
15	04/Jan/15 20:08:52.5	7277000	453.61801
16	04/Jan/15 20:08:52.5	7334000	51.810902
17	04/Jan/15 20:08:52.5	7363000	54.658401
18	04/Jan/15 20:08:52.5	7675000	449.487
19	04/Jan/15 20:08:52.5	7678000	754.48401
20	04/Jan/15 20:08:52.5	7679000	158.774
21	04/Jan/15 20:08:52.5	7680000	450.56201
22	04/Jan/15 20:08:52.5	7682000	451.87201
23	04/Jan/15 20:08:52.5	7978000	158.774
24	04/Jan/15 20:08:53.4	4315100	-3.83E-02
25	04/Jan/15 20:08:53.4	7270000	448.41199
26	04/Jan/15 20:08:53.4	7277000	453.50299
27	04/Jan/15 20:08:53.4	7678000	752.10901
28	04/Jan/15 20:08:53.4	7680000	450.608
29	04/Jan/15 20:08:53.4	7682000	452.056
30	04/Jan/15 20:08:53.5	1626000	14.4169



Librairies

- networkd3

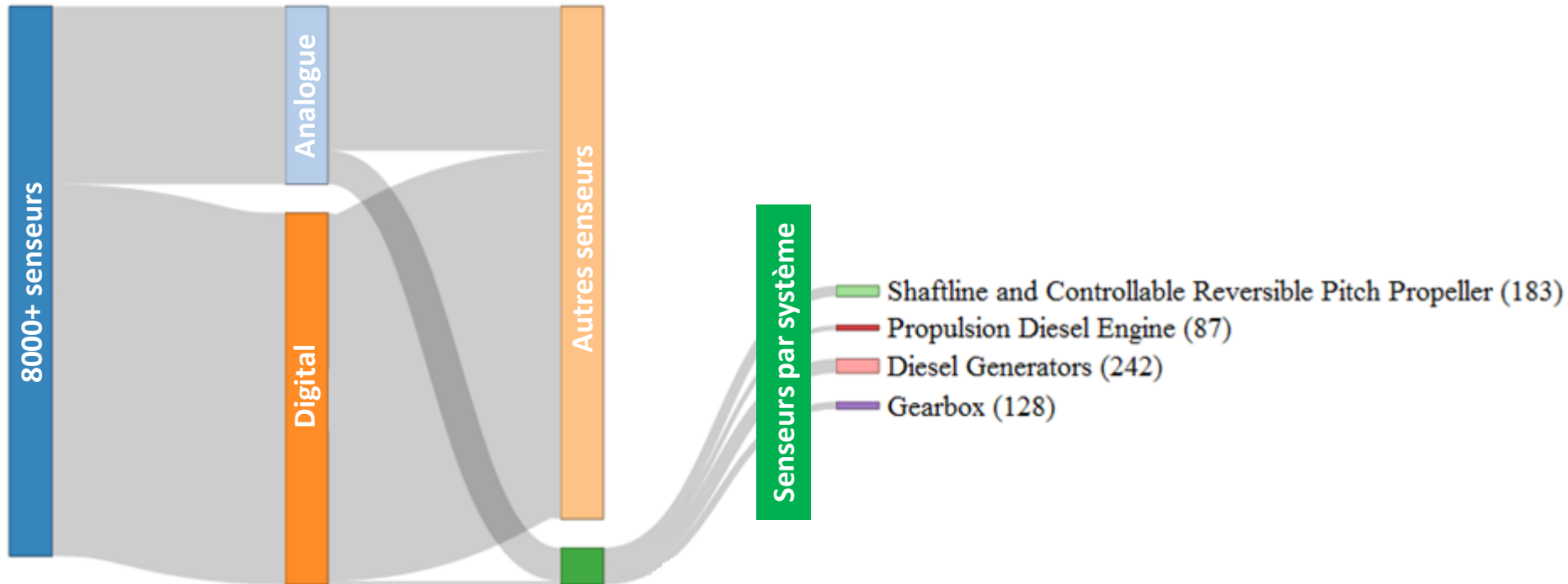
Fonctions

- sankeyNetwork

Préparation des données

Réduction des données (avant de créer le Spark "dataframe")

- Identification, par experts, des senseurs d'intérêts par système



Analogue: Caractéristique continue (e.g. température, pression)

Digitale: Valeur discrète (e.g. 0 et 1's, "On" et "Off")

Préparation des données

Création des Spark "dataframe" (un par bateau)

