

## 1. Project Introduction

### 1.1 Background

### 1.2 Planification

## 2. Work Conducted

### 2.1 Data preprocessing

#### 2.1.1 Data understanding

#### 2.1.2 Look into data by `cell_id`

(1) Features with too many missing values

(2) Highly relevant features

#### 2.1.3 Fill NaN & Add new features

### 2.2 Modeling

#### 2.2.1 Isolation Forest

(1) Introduction

(2) Implementation

#### 2.2.2 Variational Auto-Encoder (VAE)

(1) Introduction

(2) Implementation

#### 2.2.3 Combination of *IF* and *VAE*

(1) Stacking

(2) Bagging

### 2.3 Result Evaluation

#### 2.3.1 Analysis on metrics obtained

(1) Distribution

(2) Interrelation

(3) Visualize metrics on original data

#### 2.3.2 Identification of anomalies

IQR method

(1) Result - Stacking Model

(2) Result - Bagging Model

#### 2.3.3 Conclusion

## 3. Summary & Reflection

### 3.1 Problems encountered

#### 3.1.1 Handling large number of missing values

#### 3.1.2 Evaluation of model output

### 3.2 Knowledge acquired

#### 3.2.1 Theoretical aspect

#### 3.2.2 Practical aspect

### 3.3 Possible improvements

# 1. Project Introduction

---

## 1.1 Background

---

Telecom networks are generating huge amounts of data coming from traffic on this network due to customer calls and services consumptions. This data contains a lot of valuable information that enable machine learning models to learn from and predict outcomes to maintain service quality.

The main purpose of the proposed project is detecting anomalies and predicting incident/failures on the network in real time.

In order to achieve the target, work is to be performed on comparing and validating the results of three approaches: 1) Auto-Encoder (Deep learning - Neural network approach) 2) Isolation Forest and 3) the combination of the two previous approaches. Finally, a conclusion about the optimal approach needs to be obtained.

This project is proposed by [B-Yond](#) and Ecole Mines Saint-Etienne, and will be under the joint guidance of the corporate mentor Mr. Michel KAMEL and the school mentor Mr. Anis HOAYEK.

## 1.2 Planification

---

The project starts on October 1, 2021 and ends on January 27, 2022, and each week with 1~2 days of work on it is required.

The project is divided into three main parts, the details and specifics/methodology are shown in the table below.

	Theoretical phase	Practical phase	Summarization phase
Objective	Understanding the theoretical background of the algorithm to be used	Data processing, model building, model performance analysis	Summarize the work conducted, further improve/optimize the work if possible
Content	Read the papers on <i>Isolation Forest</i> and <i>(Variational) Auto-Encoder</i>	1. Understand the data and complete the processing of the data (missing value processing, feature engineering..) 2. Build the three required models and get the corresponding indicators for evaluating sample anomalies 3. Analyze the results of the three models and further propose methods to detect anomalies	1. Discuss the project results with the mentors 2. Write project report
Date	2021.10.01-2021.10.24	2021.10.25 - 2022.12.31	2022.1.1-2022.1.23

## 2. Work Conducted

### 2.1 Data preprocessing

#### 2.1.1 Data understanding

In this project, we begin with a small sample dataset, which has only 8280 records.

	index	cell_id	DL_TRAFFIC_VOLUME	UL_TRAFFIC_VOLUME	Inter_X2_based_HO_prep	PDCP_SDU_Volume_DL	VoLTE_total_traffic
0	2021-05-09 00:00:00	2.2265366483183206e+17	3.779737e+10	3.947172e+09	15.0	3.779737e+10	4727.0
1	2021-05-09 01:00:00	2.2265366483183206e+17	3.684898e+10	4.088752e+09	6.0	3.684898e+10	3076.0
2	2021-05-09 02:00:00	2.2265366483183206e+17	3.292677e+10	5.016897e+09	8.0	3.292677e+10	3501.0
3	2021-05-09 03:00:00	2.2265366483183206e+17	3.021547e+10	5.139107e+09	9.0	3.021547e+10	2275.0
4	2021-05-09 04:00:00	2.2265366483183206e+17	3.082176e+10	4.250716e+09	17.0	3.082176e+10	2178.0

In this data 26 features are measured, including time, cell id and some KPIs of LTE (Long Term Evolution, a wireless data communication technology standard).

These data come from 5 cells, that is, each cell contains 1656 records.

**cell** : In the field of mobile communication, the area covered by wireless signals is called a *cell*, which generally refers to the area that can be covered by the signal of a base station.

First Let's check the null values in our dataset :

	FEATURES	Num_nu1l
1	-----	-----
2		
3	index	0
4	cell_id	0
5	DL_TRAFFIC_VOLUME	21
6	UL_TRAFFIC_VOLUME	21
7	Inter_X2_based_HO_prep	3313
8	PDCP_SDU_Volume_DL	3313
9	VoLTE_total_traffic	3313
10	INTRA_FREQ_HO_SR_RATIO	40
11	RRC_SR_RATIO	28
12	Intra_eNB_HO_SR_total_RATIO	3321
13	E_UTRAN_RRC_Conn_Stp_Failure_due_RRC_timer_expiry_RATIO	3313
14	CELL_AVAILABILITY_RATIO	20
15	RACH_Stp_Completion_SR_RATIO	3313
16	Total_E_UTRAN_RRC_Conn_Stp_SR_RATIO	3313
17	Inter_RAT_HO_SR_UTRAN_SRVCC_RATIO	5595
18	UL_THROUGHPUT_RATIO	28
19	E_RAB_QC11_DR_RATIO	3324
20	DCR_LTE_RATIO	28
21	CSSR_LTE_RATIO	28
22	LTE_INTER_ENODEB_HOSR_RATIO	1695
23	E_UTRAN_Inter_Freq_HO_SR_RATIO	4967
24	Inter_RAT_HO_SR_GERAN_SRVCC_RATIO	3989
25	Inter_RAT_Total_HO_SR_RATIO	3731
26	E_UTRAN_tot_HO_SR_inter_eNB_X2_RATIO	3347
27	DL_THROUGHPUT_RATIO	28
28	E_RAB_DR_RATIO	3314

In the data, many variables have a large number of missing values! Usually, for missing values, we will either fill them using other means or delete features/records with many missing values. In our case, it can be seen that there is perhaps some correlation between these missing values (e.g., many variables have 3313 missing values). We will start with one point and then move on to consider how to deal with them.

