
Statistique et Data-mining dans un contexte Cyber

EL MOUSTAPHA MALICK

MASTER INGÉNIERIE STATISTIQUE ET NUMÉRIQUE – DATA SCIENCES (ISN)

Année Universitaire 2022-2023

Tuteur université :
CÉLINE DUVAL

Tuteur entreprise :
JONATHAN CLAIREMBAULT

Table des matières

1	Remerciments	2
2	Introduction	3
3	Rubycat	4
3.1	PAM	4
3.1.1	PROVE IT	5
4	Développement d'outils pour analyser les données d'audit	6
4.1	Premier outil	6
4.2	Deuxième outil	11
5	Création de modèle pour la détection de menace avec le Machine Learning	14
5.1	État de l'art	15
5.2	Création d'un prototype	20
5.2.1	Prétraitement des données	20
5.2.2	Ingénierie des caractéristiques	20
5.2.3	Construction et entraînement du modèle	22
5.2.4	Évaluation	24
5.3	Autre méthode de détection d'anomalies	26
5.4	Présentation et documentation	26
5.5	Création d'un script d'extraction des données de PROVE IT	27
6	Conclusion	28
7	Références	30

1 Remerciements

Je tiens à exprimer ma profonde gratitude envers ma mère, Mme Galyna Anatolivna SHUGAYLO, pour son soutien inconditionnel et sa motivation constante dans l'obtention de ce diplôme. Sa présence et son amour inébranlables ont été des sources d'inspiration inestimables tout au long de mon parcours. Je suis extrêmement reconnaissant d'avoir une personne aussi exceptionnelle dans ma vie.

Je souhaite également exprimer mes sincères remerciements à mon père, M. Mohamed Mohamed Cheikh MALICK, une source inépuisable d'inspiration. Son soutien indéfectible, ses encouragements et sa confiance en moi ont été des moteurs essentiels de ma réussite.

Ensuite, je tiens à exprimer ma profonde gratitude envers mon tuteur, Jonathan CLAIREMBAULT, CTO de Rubycat. Sa disponibilité, sa proximité et son interactivité ont été d'une valeur inestimable. Grâce à lui, j'ai pu progresser tant sur le plan technique que sur le plan humain. Je le remercie chaleureusement pour son soutien précieux.

Je souhaite également exprimer mes remerciements à la présidente, Mme Cathy LESAGE-BARON, pour avoir accepté ma candidature et m'avoir offert l'opportunité de réaliser mon stage de fin d'études chez Rubycat. Sa proximité, ses conseils continus et sa confiance en mes compétences ont été des éléments déterminants de ma réussite. Je lui suis profondément reconnaissant pour son engagement envers mon développement professionnel.

En outre, je tiens à exprimer ma gratitude envers la responsable de formation Céline DUVAL, pour sa présence et son soutien tout au long de mon stage. Ses discussions éclairantes et son appui ont contribué à trouver les meilleures solutions aux défis rencontrés. Je lui adresse mes remerciements les plus sincères.

Je souhaite également adresser mes remerciements les plus sincères à M. Anthony DAVID, Directeur technique de Rubycat, pour sa disponibilité et son aide précieuse dans la résolution de problèmes techniques. Sa gentillesse et sa collaboration ont rendu ma collaboration avec lui très agréable.

Enfin, je tiens à remercier chaleureusement tous les collaborateurs de Rubycat pour leur aide, leurs discussions amicales et les moments de convivialité partagés pendant le stage. Leur soutien et leur collaboration ont contribué à créer un environnement de travail positif et enrichissant.

2 Introduction

Ce document constitue le rapport du stage de fin d'études pour la session 2022-2023 du Master 2 : Ingénierie Statistique et Numérique – Data sciences (ISN) à l'Université de Lille.

Ce stage a été réalisé au sein de la société Rubycat. Du côté de l'entreprise, il a été supervisé par mon tuteur, Jonathan CLAIREMBAULT, CTO de Rubycat. Le professeur référent à l'Université de Lille était Céline DUVAL, responsable du Master 2 ISN.

Dans ce rapport de stage, le travail est divisé en quatre parties :

1. La première partie présente l'entreprise Rubycat ainsi que sa solution commerciale, PROVE IT, qui est une plateforme de gestion des accès au système d'information(SI).
2. La deuxième partie aborde les outils que j'ai développés pour effectuer l'analyse d'audit de la solution PROVE IT.
3. La troisième partie concerne la création d'un modèle pour détecter les menaces de sécurité liées à PROVE IT basé sur l'apprentissage automatique.
4. La quatrième partie est dédiée à une conclusion générale qui synthétise les principales conclusions et enseignements tirés de mon stage.

Chacune de ces parties contribue à une compréhension approfondie de mon travail réalisé lors de ce stage et permet de mettre en valeur les différents aspects de mon expérience.

3 Rubycat

Rubycat est une startup dynamique basée à Rennes, composée d'une équipe de 12 collaborateurs. Spécialisée dans le domaine de la cybersécurité et de la sécurisation des accès. Rubycat propose des solutions simples et efficaces pour résoudre le défi crucial du manque de visibilité sur les actions effectuées par les comptes à privilèges sur les systèmes d'information.

3.1 PAM

Une solution de gestion des accès à privilèges PAM (Privileged Access Management) est une solution visant à gérer et protéger les comptes utilisateurs possédant de forts privilèges (administrateurs internes et prestataires) et à gérer les accès d'administration aux équipements d'un SI.

Une solution de PAM peut englober plusieurs composants, les plus courants sont :

- Un bastion, composant principal qui servira d'intermédiaire entre un administrateur et une ressource.
- Un coffre-fort de mots de passe pour stocker les informations d'accès aux ressources.
- Un portail d'accès web, pour simplifier et sécuriser les accès externes.

Le déploiement d'un PAM permet de contrôler et de surveiller les actions effectuées par les utilisateurs privilégiés sur un actif donné.

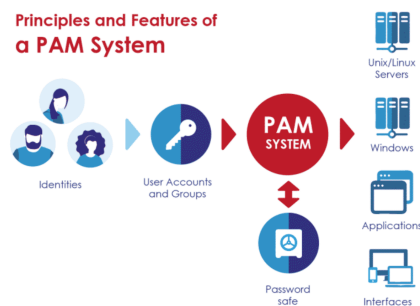


FIGURE 2 – Principes et fonctionnalités du système PAM

3.1.1 PROVE IT

Parmi ses offres, Rubycat commercialise PROVE IT, une solution logicielle de type PAM qui permet de contrôler, tracer et enregistrer les activités des comptes à privilèges sur le système d'information.

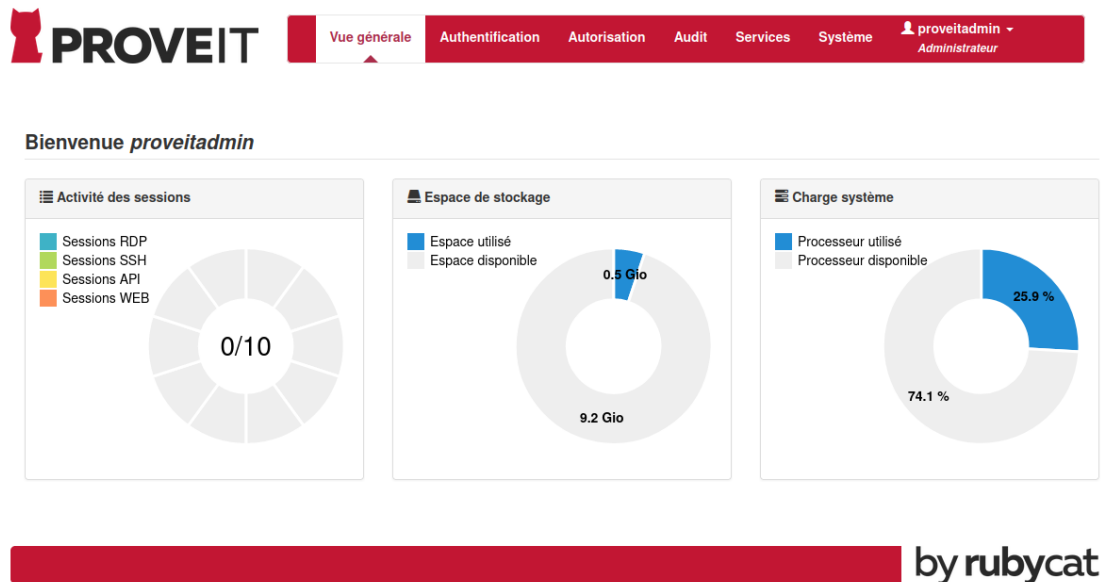


FIGURE 3 – Portail d'accès web de PROVE IT

L'utilisateur peut se connecter sur PROVE IT de trois façon :

- Session RDP : permet de prendre le contrôle d'un ordinateur à distance, visualisant son écran et exécutant des actions comme si vous étiez sur place. C'est utile pour le dépannage, la gestion de serveurs et le travail à distance.
- Session SSH : permet une connexion sécurisée à distance à un ordinateur pour l'administration en ligne de commande.
- Session WEB (HTTPS) : assure une communication sécurisée entre un navigateur web et un site, protégeant les données lors de la navigation en ligne.