- Pour chaque fichier CSV d'un bateau
- Extraire les senseurs d'intérêts
- Sauver en format "parquet"

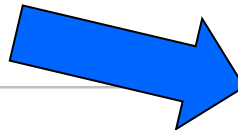
Librairies

- sparklyr
- dplyr
- tibble

Fonctions

- spark_connect
- spark_write_parquet
- spark_disconnect
- read.csv

```
1 library(sparklyr)
2 library(dplyr)
3 devtools::install_github("rstudio/sparklyr")
4 # ...
5
6 sc <- spark_connect(master = "local", app_name = "sparklyr")
7 # ...
8 SensorList <- ReadSensorOfInterest ()
9 # ...
10 # LOOP for each files of "Ship"
11 temp <- read.csv(name=csvPath)
12 # ...
13 temp <- temp %>% select(SensorCode, Time, value) %>% filter(SensorCode %
14 # ...
15 system.time(spark_write_parquet(x=temp, name=Ship, path=SparkDFLoc, mod
16 # ...
17 # END LOOP
18 spark_disconnect(sc)
19
```



._SUCCESS.crc	1 KB	CRC File
.part-00000-3dc90c40-6f1d-4307-ace9-2056c45b799b-c000.snappy.parquet.crc	54 KB	CRC File
.part-00000-8fd50135-1278-405e-a7b9-af5b1c4fa1d2-c000.snappy.parquet.crc	49 KB	CRC File
.part-00000-294972e1-1554-4505-ab11-e3b76b958e42-c000.snappy.parquet.crc	32 KB	CRC File
.part-00000-b9d358d6-ec92-49e8-8b4b-9c62f532a7b2-c000.snappy.parquet.crc	75 KB	CRC File
.part-00000-d40a7cbd-d440-4728-98d8-25ce59702668-c000.snappy.parquet.crc	60 KB	CRC File
.part-00000-d251853a-eb29-4026-a1e7-e42eac7057b7-c000.snappy.parquet.crc	37 KB	CRC File
.part-00000-d7474764-2130-4c15-aad2-4692e2d0cebe-c000.snappy.parquet.crc	46 KB	CRC File
.part-00001-3dc90c40-6f1d-4307-ace9-2056c45b799b-c000.snappy.parquet.crc	55 KB	CRC File
.part-00001-8fd50135-1278-405e-a7b9-af5b1c4fa1d2-c000.snappy.parquet.crc	64 KB	CRC File
.part-00001-294972e1-1554-4505-ab11-e3b76b958e42-c000.snappy.parquet.crc	57 KB	CRC File
.part-00001-b9d358d6-ec92-49e8-8b4b-9c62f532a7b2-c000.snappy.parquet.crc	72 KB	CRC File
.part-00001-d40a7cbd-d440-4728-98d8-25ce59702668-c000.snappy.parquet.crc	75 KB	CRC File
.part-00001-d251853a-eb29-4026-a1e7-e42eac7057b7-c000.snappy.parquet.crc	65 KB	CRC File
.part-00001-d7474764-2130-4c15-aad2-4692e2d0cebe-c000.snappy.parquet.crc	89 KB	CRC File
.part-00002-3dc90c40-6f1d-4307-ace9-2056c45b799b-c000.snappy.parquet.crc	44 KB	CRC File
.part-00002-8fd50135-1278-405e-a7b9-af5b1c4fa1d2-c000.snappy.parquet.crc	70 KB	CRC File
.part-00002-294972e1-1554-4505-ab11-e3b76b958e42-c000.snappy.parquet.crc	56 KB	CRC File
.part-00002-b9d358d6-ec92-49e8-8b4b-9c62f532a7b2-c000.snappy.parquet.crc	68 KB	CRC File
.part-00002-d40a7cbd-d440-4728-98d8-25ce59702668-c000.snappy.parquet.crc	81 KB	CRC File

Les fonctions de "dplyr" peuvent être utilisées sur les Spark "Dataframe" (select, filter, arrange, mutate, summarise, etc.)