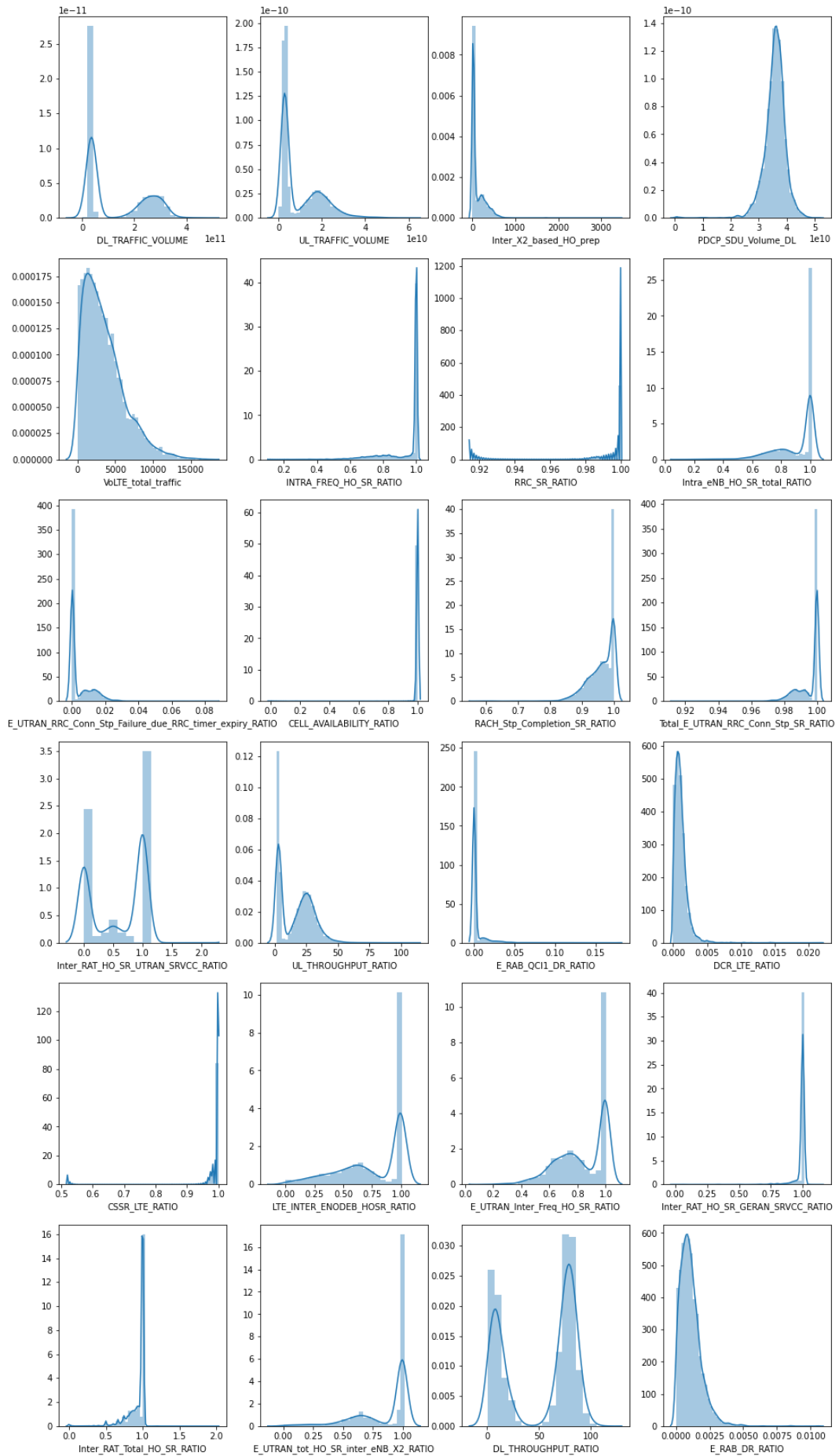
After a brief glance at the data file in Excel, we found this:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
3821	03/06/2021 7:00	0.115-30	276871395	3659174440.0				0.97560970	0.999132041	119375		1.0				2.519 596 178 620 730		0.0089950	0.9724541	0.5949540494949494					7.011 633 365 613 380			
3822	03/06/2021 8:00	0.115-30	34138435	146697240.0				1.0	0.9985720948009			1.0				21 790 431 541 708 900		0.0018994	0.9778281	0.6136036303693606					6.351 212 348 211 070			
3823	03/06/2021 9:00	0.115-30	252007841	4338315204.0				1.0	0.9988151872210178			1.0				20 906 655 587 751 100		0.0018890	0.9778380	0.60496403080719					7.494 692 872 25 000			
3824	03/06/2021 10:00	0.115-30	266897995	18112169920.0				1.0	0.9990238478769389			1.0				23 322 285 050 713 300		0.0020590	0.97888396	0.5714285714285714					7.340 607 463 013 650			
3825	03/06/2021 11:00	0.115-30	267778169	158811584624.0				1.0	0.99917738511584625			1.0				530 707 700 396 862		0.0017703	0.97930770	0.593483744688875					5.730 266 036 032 700			
3826	03/06/2021 12:00	0.115-30	288937810	21144848440.0				1.0	0.9992068717607402			1.0				2 452 388 398 886 760		0.00181510	0.97486080	0.5					6.695 264 653 709 940			
3827	03/06/2021 13:00	0.115-30	286978169	1777699320.0				1.0	0.9994995490404315			1.0				2 840 735 440 923 040		0.00145940	0.98395900	0.48					544 174 615 148 478			
3828	03/06/2021 14:00	0.115-30	286048196	18617734720.0				1.0	0.9987562188904728			1.0				20 883 614 039 637 500		0.00134000	0.97789000	0.2					61 688 761 461 120 700			
3829	03/06/2021 15:00	0.115-30	279793115	4133028480.0				1.0	0.99937884841222			1.0				1 474 050 050 624 280		0.0004842	0.9713180	0.61379303944487598					5 009 594 293 086 430			
3830	03/06/2021 16:00	0.115-30	245884802	27410835440.0				1.0	0.9994058570781880			1.0				1 885 935 905 198 172		0.0038860	0.97763800	0.5					5 009 594 293 086 430			
3831	03/06/2021 17:00	0.115-30	230903131	1393148720.0				1.0	0.99894405827049			1.0				18 777 000 900 807 600		0.0012248	0.9628701	0.584905669377558					7 052 145 567 043 910			
3832	03/06/2021 18:00	0.115-30	238440082	1404820320.0				1.0	0.9991300048027072			1.0				21 846 846 581 637 200		0.0019990	0.98088502	0.5777777777777777					8 144 335 492 296 490			
3833	03/06/2021 19:00	0.115-30	232193959	18464930920.0				1.0	0.99876965307062			1.0				21 454 025 246 445 600		0.0022200	0.9855950	0.7329411747470899					4 268 943 190 587 500			
3834	03/06/2021 20:00	0.115-30	238623438	15841584120.0				1.0	0.9984053148205715			1.0				19 637 057 540 603 600		0.0037771	0.97574240	0.64705826350418					4 094 724 625 742 600			
3835	03/06/2021 21:00	0.115-30	285481842	15550597240.0				1.0	0.9994516498395827			1.0				1 940 995 126 144 320		0.0010000	0.9715590	0.620696551724138					18 033 334 334 275 100			
3836	03/06/2021 22:00	0.115-30	274865161	2333899904.0				1.0	0.9991421201819757			1.0				19 071 180 444 375 200		0.0026464	0.9787800	0.645164290325066					4 095 963 894 407 470			
3837	03/06/2021 23:00	0.115-30	282557720	1979750096.0				1.0	0.9995815400120706			1.0				19 616 702 788 417 000		0.0012600	0.97567140	0.743857142857143					31 979 398 309 314 300			
3838	04/06/2021 0:00	0.115-30	27387311	10331748784.0				1.0	0.99986385086619			1.0				20 347 884 573 014 000		0.0026202	0.9804800	0.56					2 462 663 373 275 800			
3839	04/06/2021 1:00	0.115-30	27072048	12495936480.0				1.0	0.9991762513064418			1.0				2 461 841 561 160 070		0.0026202	0.9807280	0.584615841538464					4 260 741 742 799 600			
3840	04/06/2021 2:00	0.115-30	23224442	1830240216.0				1.0	0.9998511240310986			1.0				24 745 851 614 880 100		0.0042200	0.9835420	0.574285714285714					27 461 439 030 184 800			
3841	04/06/2021 3:00	0.115-30	28464197	11641116400.0				1.0	0.999804786782235			1.0				33 320 058 890 471 000		0.0022400	0.9806000	0.5708442102635178					462 273 447 446 948			
