IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Supervision des systèmes

Projet de détection d'intrusion basé sur Netflow
Algorithme KGB

21 mars 2025

Dan ACHILLE - Damien CHARCONNET - Alexandre CUZOU - Arthur DEGEORGES

PLAN

PARTIE 1

Introduction à l’algorithme KGB

ARTICLE

COMPUTERS & SECURITY 45 (2014) 100–123

Available online at www.sciencedirect.com

ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

An empirical comparison of botnet detection methods

S. García [a,b,*], M. Grill [b], J. Stiborek [b], A. Zunino [a]

[a] ISISTAN Research Institute – CONICET, Faculty of Sciences, UNICEN University, Argentina
[b] Agents Technology Group, Department of Computer Science and Engineering, Czech Technical University in Prague, Czech Republic

ARTICLE INFO

Article history:
Received 21 October 2013
Received in revised form
29 April 2014
Accepted 27 May 2014
Available online 5 June 2014

Keywords:
Botnet detection
Malware detection
Methods comparison
Botnet dataset
Anomaly detection
Network traffic

ABSTRACT

The results of botnet detection methods are usually presented without any comparison. Although it is generally accepted that more comparisons with third-party methods may help to improve the area, few papers could do it. Among the factors that prevent a comparison are the difficulties to share a dataset, the lack of a good dataset, the absence of a proper description of the methods and the lack of a comparison methodology. This paper compares the output of three different botnet detection methods by executing them over a new, real, labeled and large botnet dataset. This dataset includes botnet, normal and background traffic. The results of our two methods (BClus and CAMNEP) and BotHunter were compared using a methodology and a novel error metric designed for botnet detections methods. We conclude that comparing methods indeed helps to better estimate how good the methods are, to improve the algorithms, to build better datasets and to build a comparison methodology.

1. Introduction

It is difficult to estimate how much a new botnet detection method improves the current results in the area. It may be done by comparing the new results with other methods, but this has already been proven hard to accomplish (Aviv and Haeberlen, 2011). Among the factors that prevent these comparisons are: the absence of proper documentation of the methods (Tavallaee et al., 2010), the lack of a common, labeled and good botnet dataset (Rossow et al., 2012), the lack of a comparison methodology (Aviv and Haeberlen, 2011) and the lack of a suitable error metric (Salgarelli et al., 2007).

Although the comparison of methods can greatly help to improve the botnet detection area, few proposals made such a comparison (García et al., 2013). As far as we know, only three papers (Wurzinger et al., 2010; Zhao et al., 2013; Li et al., 2010) made the effort so far.

Obtaining a good dataset for comparisons is difficult. Currently, most detection proposals tend to create their own botnet datasets to evaluate their methods. However, these datasets are difficult to create (Lu et al., 2009) and usually end up being suboptimal (Shiravi et al., 2012), i.e. they lack some important features, such as ground-truth labels, heterogeneity or real-world traffic. These custom datasets are often difficult to use for comparison with other methods. This is because each

* Corresponding author. ISISTAN Research Institute–CONICET, Faculty of Sciences, UNICEN University, Argentina.
E-mail addresses: sebastian.garcia@isistan.unicen.edu.ar, eldraco@gmail.com (S. García), grill@agents.fel.cvut.cz (M. Grill), jan.stiborek@agents.fel.cvut.cz (J. Stiborek), ajzunino@isistan.unicen.edu.ar (A. Zunino).
http://dx.doi.org/10.1016/j.cose.2014.05.011
0167-4048/

"An empirical comparison of botnet detection methods"

2014

Année de publication dans le cadre du journal Computers & Security

Auteurs affiliés à l'université **UNICEN University,** Argentina et **Czech Technical University**, Prague - S. García, M. Grill, J. Stiborek, A. Zunino

Article cité

1024 fois

INTRODUCTION À L'ALGORITHME KGB

KGB est basé sur l'article de Pevny et al. (2012)
"Identifying suspicious users in corporate networks"

- **Plus performant que les autres algorithmes**
- **Temps d'entraînement plus long et énergivore**
- **Le plus proche des solutions d'entreprise**
- **Recherche de complexité**

INTRODUCTION À L'ALGORITHME KGB

IDÉE

Basé sur les travaux de l'entropie de Lakhina, l'anomalie finale est déterminée à partir des écarts moyens

BUT

La méthode utilise les entropies des distributions d'adresses IP et de ports pour construire deux modes de détection

Il existe deux versions du détecteur KGB:

KGBf examine les composantes principales avec de fortes variances.