4 Développement d'outils pour analyser les données d'audit

Dans l'entreprise Rubycat, une solution innovante nommée PROVE IT. Cette solution a pour objectif d'enregistrer les activités des utilisateurs au sein de l'entreprise. Installée sur les ordinateurs de l'entreprise, elle a deux fonctions essentielles : protéger les utilisateurs contre les attaques cybernétiques et enregistrer leurs sessions d'activité.

Pour commercialiser PROVE IT, Rubycat propose des licences conditionnées par le nombre de sessions simultanées, offrant ainsi une flexibilité aux entreprises pour choisir la formule adaptée à leurs besoins.

C'est dans ce contexte que débuta mon stage au sein de l'entreprise. Ma première mission consistait à configurer mon propre PC en installant un système d'exploitation sécurisé, à savoir Ubuntu. Une fois l'OS installé, mon tuteur me recommanda de tester PROVE IT sur mon propre ordinateur afin de mieux comprendre son fonctionnement et d'évaluer ses différentes fonctionnalités.

L'installation de PROVE IT nécessitait la mise en place d'une machine virtuelle (VM), pour laquelle j'utilisai l'application virt-manager. Grâce à la documentation fournie, aux conseils des membres de l'équipe R&D et aux retours sur d'éventuelles erreurs de configuration, je réussis à installer la solution avec succès et pus tester ses différentes fonctionnalités.

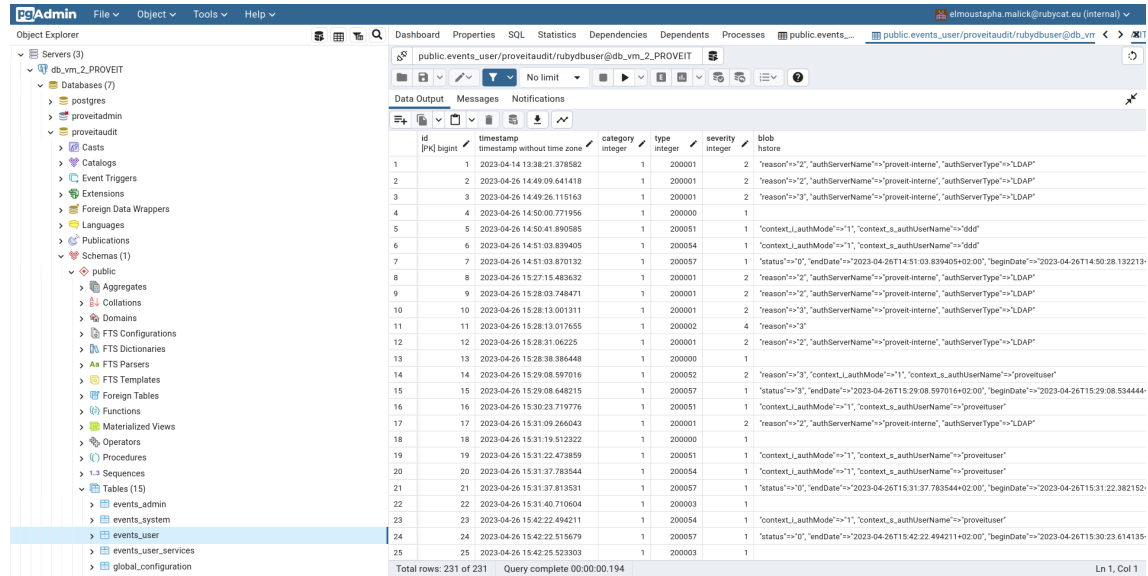
Mon tuteur prit le temps de m'expliquer brièvement le fonctionnement global de PROVE IT, ainsi que les bases du réseau, telles que le tunnel SSH, le port TCP et Bash. Ces notions me permettraient de mieux appréhender les rouages de la solution et de travailler efficacement.

4.1 Premier outil

Au fur et à mesure que j'avancais dans le développement de mon script, l'excitation grandissait. J'avais réussi à installer mon environnement virtuel sur ma machine virtuelle, découvrant ainsi les outils Git, GitLab et SQLAlchemy qui allaient m'être d'une grande utilité.

Avec l'aide précieuse de mon tuteur et après une documentation approfondie sur SQLAlchemy, je parvins à comprendre le script nécessaire de PROVE IT qui m'aiderait à créer ma fonctionnalité tant attendue.

Puis vint le moment de faire l'interfaçage entre les tables de bases de données de PROVE IT et l'ORM SQLAlchemy, tout en permettant une visualisation claire et concise des données. Grâce à l'outil pgAdmin pour PostgreSQL, j'eus la possibilité de représenter la structure des tables de la base de données de manière visuelle, facilitant ainsi la compréhension globale du système.



The screenshot shows the pgAdmin interface with the 'public.events_user' table selected. The table structure is as follows:

id	timestamp	category	type	severity	blob
1	2023-04-14 13:38:21.378582	1	200001	2	'reason'=>'2', 'authServerName'=>'proveit-interne', 'authServerType'=>'LDAP'
2	2023-04-26 14:49:09.641418	1	200001	2	'reason'=>'2', 'authServerName'=>'proveit-interne', 'authServerType'=>'LDAP'
3	2023-04-26 14:49:26.115163	1	200001	2	'reason'=>'3', 'authServerName'=>'proveit-interne', 'authServerType'=>'LDAP'
4	2023-04-26 14:50:00.771956	1	200000	1	
5	2023-04-26 14:50:41.890585	1	200051	1	'context.L_authMode'=>'1', 'context.s_authUserName'=>'ddd'
6	2023-04-26 14:51:03.839405	1	200054	1	'context.L_authMode'=>'1', 'context.s_authUserName'=>'ddd'
7	2023-04-26 14:51:03.870132	1	200057	1	'status'=>'0', 'endDate'=>'2023-04-26T14:51:03.839405+02:00', 'beginDate'=>'2023-04-26T14:50:28.132213'
8	2023-04-26 15:27:15.483632	1	200001	2	'reason'=>'2', 'authServerName'=>'proveit-interne', 'authServerType'=>'LDAP'
9	2023-04-26 15:28:03.748471	1	200001	2	'reason'=>'2', 'authServerName'=>'proveit-interne', 'authServerType'=>'LDAP'
10	2023-04-26 15:28:13.001311	1	200001	2	'reason'=>'3', 'authServerName'=>'proveit-interne', 'authServerType'=>'LDAP'
11	2023-04-26 15:28:13.017655	1	200002	4	'reason'=>'3'
12	2023-04-26 15:28:31.06225	1	200001	2	'reason'=>'2', 'authServerName'=>'proveit-interne', 'authServerType'=>'LDAP'
13	2023-04-26 15:28:38.386448	1	200000	1	
14	2023-04-26 15:29:08.597016	1	200052	2	'reason'=>'3', 'context.L_authMode'=>'1', 'context.s_authUserName'=>'provetuser'
15	2023-04-26 15:29:08.646215	1	200057	1	'status'=>'0', 'endDate'=>'2023-04-26T15:29:08.597016+02:00', 'beginDate'=>'2023-04-26T15:29:08.534444'
16	2023-04-26 15:30:23.719776	1	200051	1	'context.L_authMode'=>'1', 'context.s_authUserName'=>'provetuser'
17	2023-04-26 15:31:09.264043	1	200001	2	'reason'=>'2', 'authServerName'=>'proveit-interne', 'authServerType'=>'LDAP'
18	2023-04-26 15:31:22.472322	1	200000	1	
19	2023-04-26 15:31:22.473859	1	200051	1	'context.L_authMode'=>'1', 'context.s_authUserName'=>'provetuser'
20	2023-04-26 15:31:37.783544	1	200054	1	'context.L_authMode'=>'1', 'context.s_authUserName'=>'provetuser'
21	2023-04-26 15:31:37.813531	1	200057	1	'status'=>'0', 'endDate'=>'2023-04-26T15:31:37.783544+02:00', 'beginDate'=>'2023-04-26T15:31.22.382152'
22	2023-04-26 15:31:40.710604	1	200003	1	
23	2023-04-26 15:42:22.494211	1	200054	1	'context.L_authMode'=>'1', 'context.s_authUserName'=>'provetuser'
24	2023-04-26 15:42:22.515679	1	200057	1	'status'=>'0', 'endDate'=>'2023-04-26T15:42:22.494211+02:00', 'beginDate'=>'2023-04-26T15:30.23.614135'
25	2023-04-26 15:42:25.523303	1	200003	1	

Total rows: 231 of 231 Query complete 00:00:00.194 Ln 1, Col 1

FIGURE 4 – Tables de la base de données de PROVE IT

Avec ces informations en main, je me lançai dans l'élaboration du graphique d'activité des utilisateurs.