3842	04/06/2021 4:00	0.115-30	14931053	17897588896.0				0.94117840	0.990734315839599			1.0				28 672 566 351 039 500		0.0022148	0.9802740	0.4					65 028 562 194 038 600			
3843	04/06/2021 5:00	0.115-30	284139096	9230995904.0				1.0	0.9991743155256863			1.0				20 376 953 665 557 800		0.0020290	0.9879540	0.5					9 931 480 695 570 800			
3844	04/06/2021 6:00	0.115-30	21821751	17060498944.0				1.0	0.999434060432787			1.0				28 838 571 173 700 000		0.0099270	0.9863709	0.55294117647058826					9 527 100 536 957 300			
3845	04/06/2021 7:00	0.115-30	21831376	25506367360.0				1.0	0.9989353711702939			1.0				2 943 020 952 251 940		0.0024400	0.9840371	0.5929292929292929					8 960 368 188 149 000			
3846	04/06/2021 8:00	0.115-30	25025506	1670278576.0				1.0	0.999617767057853			1.0				28 885 866 797 662 700		0.0018160	0.9804000	0.4375					8 960 368 188 149 000			
3847	04/06/2021 9:00	0.115-30	250387556	14905232240.0				1.0	0.999744607380988			1.0				28 307 540 541 111 100		0.0003144	0.9745395	0.5842120261578946					8 984 813 417 050 400			
3848	04/06/2021 10:00	0.115-30	24566077	13836498880.0				1.0	0.9996811239787824			1.0				2 392 151 095 997 800		0.0006014	0.9802240	0.5382714285714286					8 713 156 588 680 800			
3849	04/06/2021 11:00	0.115-30	24697320	17174643120.0				1.0	0.999725606369362			1.0				2 134 140 602 901 840		0.0008294	0.9805612	0.23529411764705882					8 180 154 562 344 000			
3850	04/06/2021 12:00	0.115-30	24692240	17176464480.0				1.0	0.99914250441706			1.0				14 815 415 411 148 000		0.0012911	0.98411300	0.5333333333333333					5 688 721 864 516 400			
3851	04/06/2021 13:00	0.115-30	238033872	11399507520.0				1.0	0.9979204460439062			1.0				2 127 145 595 828 110		0.0016070	0.9888300	0.8111111111111111					4 131 953 866 124 600			
3852	04/06/2021 14:00	0.115-30	24692240	17176464480.0				1.0	0.9991234649414178			1.0				16 440 821 462 858 400		0.0014261	0.9820795	0.2681533333333333					20 222 446 304 189 100			
3853	04/06/2021 15:00	0.115-30	30140174	12842158464.0				1.0	0.9996158614594948			1.0				20 028 358 813 519 800		0.0007130	0.9818355	0.46335841584158416					39 875 735 278 078 000			
3854	04/06/2021 16:00	0.115-30	256698752	20994334720.0				1.0	0.998171846451005			1.0				19 980 538 355 270 000		0.0017390	0.9732144	0.580604615095225					3 021 345 109 768 760			
3855	04/06/2021 17:00	0.115-30	24589977	9794168920.0				1.0	0.998405401691133			1.0				2 496 552 998 941 420		0.0013010	0.97788840	0.75					3 264 861 514 648 400			
3856	04/06/2021 18:00	0.115-30	207373878	105613912720.0				1.0	0.999088988750862			1.0				16 611 26 527 611 700		0.0019400	0.9742900	0.4230769230769231					2 727 166 385 578 900			
3857	04/06/2021 19:00	0.115-30	267778169	15273774986.0				1.0	0.997737835338096			1.0				18 240 846 462 452 000		0.0018948	0.9802690	0.407142857142857					259 012 641 169 447			
3858	04/06/2021 20:00	0.115-30	24593897	15273774986.0				1.0	0.997737835338096			1.0				18 240 846 462 452 000		0.0018948	0.9802690	0.407142857142857					259 012 641 169 447			
3859	04/06/2021 21:00	0.115-30	24593897	15273774986.0				1.0	0.997737835338096			1.0				18 240 846 462 452 000		0.0018948	0.9802690	0.407142857142857					259 012 641 169 447			
3860	04/06/2021 22:00	0.115-30	24593897	15273774986.0				1.0	0.997737835338096			1.0				18 240 846 462 452 000		0.0018948	0.9802690	0.407142857142857					259 012 641 169 447			
3861	04/06/2021 23:00	0.115-30	24593897	15273774986.0				1.0	0.997737835338096			1.0				18 240 846 462 452 000		0.0018948	0.9802690	0.407142857142857					259 012 641 169 447			
3862	04/06/2021 0:00	0.115-30	24593897	15273774986.0				1.0	0.997737835338096			1.0				18 240 846 462 452 000		0.0018948	0.9802690	0.407142857142857					259 012 641 169 447			
3863	04/06/2021 1:00	0.115-30	24593897	15273774986.0				1.0	0.997737835338096			1.0				18 240 846 462 452 000		0.0018948	0.9802690	0.407142857142857					259 012 641 169 447			
3864	04/06/2021 2:00	0.115-30	24593897	15273774986.0				1.0	0.997737835338096			1.0				18												