KGBfog examine les composantes principales avec de faibles variances.

PARTIE 2

Exposé des traitements sur dataset

CTU-13 DATASET

Stratosphere Lab

Dataset CTU-13
Captures réseau (2011, CTU University, République Tchèque) 🇨🇿

Flux labellisés :
botnet, normal, background → Évaluation fiable.

Trafic mixte :
normal, malveillant, bruit de fond.

Scénario 10, idéal pour KGB :
DDoS UDP intense, contexte réaliste, variations statistiques marquées.

EXTRACTION ET CONVERSION DES DONNÉES

Fichiers .binetflow → CSV : Analyse simplifiée (pandas, scikit-learn…).

```
StartTime,Dur,Proto,SrcAddr,Sport,Dir,DstAddr,Dport,State,sTos,dTos,TotPkts,TotBytes,SrcBytes,Label
2011/08/18 10:21:46.633335,1.060248,tcp,93.45.239.29,1611,   ->,147.32.84.118,6881,S_RA,0,0,4,252,132,flow=Background-TCP-Attempt
2011/08/18 10:19:49.027650,279.349152,tcp,62.240.166.118,1031,  <?>,147.32.84.229,13363,SRPA_PA,0,0,15,1318,955,flow=Background-TCP-Attempt
2011/08/18 10:22:07.160628,166.390015,tcp,147.32.86.148,58067,   ->,66.235.132.232,80,SR_SA,0,0,3,212,134,flow=Background-TCP-Established
2011/08/18 10:26:02.052163,1.187083,tcp,147.32.3.51,3130,   ->,147.32.84.46,10010,S_RA,0,0,4,244,124,flow=Background-TCP-Attempt
```

Colonnes principales:

- ➔ **StartTime / Dur** : début & durée du flux
- ➔ **Proto** : protocole (TCP, UDP…)
- ➔ **SrcAddr / DstAddr** : IP source & destination
- ➔ **Sport / Dport** : ports source & destination
- ➔ **State** : état de la connexion
- ➔ **TotPkts / TotBytes** : paquets & octets échangés
- ➔ **Label** : nature du trafic (Background, Botnet…)

NETTOYAGE ET TRANSFORMATION

Conversion des dates:

```
df['is_weekend'] = df['timestamp'].dt.weekday.apply(lambda x: 1 if x >= 5 else 0)
df['is_work_hours'] = df['timestamp'].dt.hour.apply(lambda h: 1 if 9 <= h < 17 else 0)
```

- *Flags* temporels (heures de bureau, heures tardives…)
- ***is_weekend*** Flag binaire (1/0) indiquant si l'enregistrement a lieu durant le week-end. Calculé à partir de ***timestamp.dt.weekday*** (valeur 1 si >= 5).
- ***is_work_hours*** Flag binaire (1/0) indiquant si l'enregistrement se situe pendant les heures de travail (entre 9h et 17h).

Gestion des ports:

```
df['is_dst_port_well_known'] = (df['dst_port'] < 1024).astype(int)
df['is_src_port_well_known'] = (df['src_port'] < 1024).astype(int)
```

- Détection de ports ***well-known*** (ex. < 1024) ou *suspicious* (liste prédéfinie, ex. 6667 pour IRC malveillant).

PARTIE 3

L’algorithme KGB

L'ALGORITHME KGB

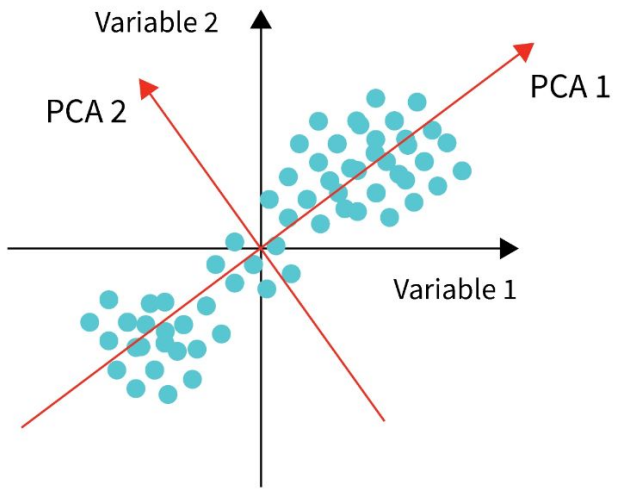
Définition

KGB est un algorithme de détection d'anomalies qui, via l'analyse en composantes principales (PCA) sur la matrice d'entropie, sépare les grandes variances (composantes majeures) et les petites variances (composantes mineures) pour repérer à la fois les anomalies franches et les comportements anormaux plus subtils.