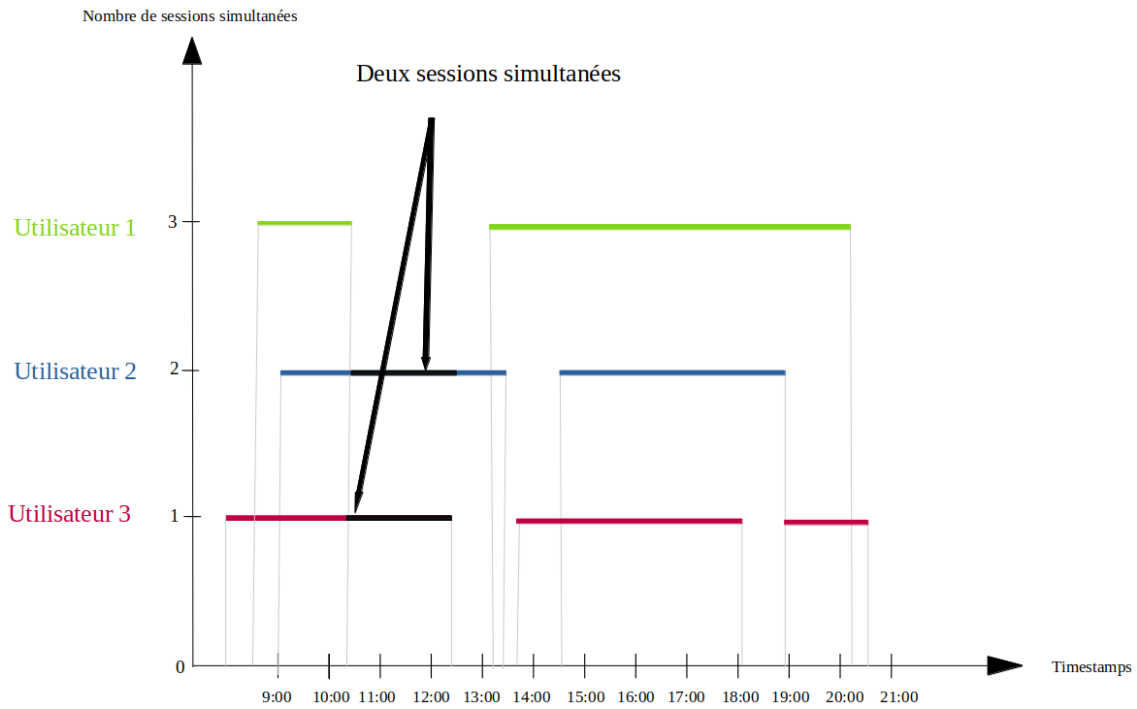


FIGURE 5 – Exemple de sessions d'activités des utilisateurs

Ce Graphique illustre des exemples de sessions sur une période donnée.

Cependant, je me retrouvai face à un défi. Pour calculer le nombre de sessions simultanées, j'avais deux approches possibles. La première impliquait de compter le nombre d'intersections d'activité des utilisateurs à un instant donné, mais cette méthode s'avéra complexe et inefficace en raison du temps de calcul nécessaire. Je me tournai donc vers la deuxième approche qui reposait sur les horodatages ou timestamps.

Avec les timestamps, j'étais en mesure de créer des paires composées de timestamp et de la valeur 1 pour une connexion ou -1 pour une déconnexion.

On obtient la liste de couple : $(t_1, +1), (t_2, -1), (t_3, +1), \dots$

$$\begin{cases} t_i : \text{Timestamp d'une connexion/déconnexion d'une session } i \\ +1 : \text{Si s'est une connexion} \\ -1 : \text{Si s'est une déconnexion} \end{cases}$$

Grâce à une liste triée selon le timestamp et une accumulation des valeurs de la deuxième partie de chaque couple, j'obtins directement le nombre de sessions simultanées pour chaque instant.

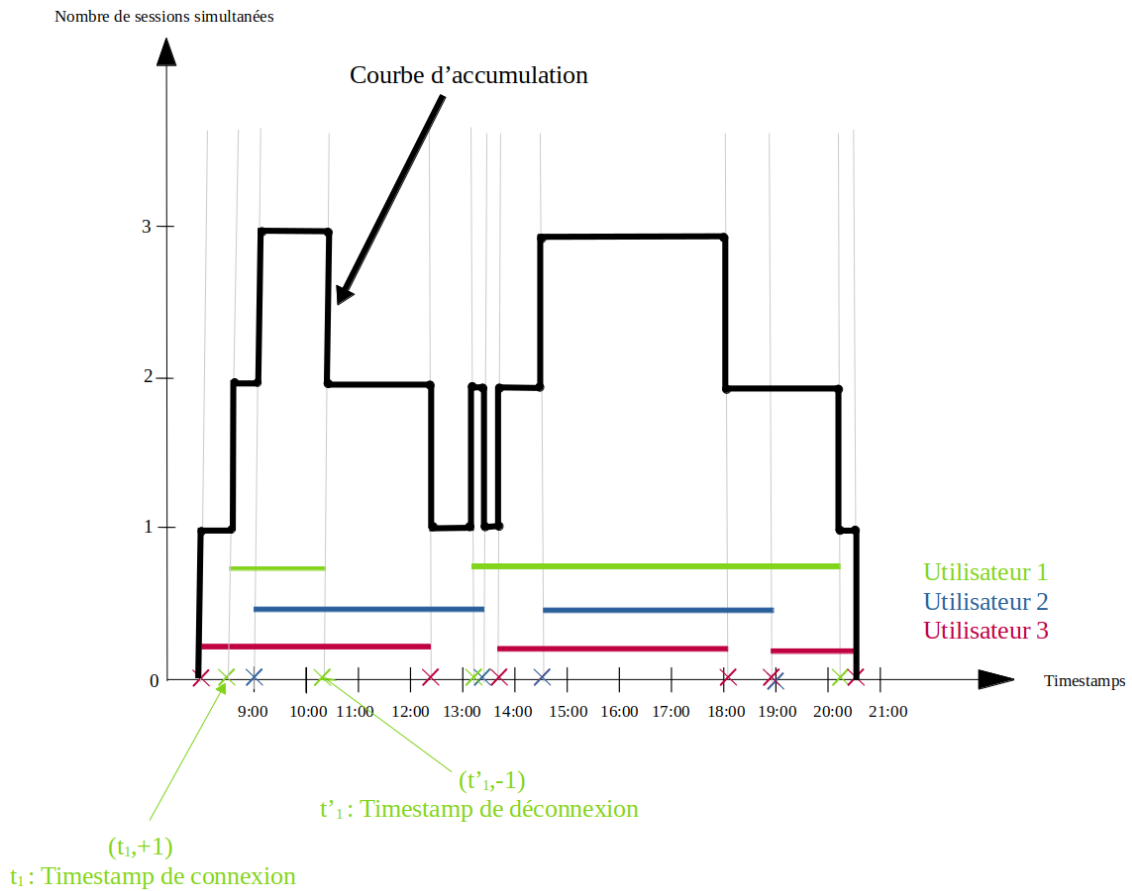


FIGURE 6 – Exemple de courbe d'accumulation pour avoir le nombre de sessions simultanées

Ce Graphique illustre le nombre de sessions simultanées sur une période donnée. Il était essentiel pour aider les entreprises à choisir la licence appropriée en fonction de leurs besoins spécifiques.

Travail accompli, il me restait à finaliser ma fonction pour répondre à la fonctionnalité demandée. Pour ce faire, je devais déterminer l'évolution du nombre maximum de sessions simultanées pour une période donnée en utilisant les données d'entrée fournies : la date de début souhaitée, la date de fin souhaitée et l'intervalle (δx) , (δx) représentant l'intervalle sur lequel on cherche le maximum de sessions simultanées. Le résultat était ensuite renvoyé sous la forme d'un graphique illustrant le nombre maximal de sessions simultanées pour une période définie par un début et une fin, ainsi que la valeur de (δx) .

Pour aborder l'implémentation de l'outil, j'ai créé une fonction qui simule des sessions en générant aléatoirement des heures et des minutes selon une distribution uniforme de 0 à 2 pour les heures et de 0 à 59 pour les minutes :

```
begin date 2023-01-01 00:16:00+00:00    end date 2023-01-01 01:29:00+00:00
begin date 2023-01-01 00:38:00+00:00    end date 2023-01-01 03:28:00+00:00
begin date 2023-01-01 00:49:00+00:00    end date 2023-01-01 01:08:00+00:00
begin date 2023-01-01 02:24:00+00:00    end date 2023-01-01 04:40:00+00:00
begin date 2023-01-01 02:58:00+00:00    end date 2023-01-01 04:41:00+00:00
begin date 2023-01-02 00:58:00+00:00    end date 2023-01-02 03:32:00+00:00
begin date 2023-01-02 01:43:00+00:00    end date 2023-01-02 04:18:00+00:00
begin date 2023-01-02 04:27:00+00:00    end date 2023-01-02 06:06:00+00:00
```

FIGURE 7 – Simulations des sessions

Après avoir appliqué la simulation des sessions, j'ai pu remplir la base de données de PROVE IT afin de tester l'outil. Pour l'exemple de la figure 6, les entrées de l'outil sont les suivantes : 01/01/23, 10/01/23 et $\delta x = 24$ heures (une journée).