Apparently, these missing values are concentrated in a certain region. More specifically, for this cell, it seems that it does not contain certain KPIs. In the next step, we can consider processing the data of different cells separately.

Viewing data in Excel is a very simple way (so simple that it is sometimes overlooked) to provide some information in a very visual way

Then we'll look into the data distribution :



It can be seen that the data corresponding to the features are all continuous. For some of the features related to "ratio", the data set is distributed around 1 or 0 (in the above figure it looks like a binary distribution, but it has actually continuous values).

Therefore, we can later use some missing value padding methods for continuous numeric variables.

### 2.1.2 Look into data by `cell_id`

## (1) Features with too many missing values

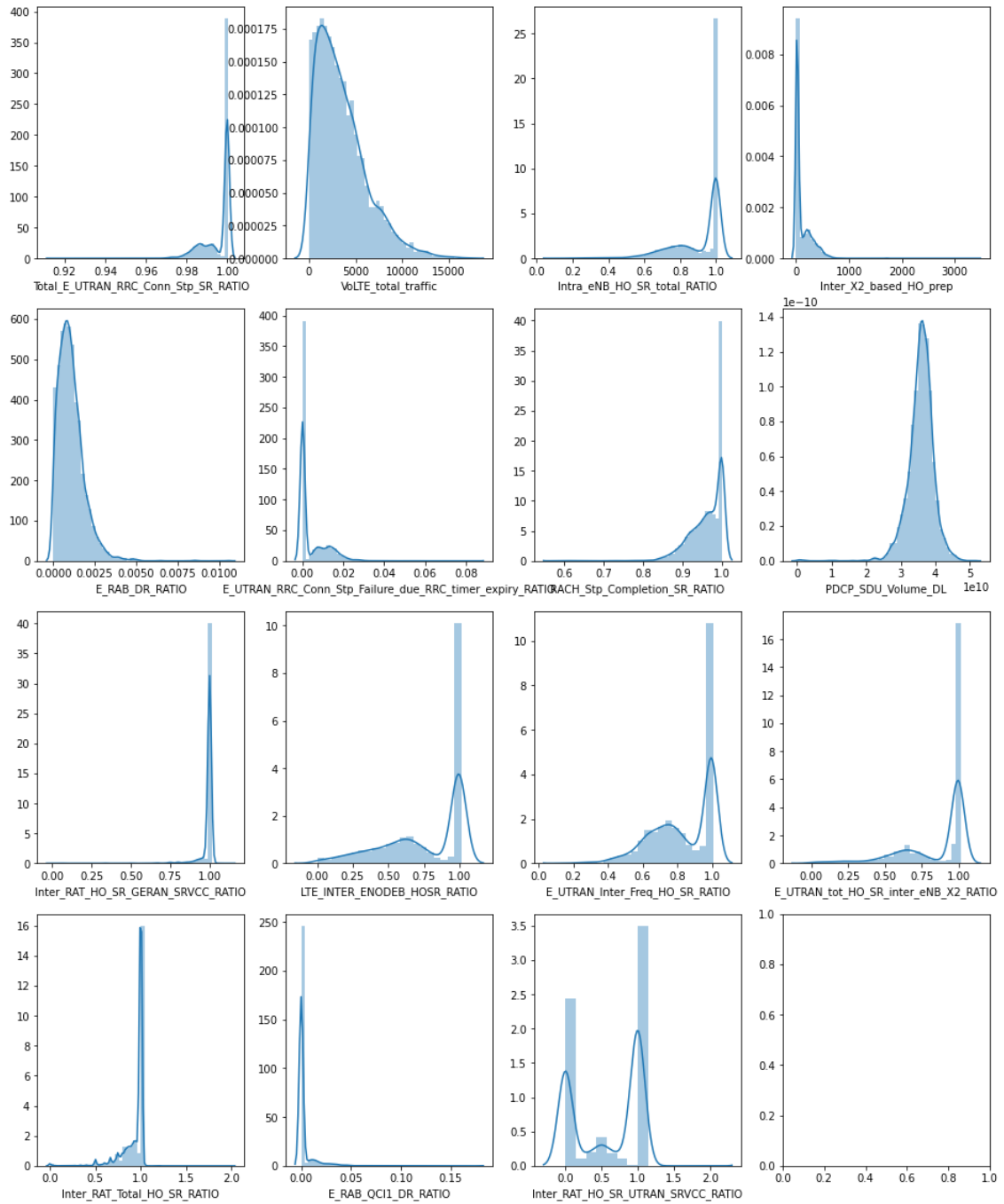
Based on the above discussion, we looked at the missing values in each cell data

[illegible]

It is obvious that for the third and fourth cell in the above figure, they have much more missing values on certain features. To see it more clearly, we can print out only those who do not have too many NaN values (less than 30% for example).

[illegible]

Now, we will look into the distribution of those columns whose NaN values need to be filled later.



For different distributions, we can propose the following strategies:

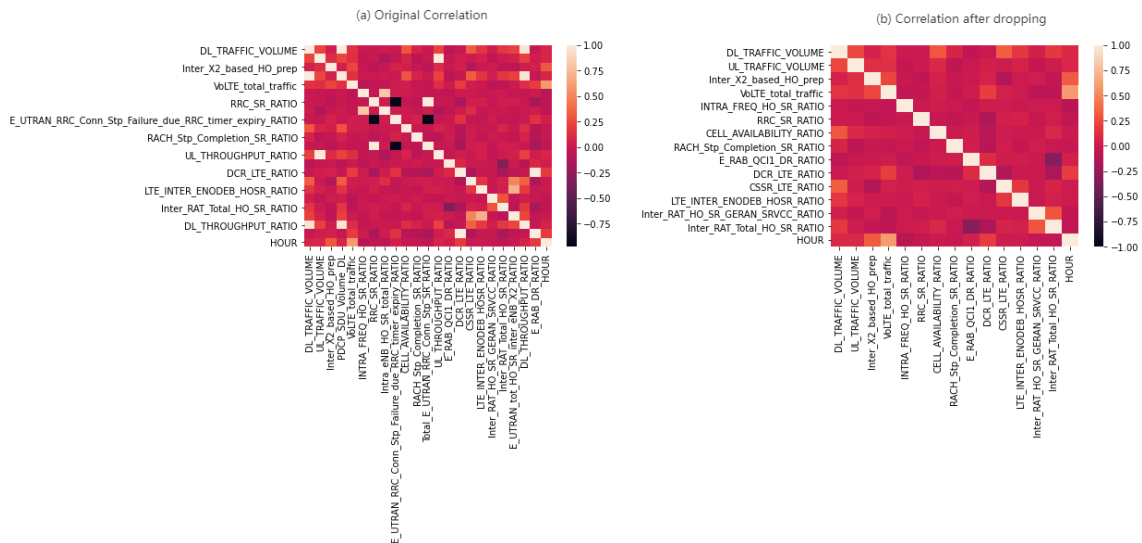
- fill with mode : for some features with very concentrated distribution (e.g. `Inter_RAT_HO_SR_GERAN_SRVCC_RATIO`, row3, col1 in figure above)
- fill with mean : for features with relatively large variance (e.g. `VoLTE_total_traffic`, row1, col2 in figure above)
- fill with specific value depending on the definition of feature : the data distribution of certain features varies widely across cells (e.g. `E_UTRAN_Inter_Freq_HO_SR_RATIO`, row3, col3 in figure above. Its left and right parts are actually the distribution in two different cells).

In our case, the median is used. This is because it is tested that the median of these characteristics is very close to the center of one of the distributions. Of course, we can be very flexible in our approach for this type of features



## (2) Highly relevant features

When modeling, features that are highly correlated can cause redundancy. We can simplify the model by keeping only the features with weak correlation.



Since our data itself is not particularly high dimensional and the correlation between the features presented in each cell is different, we need to be careful when removing them. To do this, we can record the features that are identified as redundant in each cell, and then remove those that are redundant for most cells (4 cells out of 5, in our case), and there are 6 features meet the criteria to delete.

### 2.1.3 Fill NaN & Add new features

According to the above analysis, we first remove the redundant feature values and then fill them according to different methods

On top of the above, we will also consider the effects of time of day ( hour of a day) and cell.

Finally, we get 8279 data containing 20 features.

## 2.2 Modeling

### 2.2.1 Isolation Forest

#### (1) Introduction

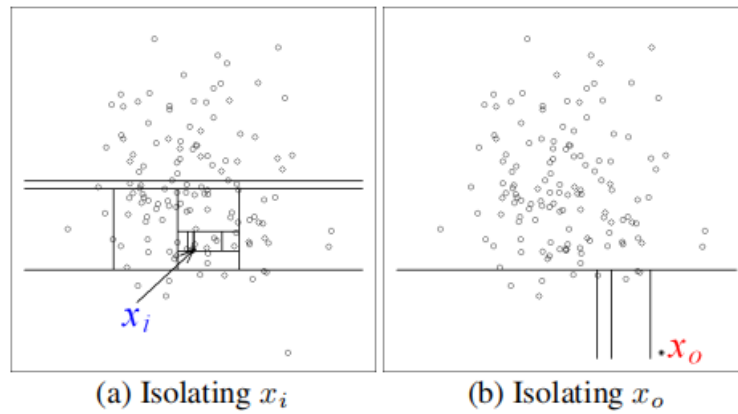
The isolation forest algorithm is proposed for detecting outliers. Regarding the anomalies, they are considered to have the following two characteristics:

- they are the minority consisting of fewer instances
- they have attribute-values that are very different from those of normal instance

In brief, the solution idea proposed by the isolation forest is as follows:

1. Randomly select a feature and its segmentation value
2. Recursively segment the dataset only until it is indivisible/attains maximum depth

In general, the more easily the point is to be isolation the more likely it would be an anomaly.



## (2) Implementation

Since there is already a wrapped isolation forest algorithm in python's `sklearn` library, we can use it directly. Here, we use the default parameters:

```
1 from sklearn.ensemble import IsolationForest
2
3 IF=IsolationForest(n_estimators=150,
4                     max_samples='auto',
5                     max_features=1, random_state=42)
```

Note: For this algorithm, we do not need to scale the features

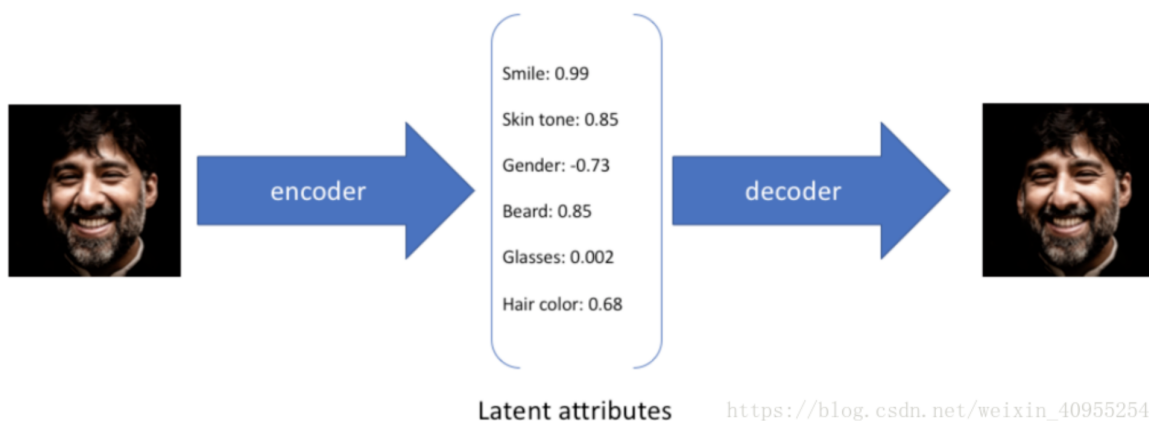
After the model is trained, we use it to calculate the anomaly score for each sample (the higher the score, the more likely it is to be an outlier)