- Entropie de Shannon : $$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log \mathrm{P}(x_i)$$
- PCA : Analyse en composantes principales
- Distance de Mahalanobis :

$$f(x^t(\iota)) = \sum_{j=1}^{k} \frac{(y_j^T x^t(\iota))^2}{\lambda_j^2},$$

$$f^{\perp}(x^t(\iota)) = \sum_{j=k+1}^{r} \frac{(y_j^T x^t(\iota))^2}{\lambda_j^2}.$$



L'ALGORITHME KGB

3 exemples d'entropie :

***HsPr* (entropie des ports sources)** : variation des ports sources utilisés par un utilisateur.

***HdPr* (entropie des ports destinations)** : diversité des ports vers lesquels l'utilisateur envoie du trafic.

***HdIP* (entropie des adresses IP de destination)** : mesure de la dispersion des communications vers différentes IP.

- Un utilisateur naviguant normalement sur le web aura une faible entropie de ports destination ***HdPr*** et une faible entropie d'adresses IP de destination ***HdIP.***
- Un utilisateur scannant les ports ouvert enverra du trafic vers un grand nombre de ports destination, ce qui augmentera l'entropie ***HdIP***.

Explication étape par étape

= Hyperparamètres

1. On groupe le Dataframe de donnée suivant l'IP source où on calcule l'entropie-type de chaque attribut par IP
2. On génère une matrice d'entropie normalisé avec les IP sources en ligne et les entropies-types en colonne
3. On calcule une matrice PCA pour réduire la dimension des attributs en conservant les composantes principales
4. L'analyse en composantes principales (PCA) identifie les directions majeures permettant d'expliquer la variance
5. On divise ces composantes en deux catégories KGBf (composantes majeurs) et KGBfog (composantes mineures)
6. On calcule la distance global de Mahalanobis pour KGBf et KGBfog
7. On teste si la distance pour chaque IP source est supérieur à un des pourcentages des distances globals (KGBf et KGBfog)
8. Si oui, c'est une anomalie.

   Si non, c'est normale.

LES HYPERPARAMÈTRES

1. Taille de l'échantillon

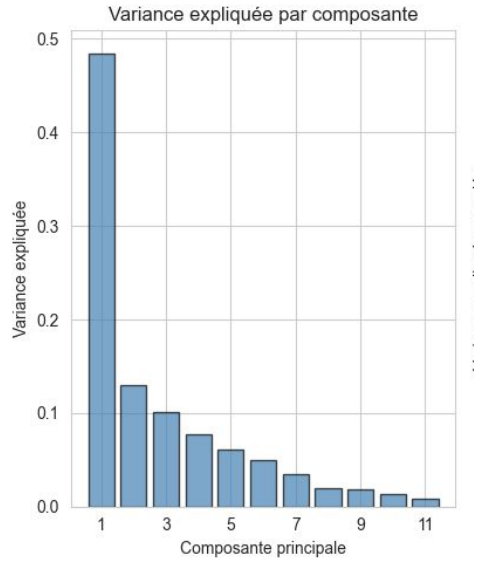
2. Seuil en pourcentages de variance à expliquer

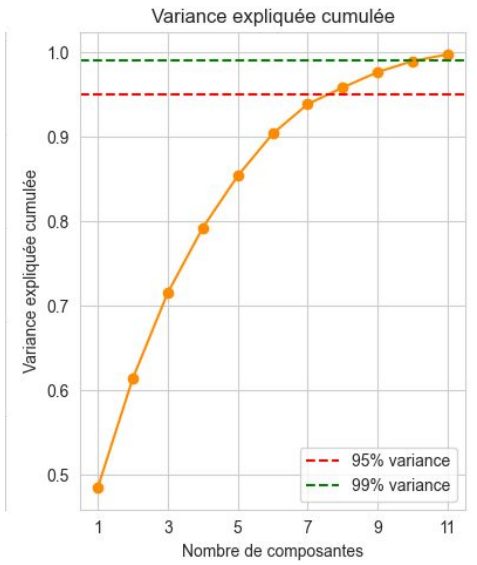
$$PCA\ [\ C_1, C_2\ C_3, \dots, C_n\ ]\ \dots C_z$$