On obtient le graphe suivant :

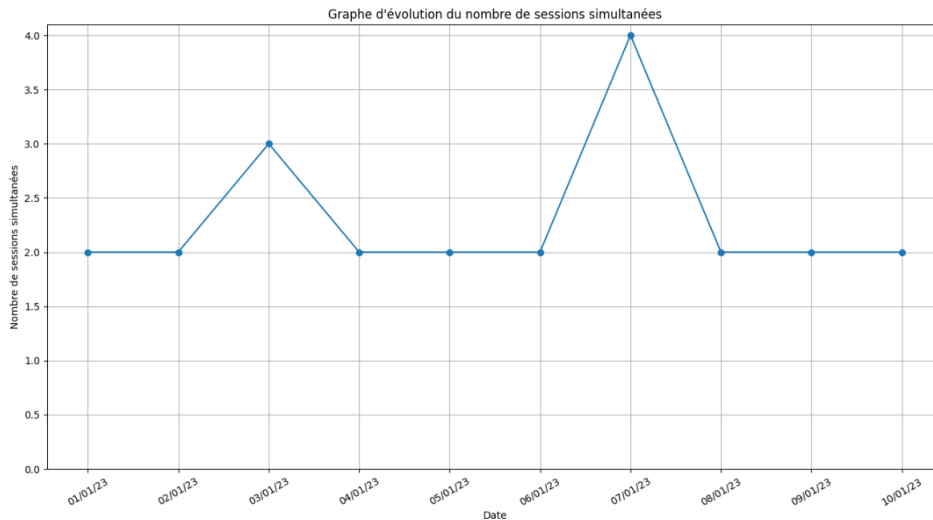


FIGURE 8 – Graphe d'évolution du nombre de sessions simultanées

Qui illustre bien l'évolution du nombre maximal de sessions simultanées pour l'exemple de la figure 6.

Pour m'assurer de la fiabilité de ma fonction, j'ai effectué des tests en utilisant le package pytest de Python. Une fois convaincu de son bon fonctionnement, j'ai soumis mon travail sur le GitLab de l'entreprise, utilisant le gestionnaire de version Git pour rendre mon code accessible et partageable avec l'équipe.

4.2 Deuxième outil

Pour le deuxième outil recherché, l'objectif était clair : mettre en place une revue des droits d'accès au sein de la plateforme. Cette procédure, cruciale dans la gestion de la sécurité, consistait à évaluer les privilèges et les autorisations attribués aux utilisateurs. L'enjeu principal était de s'assurer que seules les personnes ayant les droits appropriés puissent accéder aux données sensibles et aux ressources essentielles.

Pour mener à bien cette mission, la première étape était de localiser les données nécessaires au sein du code de la plateforme PROVE IT. Dans cette quête, une collaboration étroite avec les experts s'avérait nécessaire. Moi, Jonathan, Anthony DAVID et Maxime Jonckiere (un développeur full-stack), explorions ensemble l'architecture complexe de la base de données et identifions le précieux emplacement où résidaient les informations requises.

Cependant, avant d'utiliser la solution PROVE IT, il a fallu configurer la relation entre utilisateurs et services dans une politique d'accès pour garantir son fonctionnement. Pour clarifier les choses, prenons un exemple illustratif : imaginons un royaume (population), appelée : défaut, possède des utilisateurs ayant accès à différents groupes de services selon une politique d'accès définie. Pour mieux visualiser cette idée, voici un schéma illustratif :

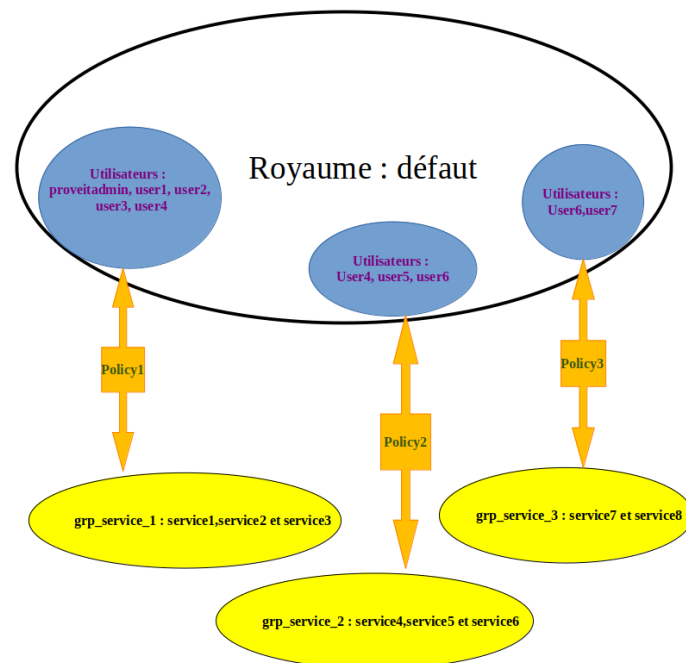


FIGURE 9 – Schéma illustratif

Fort de ces informations, je me suis lancé dans le développement de la fonction de revue des droits d'accès. Guidé par les retours de mes collègues, j'ai mis en place une séquence d'instructions permettant d'effectuer des jointures entre les tables de la base de données. J'ai également utilisé des filtres en SQL, en faisant appel à l'outil SQLAlchemy pour garantir une manipulation précise des données. À l'issue de ce processus ingénieux, une fonctionnalité a émergé, capable de générer des rapports détaillés.

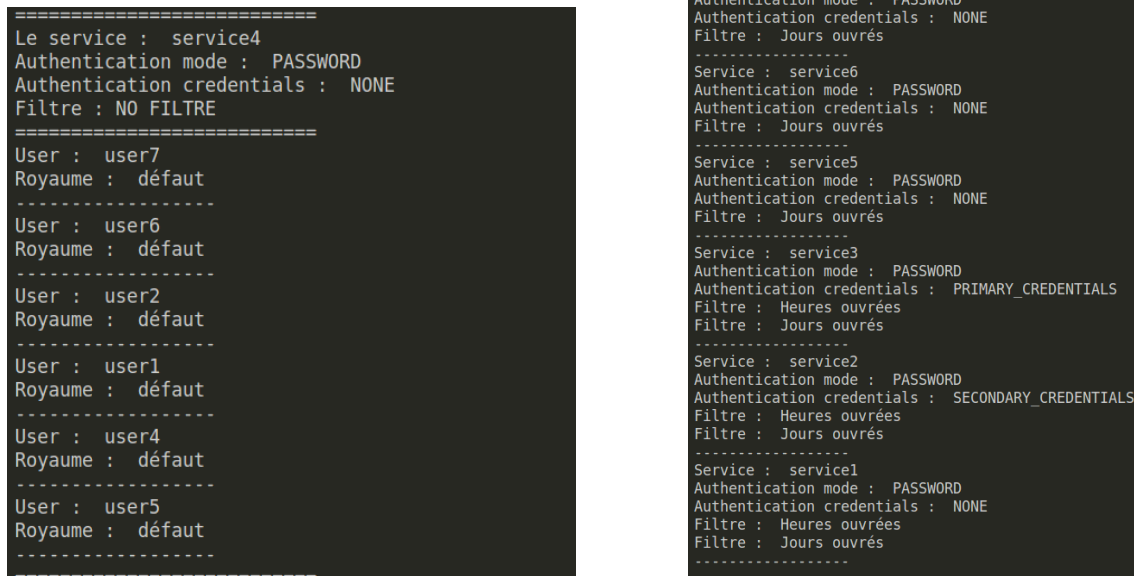
Pour vérifier s'il y a une bonne communication de SQLAlchemy avec le portail d'accès web, j'ai créé une fonction capable de mettre à plat toutes les informations en lien avec l'architecture des utilisateurs et des services de PROVE IT :

```
Les services disponibles sont les suivants :_____
> Service : service4
> Service : service6
> Service : service7
> Service : service8
> Service : service3
> Service : service2
> Service : service1
> Service : service5
Les royaumes disponibles sont les suivants :_____
> Royaume : défaut ---- Royaume Id : 1
Les utilisateurs disponibles sont les suivants :_____
> User : user5 ---- Royaume Id : 1
> User : proveitadmin ---- Royaume Id : 1
> User : user2 ---- Royaume Id : 1
> User : user7 ---- Royaume Id : 1
> User : user3 ---- Royaume Id : 1
> User : user1 ---- Royaume Id : 1
> User : user6 ---- Royaume Id : 1
> User : user4 ---- Royaume Id : 1
Les serveurs disponibles sont les suivants :_____
> Server : proveit-interne
Les politiques disponibles sont les suivantes :_____
> Policy : Policy1
> Policy : Policy2
> Policy : Policy3
```

FIGURE 10 – Sorti de la fonction qui met à plat toutes les informations.

Ainsi, on constate que toutes les informations de la figure 8 sont bien présentes.