Librairies

- dplyr
- ggplot2

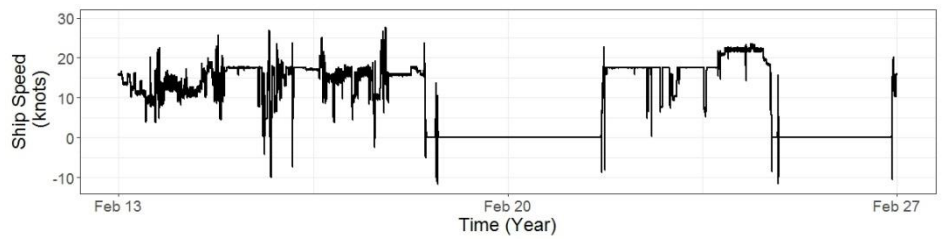
Fonctions

- summarise
- filter
- group_by

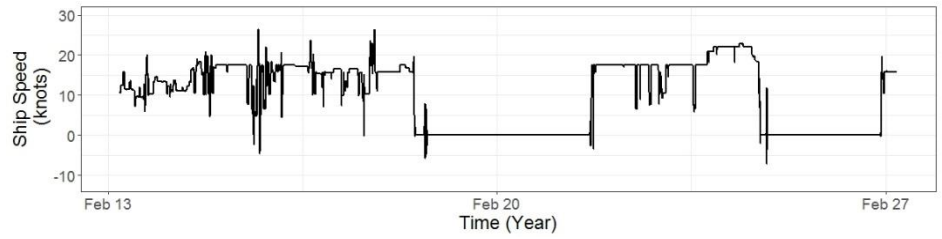
Préparation des données

Création des séries temporelles régulières par senseurs

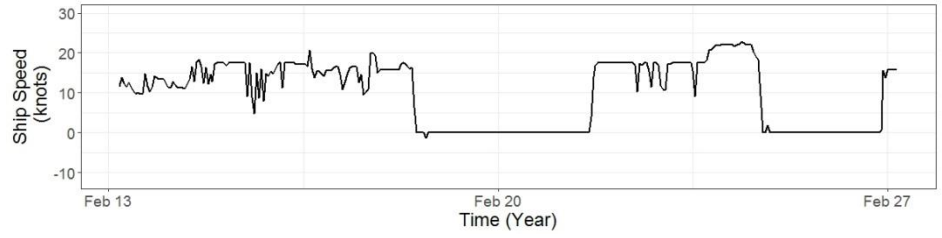
Données brutes
(172899 points)



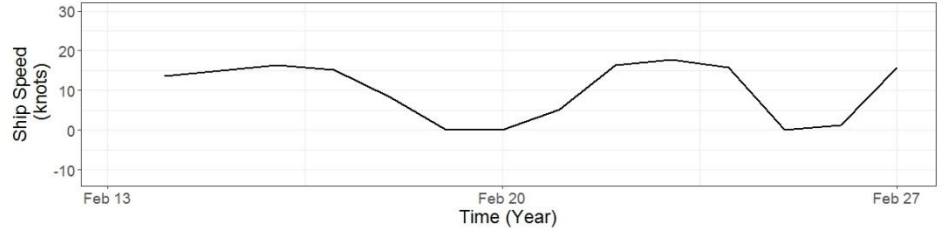
Au 5 min
(4031 points)



À l'heure
(336 points)



À la journée
(14 points)



Nettoyage

- Enlever les "NA"
- Remplir les "gaps" en utilisant *last observed value carried over*

Modèles créés un système à la fois, donc, création d'un fichier ".Rda" pour chaque senseur

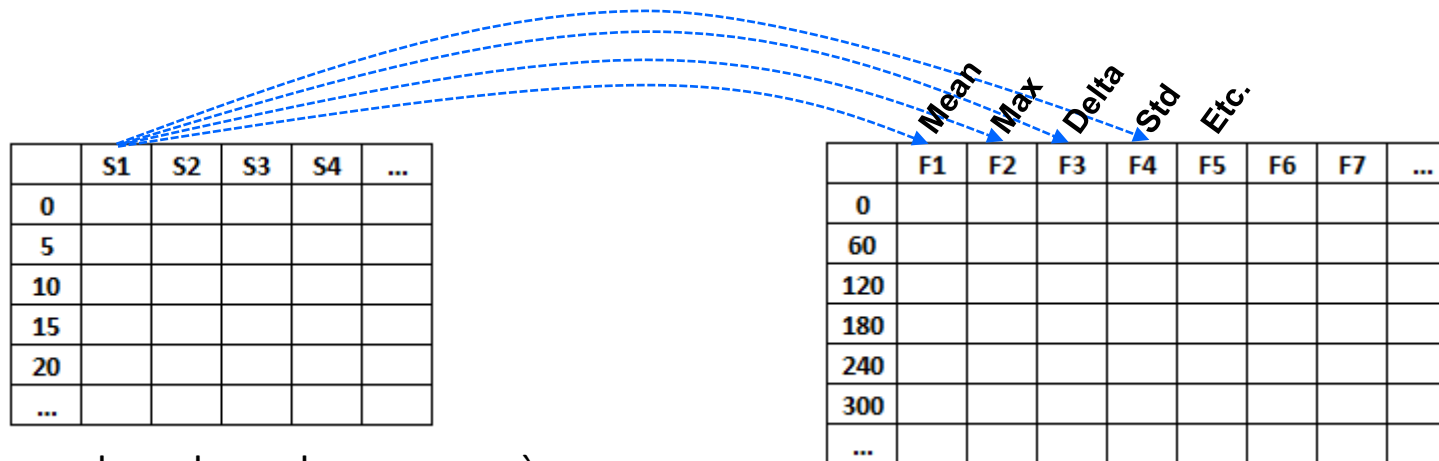
Name	Size
8641000-5min.Rda	3,976 KB
8639000-5min.Rda	4,051 KB
8637000-5min.Rda	3,985 KB
8635000-5min.Rda	4,062 KB
8633000-5min.Rda	3,985 KB
8631000-5min.Rda	4,062 KB
8629000-5min.Rda	3,976 KB
8627000-5min.Rda	4,051 KB
...	...