3. Nombre de composantes principales

4. Seuil d'anomalie pour les composantes principales

5. Seuil d'anomalie pour les composantes secondaires





L'ALGORITHME KGB

Détection des anomalies

Deux scores sont utilisés:

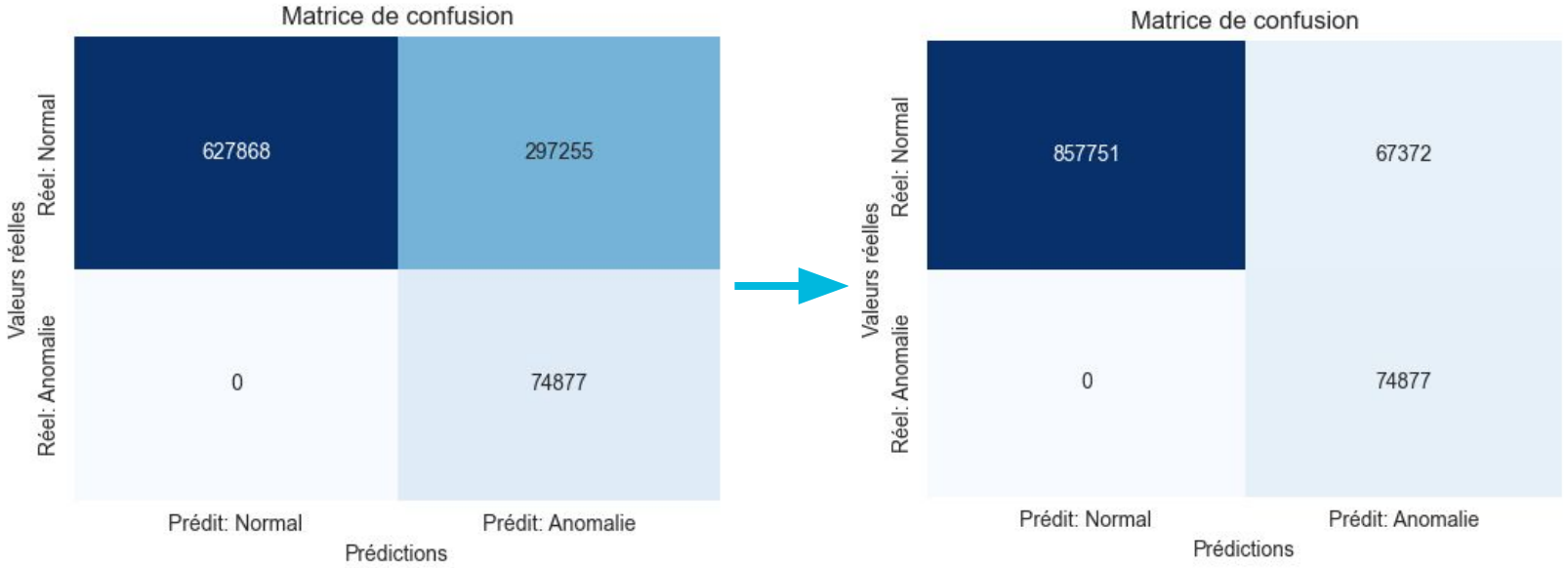
- Variance projetée sur les principales composantes (mesure de dispersion des comportements normaux).
- Variance dans l'espace résiduel (où les anomalies sont supposées se situer).

Si ces scores dépassent un seuil défini, l'utilisateur est identifié comme suspect.