Pour réussir l'implémentation d'une fonction qui nous permet d'avoir la revue des droits d'accès, il fallait créer une fonction qui, pour chaque utilisateur, nous donne tous les services auxquels il a accès avec les informations pertinentes (filtres, mode d'authentification). De même, pour un service donné, il fallait identifier tous les utilisateurs qui ont accès à ce service et obtenir les détails associés :



```

=====
Le service : service4
Authentication mode : PASSWORD
Authentication credentials : NONE
Filtre : NO FILTRE
=====
User : user7
Royaume : défaut
-----
User : user6
Royaume : défaut
-----
User : user2
Royaume : défaut
-----
User : user1
Royaume : défaut
-----
User : user4
Royaume : défaut
-----
User : user5
Royaume : défaut
-----
=====

=====
L'utilisateur : user1
Royaume : défaut
=====
Service : service4
Authentication mode : PASSWORD
Authentication credentials : NONE
Filtre : Jours ouvrés
-----
Service : service6
Authentication mode : PASSWORD
Authentication credentials : NONE
Filtre : Jours ouvrés
-----
Service : service5
Authentication mode : PASSWORD
Authentication credentials : NONE
Filtre : Jours ouvrés
-----
Service : service3
Authentication mode : PASSWORD
Authentication credentials : PRIMARY_CREDENTIALS
Filtre : Heures ouvrées
Filtre : Jours ouvrés
-----
Service : service2
Authentication mode : PASSWORD
Authentication credentials : SECONDARY_CREDENTIALS
Filtre : Heures ouvrées
Filtre : Jours ouvrés
-----
Service : service1
Authentication mode : PASSWORD
Authentication credentials : NONE
Filtre : Heures ouvrées
Filtre : Jours ouvrés
-----
=====

```

FIGURE 11 – Sortis des fonctions pour mettre à plat les utilisateurs pour un service et vice versa.

Dans cet exemple, on remarque que le service4 est accessible par les utilisateurs appropriés, et inversement, user1 a accès aux services adéquats. Tout est en conformité avec la relation établie entre les utilisateurs et les services, illustrant ainsi la politique d'accès de l'exemple présenté dans la figure 8.

étapes : Un état d'art, suivi de la conception d'un prototype.

5.1 État de l'art

Dans un premier temps, pour faire un état d'art, j'ai entrepris une recherche des solutions disponibles sur le marché. J'ai découvert diverses approches telles que :

1. SIEM (Security Information and Event Management) : cette méthode collecte des données à partir de sources multiples et utilise des règles de corrélation pour repérer les menaces de sécurité.

— Avantages : Centralisation et corrélation des données de sécurité.

— Inconvénients : Peut nécessiter une quantité importante de stockage.

2. XDR/EDR (Extended Detection and Response)/(Endpoint Detection and Response) : reposant sur les événements d'activité des points d'extrémité généralement stockés.

— Avantages : Permet la détection en temps réel des menaces.

— Inconvénients : Peut engendrer des alertes fausses positives.

3. UBA (User Behavior Analytics) : une approche scientifique générique pour analyser le comportement des utilisateurs.

— Avantages : Adaptable aux nouvelles données et réduit les faux positifs.

— Inconvénients : Nécessite un accès à une variété étendue de données.

Puisque l'UBA semblait prometteur pour réduire les faux positifs, j'ai entrepris de créer un prototype basé sur cette approche. J'ai effectué une revue de la littérature et examiné plusieurs articles scientifiques qui proposaient diverses approches. En m'inspirant de ces travaux, j'ai choisi de m'appuyer sur la thèse de Balaram Sharma intitulée "User Behavior Modeling and Anomaly Detection in Cybersecurity Data Using Deep Learning", car il est le papier le plus détaillé et explique bien les étapes, avec un score de précision d'environ 90%.

À l'heure actuelle, les menaces de cybersécurité se multiplient, en particulier celles provenant d'employés internes, intentionnellement ou non. Les hackers et les individus malveillants peuvent compromettre les comptes d'utilisateurs disposant

de privilèges au sein d'une entreprise, ce qui peut entraîner des actions irréversibles telles que le partage d'informations sensibles ou la sabotage de données. C'est pourquoi la thèse propose une approche basée sur l'UBA (User Behavior Analysis) qui consiste à modéliser les comportements des utilisateurs et à détecter, grâce à l'apprentissage automatique, les comportements anormaux pouvant représenter des menaces potentielles pour la sécurité.

La thèse présente un modèle appelé RNN LSTM Auto-encoder. Permettez-moi d'expliquer ces trois termes :

- RNN (Recurrent Neural Network) : il s'agit d'un type de réseau de neurones récurrents capable de traiter des séquences de données, telles que des séquences temporelles. Dans le contexte de la détection des menaces de sécurité, l'utilisation d'un RNN permet de prendre en compte la séquentialité des comportements des utilisateurs.

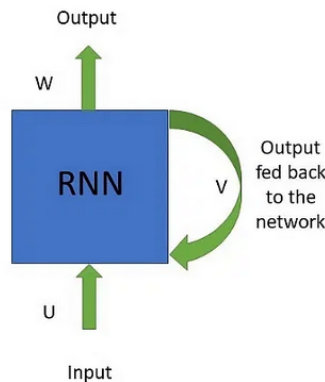


FIGURE 14 – Schéma de RNN

- LSTM (Long Short-Term Memory) : il s'agit d'une variante spécifique des réseaux de neurones récurrents conçue pour capturer et mémoriser des dépendances à long terme dans les séquences de données. Les LSTM sont couramment utilisés dans des tâches de traitement du langage naturel et de séquences temporelles, car ils permettent une gestion efficace des informations à long terme.

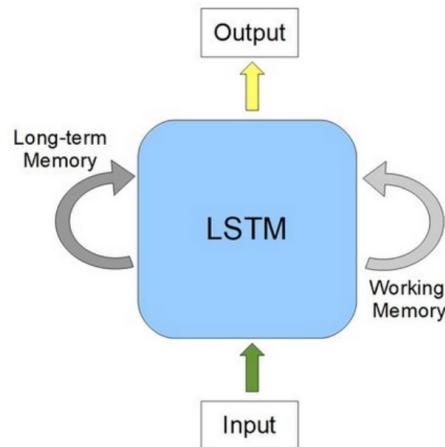


FIGURE 15 – Schéma d'un LSTM

- Auto-encodeur : il s'agit d'une architecture de réseau de neurones utilisée pour apprendre une représentation compressée des données en se basant sur une structure de codage-décodage. Dans le contexte de la détection des menaces de sécurité, l'utilisation d'un auto-encodeur permet de réduire la dimensionnalité des données et d'extraire les caractéristiques les plus pertinentes pour la détection des comportements anormaux.

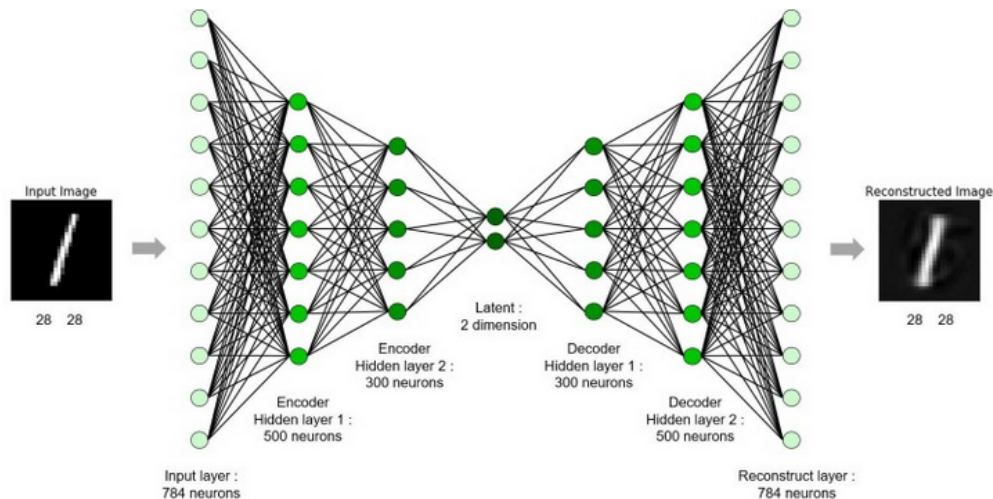


FIGURE 16 – Architecture d'un Auto-encodeur

Ainsi, le modèle proposé dans le papier de thèse utilise une combinaison de RNN LSTM et d'un auto-encodeur pour modéliser les comportements des utilisateurs et détecter les anomalies dans les données de cybersécurité.

Le papier propose également un jeu de données appelé "insider threat Test dataset" ou "CERT Dataset", qui est une base de données générée par l'université de Carnegie Mellon. C'est la référence en ce qui concerne les recherches en détection d'anomalies dans le domaine de la cybersécurité.