## 2.2.2 Variational Auto-Encoder (VAE)

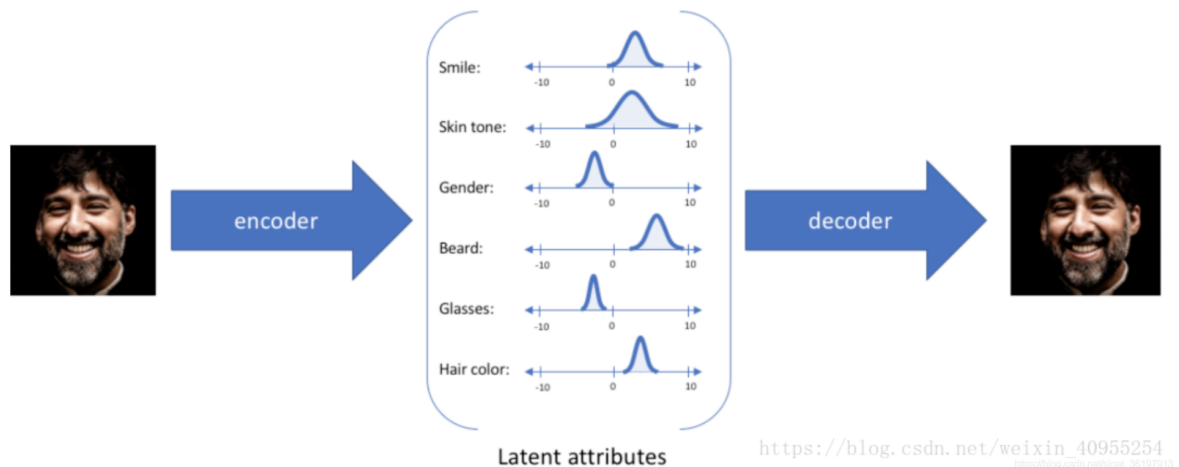
### (1) Introduction

Before getting into VAE, let's get to see its simplified version first - the *AutoEncoder* (AE).

**AutoEncoder** is based on neural networks. Its core idea is to take the features of the learned data and thus reduce the dimensionality (usually in a non-linear way), and this process is called **encoding**. And to retrieve them from the "compressed" data, we need to perform the **decoding** process. Of course, the decompressed data will not be exactly the same as the original one. Therefore, the main purpose of training the model is to reduce the discrepancy between the two.



The autoencoder is a single-valued mapping model (compressing a sample point into another point and then retrieving it), while the variational autoencoder looks for the mapping relationship of the distribution. Specifically, the VAE learns their distribution from the "compressed points", so that any point in this "compressed space" can find its counterpart in the original space.



Based on the above features, we can apply AE to detect outliers and use VAE to generate new data that is not available in reality. Although our aim is to detect anomalies, given the other requirements of this project, we will also use the VAE model.

## (2) Implementation

As before, we start with a relatively simple model structure. Specifically, in the encoder and decoder, we just use two dense layers:

Model: "encoder"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 20)]	0	[]
dense (Dense)	(None, 16)	336	['input_1[0][0]']
dense_1 (Dense)	(None, 4)	68	['dense[0][0]']
z_mean (Dense)	(None, 2)	10	['dense_1[0][0]']
z_log_var (Dense)	(None, 2)	10	['dense_1[0][0]']
lambda (Lambda)	(None, 2)	0	['z_mean[0][0]', 'z_log_var[0][0]']

=====  
Total params: 424  
Trainable params: 424  
Non-trainable params: 0

Model: "decoder"

Layer (type)	Output Shape	Param #
z_sampling (InputLayer)	[(None, 2)]	0
dense_2 (Dense)	(None, 4)	12
dense_3 (Dense)	(None, 16)	80
dense_4 (Dense)	(None, 20)	340

=====  
Total params: 432  
Trainable params: 432  
Non-trainable params: 0

For the VAE model, we need to calculate the difference (which we call "loss") between the reconstructed data and the original data, and outliers tend to have larger losses.

## 2.2.3 Combination of *IF* and *VAE*

### (1) Stacking

In *Ensemble Learning*, there is a method called "*stacking*", which means that the output of one model is used as the input of another. This provides us with a new way of thinking, and currently, there is no existing research on this approach.

Therefore, combining the above theory, we can easily come to the idea that, the output of the "encoder" can be used as input to the *IF* model. For this, we can interpret it as using the new features learned by VAE as input (they may be more representative than the original ones) and then isolating those points that are far from the majority with *IF* algorithm.

And in this section, we keep exactly the settings of the two models above, but we won't use the "decoder" of the VAE model.

### (2) Bagging

Also in integration learning, there is a method called "Bagging". It is a voting mechanism, that is, it considers multiple models equally and takes result of majority as the final answer. In this case, we do not need to reconstruct the new model. We only analyze the results of the above two existing models. Specifically, we can consider a sample as an outlier if :

- The *IF* model or the VAE model thinks it's abnormal
- OR : both models think it's abnormal

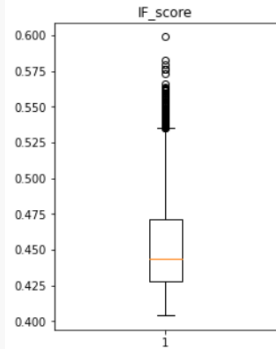
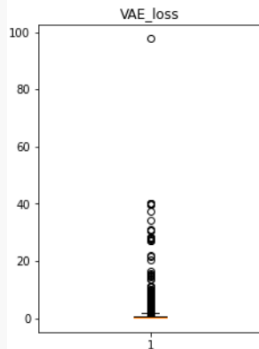
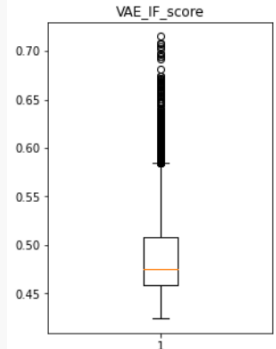
## 2.3 Result Evaluation

---

In this section, we will analyze the output metrics (abnormal probabilities or losses) of the three models mentioned above.

### 2.3.1 Analysis on metrics obtained

#### (1) Distribution

	IF_score	VAE_loss	VAE_IF_score
mean	0.451213	0.747234	0.486291
std	0.030114	2.107643	0.039233
min	0.403969	0.023347	0.423997
25%	0.428277	0.194180	0.458176
50%	0.443766	0.377740	0.474988
75%	0.471093	0.776217	0.508637
max	0.599004	97.686493	0.715104
visualization			