PARTIE 4

Analyse des résultats

ANALYSE DES RÉSULTATS



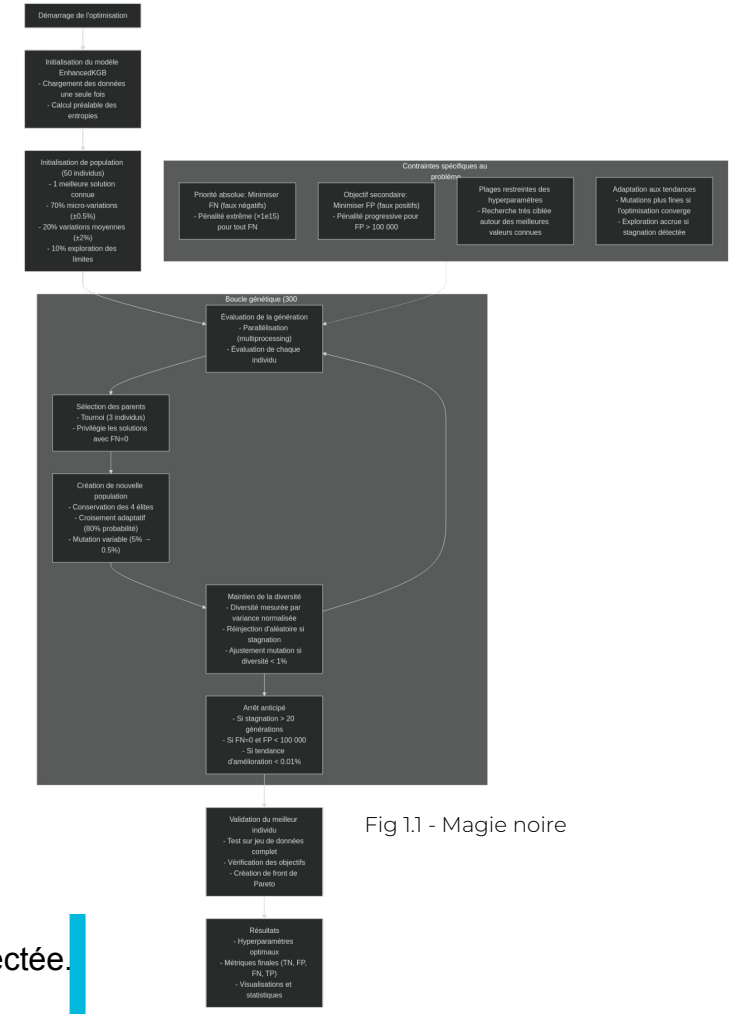


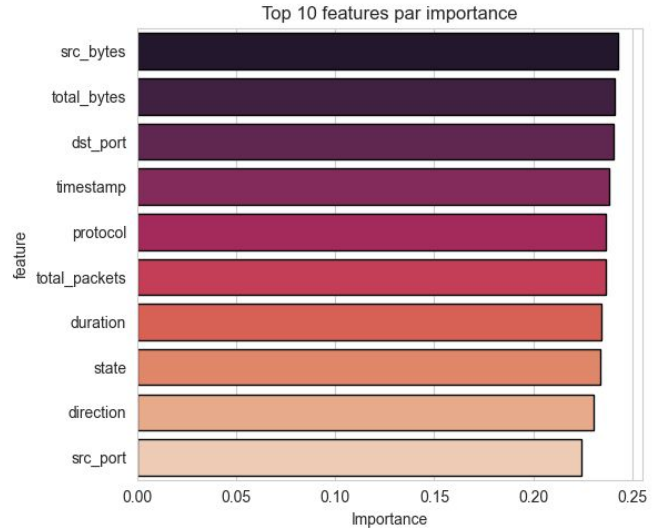
Fig 1.1 - Magie noire

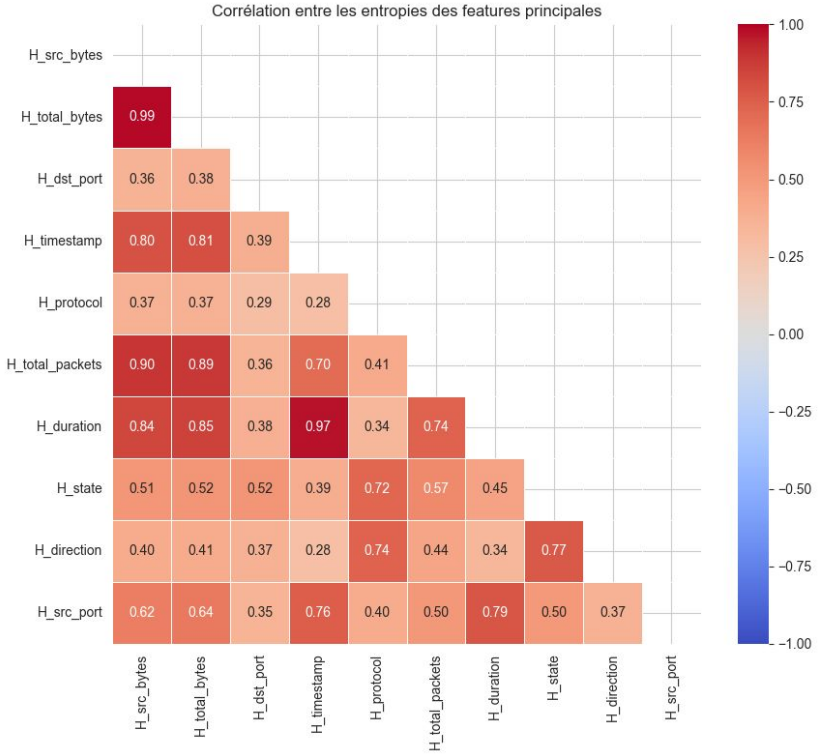
Ajustement des hyperparamètres

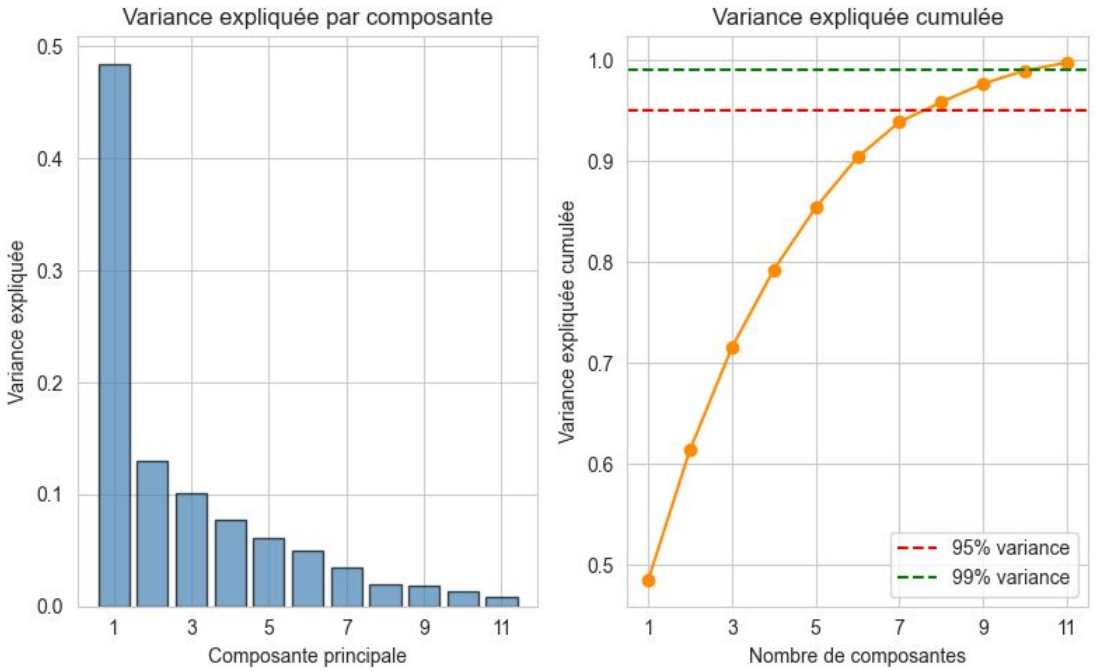
- ➔ **n_major_components** : Nombre de composantes principales utilisées pour calculer le score majeur d'anomalie.
- ➔ **threshold_PCA** : Pourcentage de la variance à conserver lors de la décomposition en composantes principales.
- ➔ **threshold_major** : Seuil déterminant le niveau critique pour le score majeur, au-delà duquel une anomalie est suspectée.
- ➔ **threshold_minor** : Seuil déterminant le niveau critique pour le score mineur, complétant l'analyse d'anomalies.