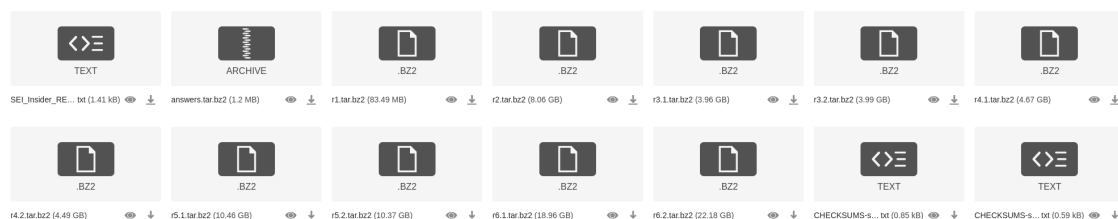


FIGURE 17 – Les fichiers dans CERT Dataset

On peut constater que le jeu de données dispose de plusieurs versions, de r1 à r6.2, ainsi que d'un fichier "answers.csv" qui contient tous les comportements anor-

5.1 État de l'art de la création de modèle pour la détection de menace avec le Machine Learning

maux.

Nous choisirons la version r4.2, qui contient le plus grand nombre de comportements anormaux. Le jeu de données CERT (version r4.2) est composé de six fichiers CSV : device.csv, logon.csv, http.csv, email.csv et file.csv. Ces fichiers contiennent les journaux (logs) de 1000 utilisateurs suivis pendant 500 jours.

Les informations concernant chaque fichier du jeu de données CERT sont les suivantes :

device : Les événements liés à l'accès aux appareils.

1	id	date	user	pc	activity
2	{I5K5-M8HD47HI-6694G0IH}	01/02/2010 07:13:39	NLR0174	PC-8272	Connect
3	{F7F1-T1QC34XE-2487PQDL}	01/02/2010 07:42:43	JZA0447	PC-9753	Connect

http : Les URL visitées par chaque employé.

1	id	date	user	pc	url
2	{U0G4-P5TW04E0-9366IUQF}	01/02/2010 06:36:20	EBH0519	PC-9573	http://usbank.com/1930_FIFA_World_C
3	{G0R5-U4RX40DR-3710WPUY}	01/02/2010 06:39:22	EBH0519	PC-9573	http://forbes.com/Chorioactis/urnul

logon : Les enregistrements de connexions/déconnexions de chaque employé.

1	id	date	user	pc	activity
2	{S7X6-Z2CH44WU-9256VRTB}	01/02/2010 06:34:00	EBH0519	PC-9573	Logon
3	{M5W6-F9WM86X0-3622DJWW}	01/02/2010 06:46:00	LAD0514	PC-2567	Logon

email : Les informations de chaque transfert d'e-mail entre les employés.

1	id	date	to	from
2	{X5L7-D1UH90YS-1333JYFH}	01/02/2010 07:14:06	Elmo.Brendan.Haynes@dtaa.com	Yeo.Xyla.Garn
3	{B5B5-M3DV33QF-7853QPVE}	01/02/2010 07:19:11	willow.Winifred.Mcgowan@dtaa.com	Fritz.Buckmin

file : Contient les accès aux fichiers par les employés.

1	id	date	user	pc	filename	content
2	{M2T9-E1GX15KS-8671LZJY}	01/02/2010 07:59:14	YXG0504	PC-4312	ZZEZ8RF0.doc	D0-CF-11-E0-A1-B1-1A-E
3	{J5V3-T2QZ50VE-3234KKVL}	01/02/2010 08:01:13	WHC0684	PC-3207	UEDJK4XP.doc	D0-CF-11-E0-A1-B1-1A-E

Il existe trois scénarios de comportement anormales :

1. Un utilisateur qui n'avait jamais utilisé de supports amovibles ni travaillé en dehors des heures de travail commence à se connecter après les heures de bureau, en utilisant une clé USB, et à télécharger des données sur wikileaks.org. Peu de temps

après, il quitte l'organisation.

2. Un utilisateur se connecte sur la machine d'un autre utilisateur et recherche des fichiers intéressants qu'il envoie à son adresse e-mail personnelle. Ce comportement devient de plus en plus fréquent sur une période de 3 mois.

3. L'administrateur système devient mécontent. Il télécharge un enregistreur et utilise une clé USB pour le transférer sur l'ordinateur de son supérieur. Le lendemain, il utilise les enregistrements collectés pour se connecter en tant que son supérieur et envoyer un e-mail de masse alarmant, provoquant la panique dans l'organisation. Il quitte immédiatement l'organisation.

Ces scénarios sont traduits par des experts en cybersécurité vers des logs qui sont injectés manuellement dans les cinq fichiers csv.

Le dossier "answers.csv" contient tous les événements anormaux, plus précisément les logs de chaque version de la base de données et de chaque utilisateurs.

5.2 Création d'un prototype

Pour la création d'un prototype, il faut passer par 4 étapes :

5.2.1 Prétraitement des données

Cela consiste à agréger tous les fichiers, dans mon cas seulement deux : "device.csv" et "logon.csv", car ils contiennent les informations les plus proches des données d'audit de PROVE IT. Ensuite, les données sont ordonnées chronologiquement et je sépare les données de chaque utilisateur afin d'obtenir un fichier propre contenant les logs de chaque utilisateur.

5.2.2 Ingénierie des caractéristiques

À cette étape, j'ai choisi 11 variables (features) en référence à des articles scientifiques ([1], [9], [10], [11]) qui permettent de modéliser au mieux le comportement d'un utilisateur. Ces variables sont les suivantes :

- nb_logon : Nombre de connexions de l'utilisateur dans la journée.
- nb_logoff : Nombre de déconnexions de l'utilisateur dans la journée.
- nb_conx : Nombre total de connexions dans la journée.
- nb_dconx : Nombre total de déconnexions dans la journée.
- nb_sec_conx : Durée totale de connexion en secondes dans la journée.

- nb_diff_pc : Nombre de connexions sur des ordinateurs différents dans la journée.
- nb_conx_after_hour : Nombre de connexions en dehors des heures de travail dans la journée.
- nb_events : Nombre total d'événements dans la journée.
- is_weekend : Un booléen pour indiquer si le jour est un week-end ou non.
- day_of_month : Le jour du mois correspondant à la journée.
- num_month : Le mois correspondant sous forme de numéro.

Ensuite, une table appelée "table of feature vectors" est créée, notée X. Cette table contient les 11 variables en tant que colonnes et chaque ligne représente une journée correspondante. **Pour créer une ligne de cette table, on regroupe tous les logs de la journée et on calcule le nombre d'occurrences de chaque variable.**

On obtient :

date	nb_logon	nb_logoff	nb_conx	nb_dconx	nb_sec_conx
2010-01-04	8	7	7	7	83616
2010-01-05	7	6	7	7	75522
2010-01-06	3	3	2	2	37931
2010-01-07	6	5	6	6	70383
2010-01-08	5	4	5	5	75700

FIGURE 18 – Table of feature vectors

5.2.3 Construction et entraînement du modèle

J'ai créé le modèle RNN LSTM Auto-Encoder suivant :

```
# Parametres
timesteps=1
n_features=X_train_normal.shape[2]
lr = 0.000001

# define model
model = Sequential()
model.add(LSTM(8, activation='relu', input_shape=(timesteps,n_features), return_sequences=True))
model.add(LSTM(6, activation='relu', input_shape=(timesteps,n_features), return_sequences=True))
model.add(LSTM(4, activation='relu', return_sequences=False))
model.add(RepeatVector(timesteps))
model.add(LSTM(4, activation='relu', return_sequences=True))
model.add(LSTM(6, activation='relu', return_sequences=True))
model.add(LSTM(8, activation='relu', return_sequences=True))
model.add(TimeDistributed(Dense(n_features)))
adam = optimizers.Adam(lr)
model.compile(optimizer='adam', loss='mse')
model.summary()
```

FIGURE 19 – Code du modèle RNN LSTM Auto-encoder

Il s'agit d'une architecture de réseau de neurones composée d'un encodeur qui réduit l'information de 8 neurones à 4 neurones, suivi d'un décodeur qui augmente l'information de 4 neurones à 8 neurones pour reproduire l'entrée du modèle. **On mesure l'erreur de reconstruction en soustrayant l'entrée reconstruite de l'entrée originale.** La fonction d'activation utilisée est ReLU, le taux d'apprentissage (learning rate) est fixé à 0.000001, l'optimiseur est Adam et la fonction de perte (loss) est définie comme "mse" (erreur quadratique moyenne).

Pour détecter les activités anormales, dans notre cas les jours anormaux, nous nous basons sur l'erreur de reconstruction. **Un comportement est considéré comme anormal si son erreur de reconstruction est supérieure à un seuil, le seuil est calculé dans notre cas en prenant la moyenne et l'écart type de l'erreur de reconstruction.**

Il existe d'autres méthodes pour calculer ce seuil. Par exemple, on peut faire varier le seuil et tracer les courbes du TNR (True Negative Rate) et du TPR (True Positive Rate), puis choisir le point d'intersection des deux courbes comme seuil. Cette approche est référencée dans le papier de thèse.