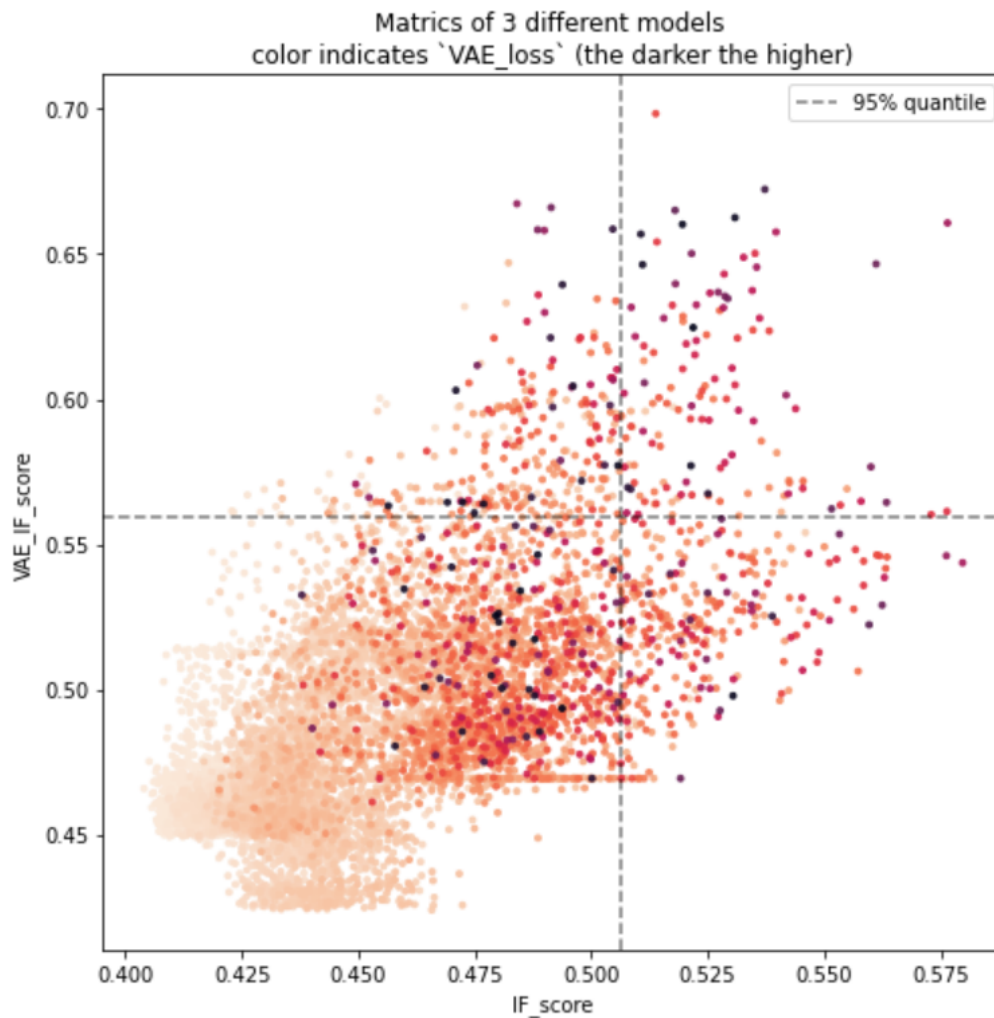
Note: In the above visualization a box line plot is used, which uses a method called **IQR** (which we will explain in detail later).

- the orange line represents the median
- the upper and lower boundaries of the rectangle box represent the upper(Q3) and lower quartiles(Q1)
- the two line segments outside of the box represent the upper and lower boundaries
- the circles beyond the boundaries are considered as outliers.

It can be seen that the two algorithms related to isolation forest get similar results. And for the *VAE\_loss*, there is a large discrepancy in its value (the mean value is 0.74 while the max value is 97.6). That is because the loss of VAE can be taken without an upper limit.

## (2) Interrelation

Next, we put the three metrics together to observe their interrelationship. Since the value range of *VAE\_loss* differs significantly from the other two, we use color to indicate it (the darker the color, the larger the value).

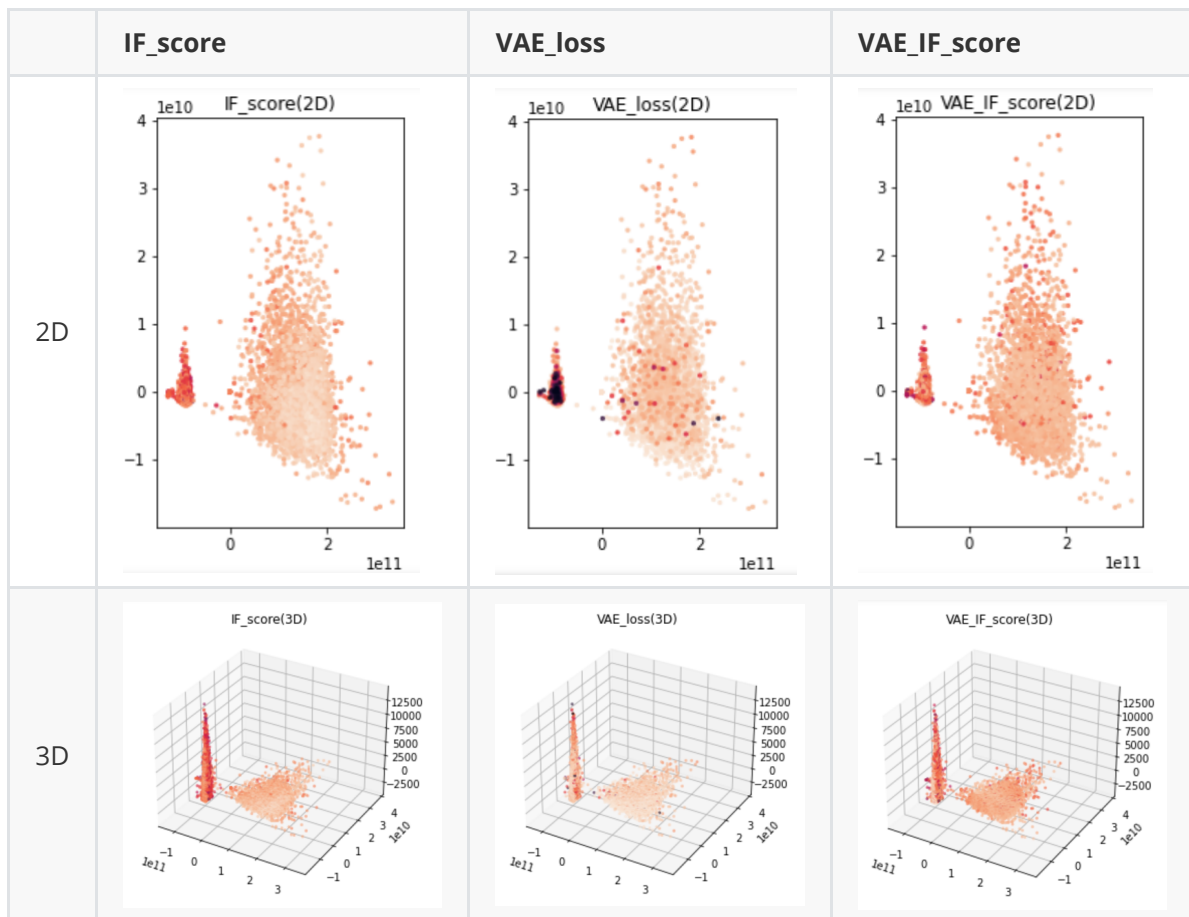


It can be seen that :