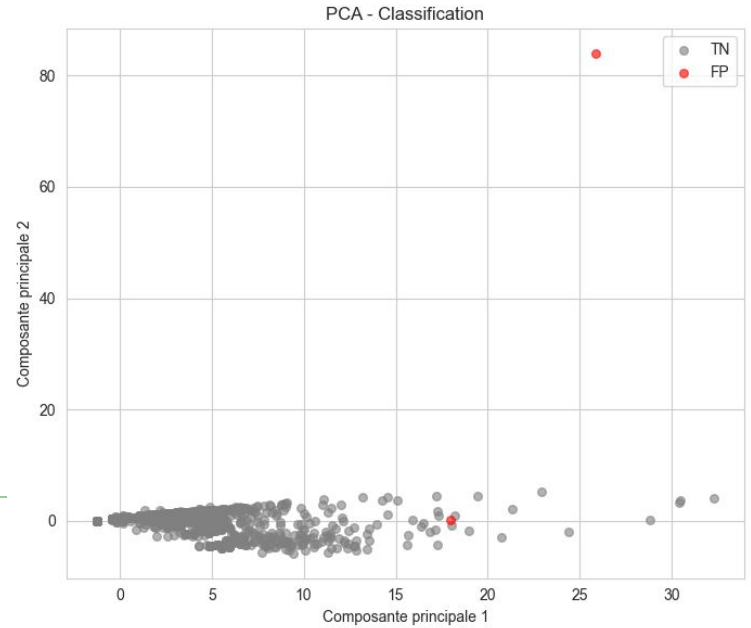
Score F1 : 0.68971 ; Précision : 0.52637 ; Recall : 1.00

ANALYSE DES RÉSULTATS









PARTIE 5

Perspectives d’amélioration

AMÉLIORATIONS À IMPLÉMENTER

Sélection précise des attributs

Choisir les paramètres réellement pertinents et **affiner les hyperparamètres** pour maximiser la performance de l'algorithme KGB

Comparaison rigoureuse

Confronter la précision de nos résultats à ceux de l'article de référence pour **évaluer l'apport** de ces choix