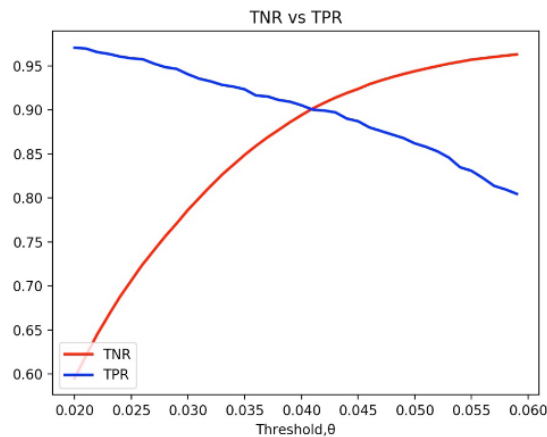


FIGURE 20 – Méthode pour calculer le seuil

Voici un exemple d'erreur de reconstruction :

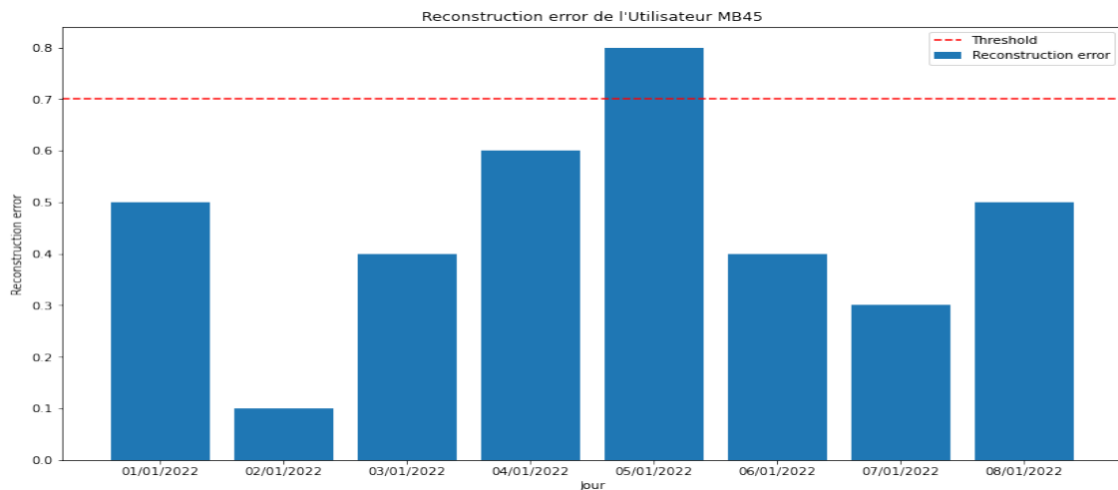


FIGURE 21 – Mesure de l'erreur de reconstruction de MB45

On constate que l'erreur de reconstruction de l'utilisateur MB45, pour le 5 janvier 2022, est supérieure au seuil (Threshold) de 0.7. Ainsi, nous pouvons conclure

que c'est un jour anormal, ce qui indique un comportement anormal et une éventuelle menace pour la sécurité.

Pour entraîner le modèle, j'ai créé deux tables de vecteurs de caractéristiques. La première table, nommée X_{Normal} , ne contient que des jours normaux, c'est-à-dire des logs de connexion normaux. La deuxième table, nommée X , contient à la fois des jours normaux et des jours anormaux. **Un jour est considéré comme anormal s'il contient au moins un seul log anormal, et nous considérons un comportement comme anormal s'il y a au moins un jour anormal.**

Nous divisons les échantillons X^{Normal} et X en trois ensembles distincts : entraînement, validation et test. Pour entraîner le modèle, nous utilisons l'ensemble X_{Train}^{Normal} et $X_{Validation}$, et pour tester le modèle, nous utilisons l'ensemble X_{Test} .

5.2.4 Évaluation

Dans cette dernière partie, j'ai évalué les performances du modèle pour deux utilisateurs en utilisant différentes metriques telles que la courbe ROC et la matrice de confusion.

Pour l'utilisateur MOS0047, voici la courbe ROC obtenue :

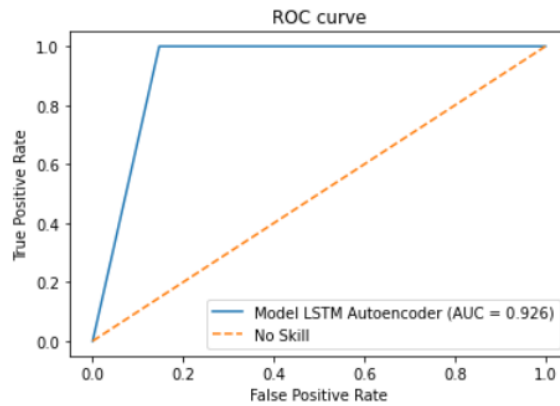


FIGURE 22 – Courbe ROC de MOS0047

On constate que l'AUC (aire sous la courbe) est de 0.926, ce qui est proche de 1. Cela indique que le modèle est un bon classifieur.

Pour la matrice de confusion :

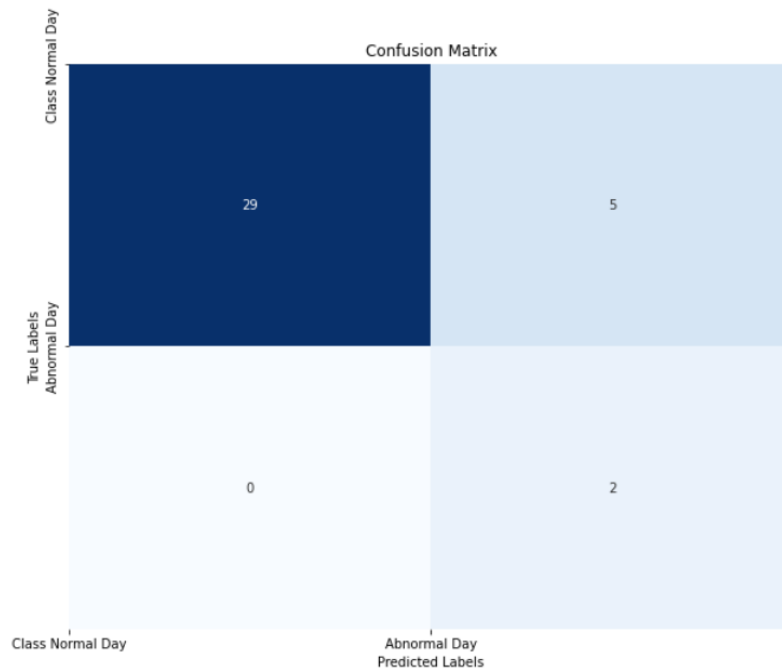


FIGURE 23 – Matrice de confusion de MOS0047

Sur l'échantillon X_{Test} , qui est constitué de 36 jours dont 2 jours sont anormaux, le modèle a réussi à détecter les deux jours anormaux correctement et a prédit 29 jours normaux de manière précise, ce qui donne une précision de 0.86.

Pour l'utilisateur KRL0501, voici la courbe ROC obtenue :

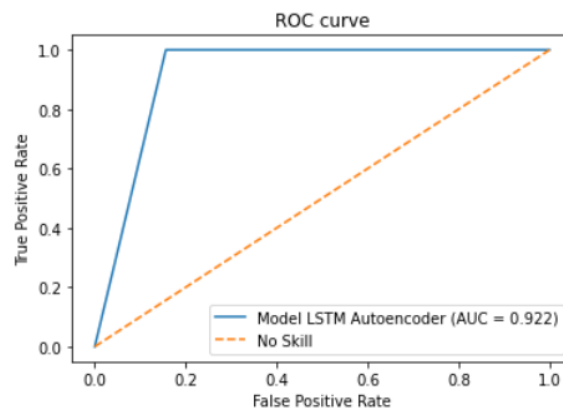


FIGURE 24 – Courbe ROC de KRL0501

On observe une AUC de 0.922, également proche de 1, ce qui indique que le modèle est un bon classifieur.

Pour la matrice de confusion :

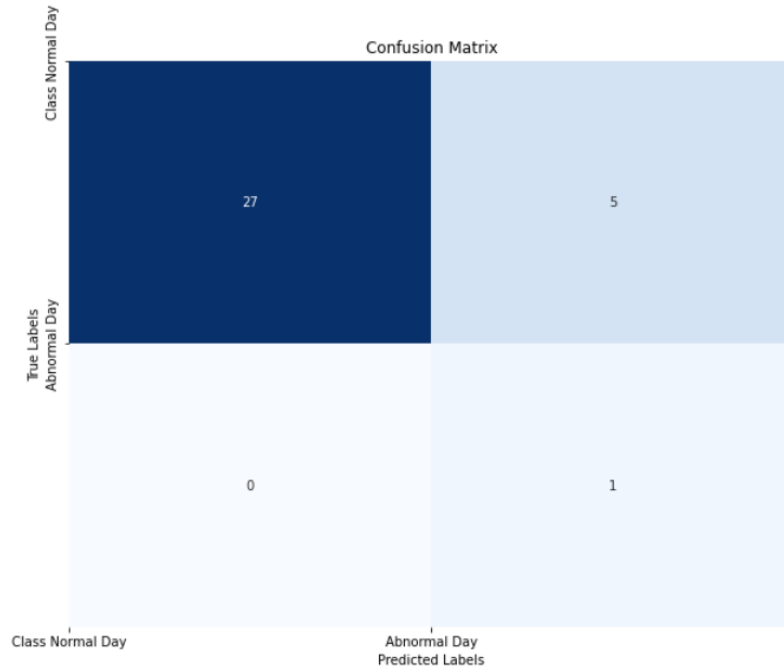


FIGURE 25 – Matrice de confusion de KRL0501

Sur l'échantillon X_{Test} , qui est constitué de 33 jours dont 1 jour anormal, le modèle a réussi à détecter correctement le jour anormal et a prédit avec précision 27 jours normaux, ce qui donne une précision de 0.84.