- sparklyr
- dplyr

- Mean
- Max
- Etc.

Préparation des données

Création et sélection de variables (i.e. "input features")



Nous avons les valeurs des capteurs à la fin de chaque intervalle de 5 min

Ce que nous voulons, ce sont des caractéristiques des capteurs dans des intervalles de temps fixes (e.g. 1 heure)

Un grand nombre de variables initiales avec beaucoup de corrélation, e.g.

- Températures et vitesse du bateau

- stats
- ggplot2

- prcomp

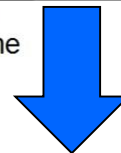
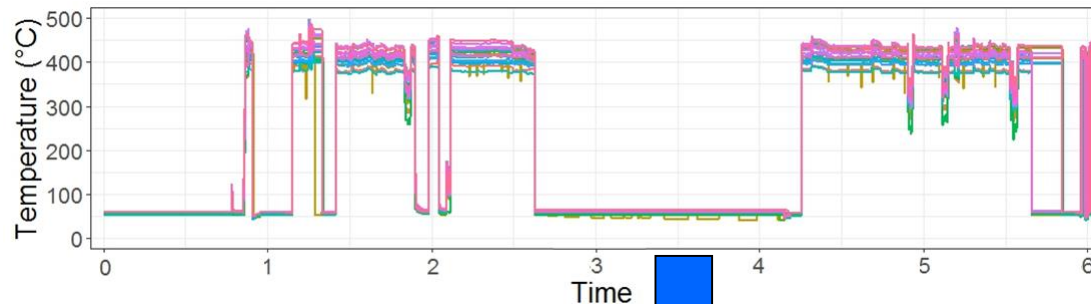
Préparation des données

Création et sélection de variables (i.e. "input features")

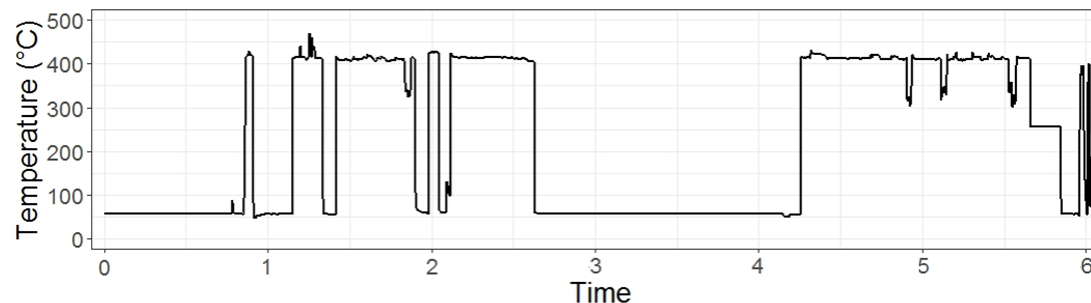
- Réduire dimensionnalité (e.g. PCA ("Principal Component Analysis"))
- Définition de variables plus représentatives / Regroupement

Regroupement de senseurs

PDE Températures d'échappement (20 senseurs)