1. In the lower left corner of the image (  $x \in [0, 0.45]$  and  $y \in [0, 0.55]$  )  
the values of all three indicators are low, which means that it is less likely to be an outlier
2. In the upper part of the image ( $y > 0.57$ ) as well as in the right part ( $x > 0.51$ ), when both `IF_score` and `VAE_IF_score` have higher scores, `VAE_loss` also tends to have higher values (i.e., all three metrics are considered more likely to be anomalous)
3.  $x \in [0.46, 0.51]$  and  $y \in [0.47, 0.56]$   
`VAE_IF_loss` indicates high loss, but for the other two indicators, the same conclusion is not always reached

### (3) Visualize metrics on original data

Since our original data is 20-dimensional, in order to visualize them, we can use *PCA* to downscale it to 2~3 dimensions. Of course, reducing the dimensionality will introduce some information loss, but we are here mainly to facilitate the demonstration of the results.



In the above figure, we compressed the original 20-dimensional data to 2~3 dimensions respectively for visualization, where the darker color indicates the higher possibility of anomaly. It can be seen that:

- **Isolation Forest** is good at finding out the data edges as well as very small clusters.
- **VAE** is very sensitive to clustering who is far away from majority
- **VAE+IF**: the difference in values is relatively small, but we can still find some points at the edges where the possibility of anomalies is high

## 2.3.2 Identification of anomalies

### IQR method

In descriptive statistics, the interquartile range (IQR) is **a measure of statistical dispersion**, which is the spread of the data. It is defined as the difference between the 75% and 25% percentiles of the data.

Mathematically, we have:

- $IQR = Q3 - Q1$
- $upper\_bound = Q3 + 1.5 * IQR$
- $lower\_bound = Q1 - 1.5 * IQR$

where  $Q1$ ,  $Q3$  are 25% and 75% quantile respectively. But in our case, the lower bound won't be used, since the lower the metric value is, the less possible a record will be an anomaly.

## (1) Result - Stacking Model

To make it clear, we calculate the upper bound (threshold) and its corresponding quantile and get the following results:

	IF_score	VAE_loss	VAE_IF_score
upper bound	0.54	1.65	0.58
Corresponding Quantile	98.90%	91.94%	97.26%
Anomaly percentage	<b>1.10%</b>	<b>8.06%</b>	<b>2.74%</b>

That is, if we use the IQR method to calculate the threshold for determining whether a sample is anomalous or not, then for each of the three models mentioned above, the proportion of outliers we obtain is 1.10%, 8.06% and 2.74%

It can be seen that the result obtained using a combination of the two models (IF + VAE) is between that of Isolation Forest and VAE.

## (2) Result - Bagging Model

If we count the number of times a sample is determined to be an outlier by the Isolated Forest model as well as the VAE model, we can get the following results

Nb of being recorded as anomaly	Percentage	Explanation
0	91.6053%	Not considered as anomaly
1	07.6338%	Considered as anomaly by IF <b>or</b> VAE
2	00.7610%	Considered as anomaly by IF <b>and</b> VAE

This means that if there are  $n$  model(s) that consider a sample to be an outlier, we judge it to be an anomaly, and then the relationship between  $n$  and the proportion of anomalous samples is:

- $n = 1$  : **8.4%**, that is, the "union" of the 2 models' results
- $n = 2$  : **0.76%**, that is, the "intersection" of the 2 models' results

## 2.3.3 Conclusion

Based on the basic isolated forest and VAE models, we use two integrated learning methods, "stacking" and "bagging", with the former yielding results in between the results of the basic model and the latter falling outside this range.



But for this, we can't directly decide which result is better. In fact, we need further expert opinion (a priori knowledge) to determine the performance of the model.



## 3. Summary & Reflection

---

### 3.1 Problems encountered

---

#### 3.1.1 Handling large number of missing values

About half of the features in the data provided have more than 40% missing values. How to handle them and minimize the impact on the distribution of the original data? I spent a lot of time on this problem.

At first, I tried to start with understanding the physical meaning of the features. It would greatly simplify the problem if it could be simply filled with 0s or 1s (e.g., if there is a feature which records the number of communications, then for missing values we can simply fill it with 0s). Unfortunately, in our data, features are always related about the success rate of connections and some features seem to be correlated with each other. Therefore this approach is not suitable.

Consider that the time is recorded in the data. Next, I tried to use the interpolation method. However, some features are missing for the entire time period recorded.

At Mr. Hoayek's suggestion, I divided the data into cells and realized that the data was missing because some features were not recorded in a particular cell and were not "randomly" generated. In each region (cell), the distribution and correlation of the values of the features are more easily observed. On this basis, I was able to use the method mentioned in the previous section for the missing values.

#### 3.1.2 Evaluation of model output

In this project, our model is unsupervised. This means that there are outliers in our data, but we do not know which and how many are. Therefore, even when the output of each model is available, it is not feasible to evaluate the performance of the model using concepts such as "accuracy" that are common in supervised learning.

For this, I visualized the results and calculated the percentage of records that are considered anomalous based on the models' output. Although these are not "precise" evaluation results, they still provide some degree of feedback on the performance of the model. This is why, in unsupervised learning, we need the a priori knowledge of experts to help us adjust and improve the model.

### 3.2 Knowledge acquired

---

#### 3.2.1 Theoretical aspect

In the theoretical phase of this project, I learned the theoretical concepts of Isolation Forest and (Variational) Auto-Encoder. More importantly, from this starting point, I reviewed and learned more about statistics (mixed Gaussian models, cross-entropy...), and thus able to understand the models from a more "statistical learning method" perspective.

### 3.2.2 Practical aspect

In this project, I had the opportunity to apply knowledge about unsupervised learning in a practical scenario and to start the content of deep learning based on Tensorflow (Keras). Even though I didn't use very advanced techniques, it's still a very meaningful start for me, and I will continue to work in this field.

## 3.3 Possible improvements

---

This project is my initial exploration in anomaly detection, therefore I did not do a very deep exploration in the construction of the model (using very simple parameter settings instead). Therefore, in this regard, there is still more learning and improvement to be done.

In addition to that, the evaluation of unsupervised models is a very open problem in this project. In my previous studies, there was no experience in this area. In industry, there should be some established practices to follow, and it may also take into account the needs of "real-time applications". These are things that I can explore further based on the work I have done so far.