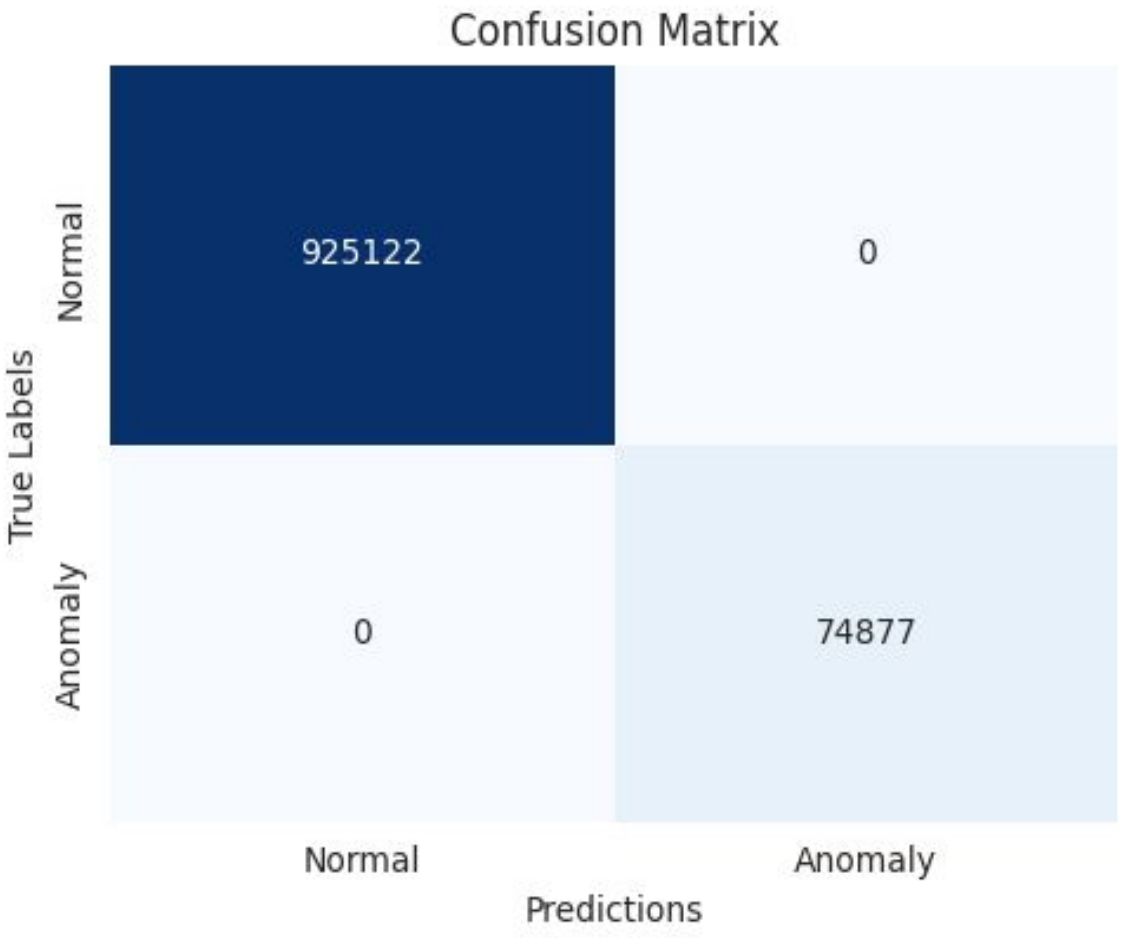
Implémenter une solution plus efficace

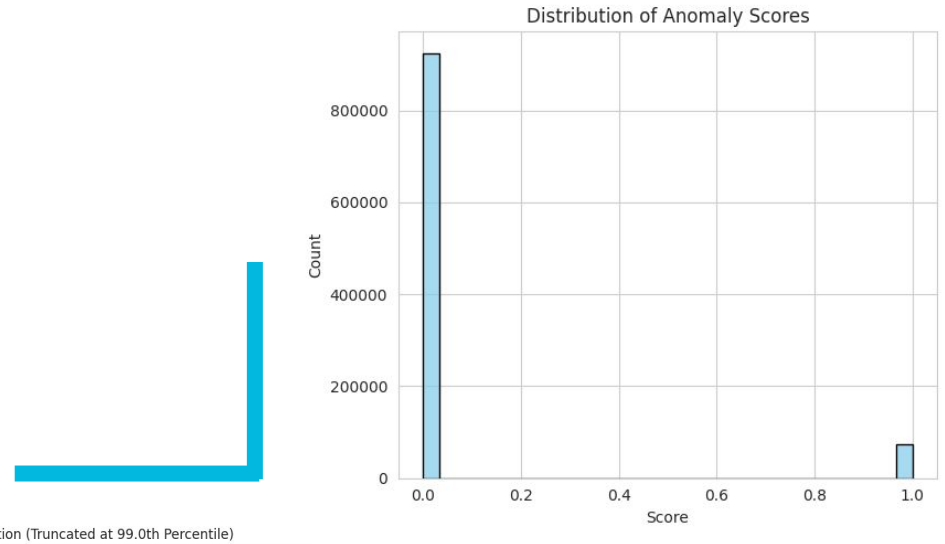
Améliorer l'efficacité de la solution afin de viser une implémentation en **temps réel**, apte à détecter **rapidement** les anomalies sur des flots de données continus

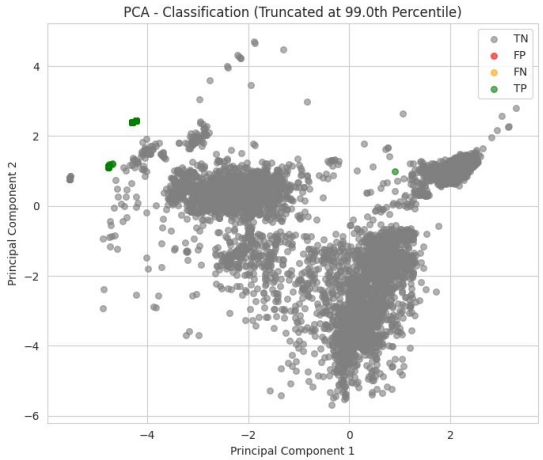
PARTIE 6

Comparaison avec KBG

Algorithme alternatif - Modèle Génétique - Algorithme KBG - K-means Based Genetic







IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Merci !

Supervision des systèmes

Projet de détection d'intrusion basé sur Netflow
Algorithme KGB

21 mars 2025

Dan ACHILLE - Damien CHARCONNET - Alexandre CUZOU - Arthur DEGEORGES