5.3 Autre méthode de détection d'anomalies

J'ai également essayé d'autres méthodes de détection d'anomalies, telles que l'Isolation Forest, mais sans succès.

5.4 Présentation et documentation

Après avoir terminé la création de mon prototype, j'ai commencé à rédiger la documentation de mon travail. Ensuite, j'ai préparé une présentation dans laquelle j'ai partagé mes résultats avec l'ensemble de l'équipe de l'entreprise. Il y a eu beaucoup de questions et de discussions après de cette présentation, principalement axées sur le modèle que j'ai développé.

5.5 Création d'un script d'extraction des données de PROVE IT

Après avoir appliqué le modèle au jeu de données CERT, j'ai porté mon attention sur les données de PROVE IT. Pour y accéder, j'ai initié le transfert des informations de la base de données depuis la machine virtuelle où elles étaient initialement stockées vers mon ordinateur, en utilisant le protocole de copie sécurisée (SCP). Par la suite, j'ai extrait les sessions contenant les informations pertinentes, lesquelles étaient au format JSON. J'ai procédé à la transformation de ces informations en lignes de tableau en utilisant la bibliothèque pandas.

	sessionId	user_name	user_ip	service_name	date_start	duration	date_finish	downloadDataSize	uploadDataSize
0	4dcee4b9-ccf8-4555-bab2-fc6d4bf0f818	user2	192.168.122.167	service2	2023-06-16 16:10:13	1686.372	2023-06-16 16:38:19	10294	0
1	49c8ec5f-9bf0-4160-920d-8de6f8691933	user3	192.168.122.167	service2	2023-06-16 16:06:23	3605.363	2023-06-16 17:06:28	20534	0
2	f5b84fc4-cc33-4803-be0f-280fc075cb06	user2	192.168.122.167	service2	2023-06-16 14:33:16	564.134	2023-06-16 14:42:40	13568	524
3	3fa41eda-a488-4f9a-a919-929d52072527	user1	192.168.122.167	service2	2023-06-16 16:05:43	1935.494	2023-06-16 16:37:59	11662	1
4	0653152d-cf57-4eb3-8a08-bf126a529e38	user1	192.168.122.167	service2	2023-06-16 14:39:20	299.795	2023-06-16 14:44:20	10120	372
5	20a0b7a8-0b1a-4dc6-9e37-5de7b6982069	user4	192.168.122.167	service2	2023-06-16 14:45:16	4042.529	2023-06-16 15:52:38	34390	709
6	2f89715b-cc59-4809-90e9-e6049960f493	user4	192.168.122.167	service2	2023-06-16 16:06:43	3605.605	2023-06-16 17:06:48	20534	0
7	60b7eca5-9e1f-42a8-a01c-87acecdebdbbe	user1	192.168.122.167	service2	2023-06-16 17:37:32	104.100	2023-06-16 17:39:16	5956	97
8	bfc2a7c-a9b7-47b4-9988-493a3ee977b2	proveitadmin	192.168.122.167	service2	2023-06-16 16:07:37	3606.008	2023-06-16 17:07:43	20534	0
9	ec5bdea8-b610-4a08-9d41-b76d60e4e6f8	user2	192.168.122.167	service2	2023-06-16 16:06:04	3609.395	2023-06-16 17:06:13	20534	0
10	cded1476-4202-44ba-870c-2dfc2f2fc171	user2	192.168.122.167	service2	2023-06-16 14:35:40	459.510	2023-06-16 14:43:20	3582	1
11	330746af-183f-4e62-a84a-6509d7948903	user3	192.168.122.167	service2	2023-06-16 16:10:29	1667.967	2023-06-16 16:38:17	10241	5

FIGURE 26 – Les sessions en DataFrame

6 Conclusion

Au terme de mon stage au sein de l'entreprise de cybersécurité, il est clair que cette expérience m'a permis de plonger dans un domaine totalement nouveau pour moi. Initialement, j'ai été confronté à des difficultés d'adaptation et de compréhension des concepts complexes de la cybersécurité. Toutefois, ma détermination inébranlable, ma capacité d'adaptation et mon désir ardent d'apprendre m'ont permis de progresser de manière significative.

L'un des aspects les plus marquants de cette expérience a été ma découverte de plusieurs solutions cyber telles que SIEM, XDR et UBA. Ces découvertes ont enrichi ma connaissance du domaine et m'ont ouvert de nouvelles perspectives sur les pratiques et les outils de la cybersécurité.

En parallèle, je me suis confronté à un manque de compétences dans le monde du développement et de ses outils. Cependant, en documentant mes recherches et en sollicitant l'aide bienveillante de mes collègues, j'ai pu acquérir les compétences nécessaires pour réussir mon stage. Cette période m'a également enseigné l'importance de la communication efficace, de la présentation claire des problèmes et des réussites, ainsi que de la collaboration en équipe.

Une étape importante de mon stage a été le développement de deux outils d'analyse de données, qui m'a permis d'explorer le data mining et ses multiples applications pour l'entreprise. Ces outils ont prouvé leur utilité en fournissant des statistiques sur l'activité des employés et en facilitant la gestion des habilitations.

Mon stage m'a également initié au monde du travail en me faisant découvrir ses règles et ses droits, ainsi que le monde du développement avec ses divers outils de communication tels que Git et GitHub.

De plus, j'ai eu l'occasion de m'immerger dans le domaine de la recherche en effectuant des revues scientifiques approfondies. Cela m'a permis de suivre les dernières avancées en matière de modèles de machine learning pour la détection des menaces de sécurité. J'ai particulièrement été captivé par la détection d'anomalies, et j'ai eu la chance de mettre en œuvre des algorithmes tels que l'autoencodeur, l'Isolation Forest et l'One-Class SVM.

Enfin, ce stage m'a offert l'opportunité de maîtriser le langage Python, différents frameworks tels que l'ORM SQLAlchemy, ainsi que des notions essentielles en réseau telles que SSH et les ports TCP.

En conclusion, cette expérience a été un véritable voyage d'apprentissage, où

j'ai surmonté des défis, élargi mes compétences techniques et professionnelles, et découvert un nouvel univers fascinant. Je suis reconnaissant envers mes collègues, mes mentors et l'entreprise pour leur soutien et leur encouragement tout au long de cette période de croissance personnelle et professionnelle.

7 Références

Références

- [1] Balaram Sharma : *User Behavior Modeling and Anomaly Detection in Cyber-security Data Using Deep Learning.*
- [2] Chitta Ranjan : *Extreme Rare Event Classification using Autoencoders in Keras.*
- [3] Chitta Ranjan : *LSTM Autoencoder for Extreme Rare Event Classification in Keras.*
- [4] Chitta Ranjan : *Step-by-step understanding LSTM Autoencoder layers.*
- [5] David Woroniuk : *Outlier Detection with RNN Autoencoders.*
- [6] Jason Brownlee : *A Gentle Introduction to LSTM Autoencoders.*
- [7] Sam Black : *Using LSTM Autoencoders on multidimensional time-series data.*
- [8] Jason Brownlee : *Multivariate Time Series Forecasting with LSTMs in Keras.*
- [9] Bushra Bin Sarhan, Najwa Altwaijry : *Insider Threat Detection Using Machine Learning Approach.*
- [10] Junhong Kim, Minsik Park , Haedong Kim, Suhyoun Cho, Pilsung Kang : *Insider Threat Detection Based on User Behavior Modeling and Anomaly Detection Algorithms.*
- [11] Dennis Chow : *Insider Threat Detection with AI Using Tensorflow and Rapid-Miner Studio.*
- [12] Wei Jiang, Yuan Tian, Weixin Liu, Wenmao Liu : *An Insider Threat Detection Method Based on User Behavior Analysis.*
- [13] Sahand Hariri, Matias Carrasco Kind, Robert J. Brunner : *Extended Isolation Forest.*
- [14] Yousra Chabchoub, Maurras Ulbricht Togbe, Aliou Boly, Raja Chiky : *An in-depth study and improvement of Isolation Forest.*
- [15] Fei Tony Liu, Kai Ming Ting, Zhi-Hua Zhou : *Isolation Forest.*
- [16] Carlos Mougan : *Isolation Forest from Scratch.*
- [17] Bibliothèque sklearn : <https://scikit-learn.org/stable/>.