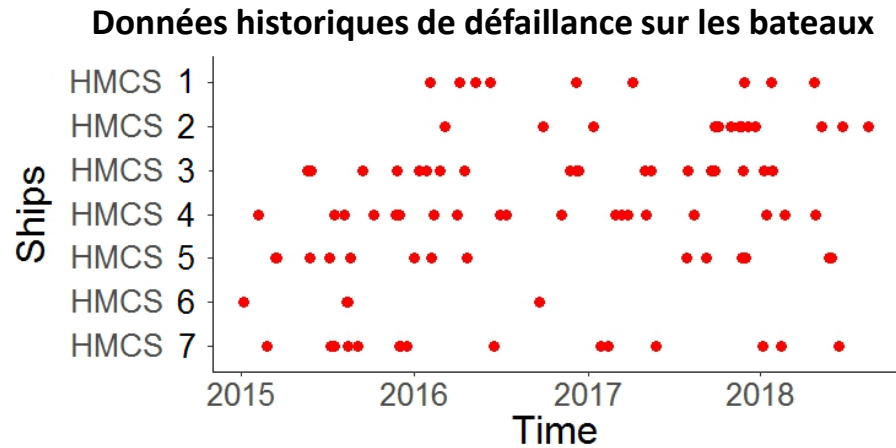
1 Variable – Moyenne des températures d'échappement



Modélisation

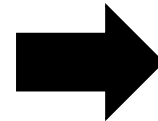
Développement de modèles basés avec apprentissage supervisé

Variables créées à partir des données IPMS



	F1	F2	F3	F4	F5	F6	F7	...
0								
60								
120								
180								
240								
300								
...								

Défaillance
Non
Non
Non
Non
Oui
Non



Données pour l'entraînement et la testing des modèles

Modélisation

Développement de modèles basés sur apprentissage supervisé

- Travail en cours ...
- Librairie "RandomForest"

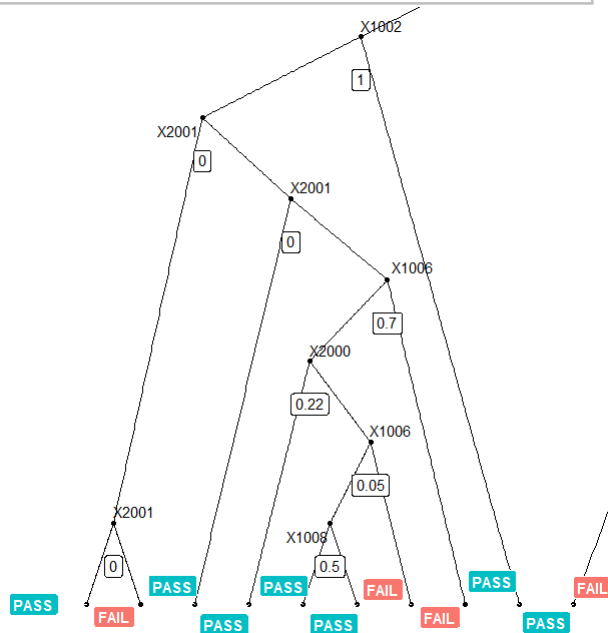
Librairies

- RandomForest

Fonctions

- randomForest
- predict

```
33 library(randomForest)
34 # ...
35 # ...
36 df_rF <- randomForest(formula=PassFail~., data=trainData, ntree=200,
37                       replace=TRUE, cutoff=c(0.5, 0.5),
38                       strata=trainData$PassFail, importance=FALSE)
39 # ...
40 dfPred <- predict(df_rF, newdata=testData)
41 # ...
42
```



MATRICE DE CONFUSION

		<u>PRÉDIT</u>		
		FAIL	PASS	class_error
OBSERVÉ	FAIL	5	1	0.1666667
	PASS	83	433	0.1608527
accuracy		0.8390805		

Conclusion

- 2.4 TBs de données IPMS
- Développement de modèles prévisionnels en R pour déterminer le pouvoir de prédiction
- Beaucoup de défis rencontrés
 - Storage / Mémoire
 - Rapidité
- Travaux futurs
 - Sur un "cloud"
 - Machines virtuelles, Microsoft Azure
 - Apprentissage non supervisé



Defence Research and
Development Canada

Recherche et développement
pour la défense Canada

DRDC | RDDC

SCIENCE, TECHNOLOGY AND KNOWLEDGE
FOR CANADA'S DEFENCE AND SECURITY

SCIENCE, TECHNOLOGIE ET SAVOIR
POUR LA DÉFENSE ET LA SÉCURITÉ DU CANADA

