



RAPPORT DE PROJET DE FIN D'ANNÉE SOUS THÈME

Conception et réalisation d'une solution d'évaluation de la qualité d'une Politique de Certificat de clés publique

Réalisé par :

EL YUBI OUMAIMA
RAJOUL ABDELJALIL

Encadré par :

Mme.EL BAKKALI HANANE
Mme.WAHABI ZAKIA

Année Universitaire : 2018-2019

Dédicace

À nos pères à qui nous devons tout.

À nos mères pour leur soutien et leur présence aux moments difficiles.

À nos grands-parents.

À notre encadrant.

À toutes nos familles.

À nos formateurs, et nos enseignants.

À tous nos amis (es).

Remerciement

Nous tenons sincèrement à remercier tous les professeurs de l' ENSIAS et particulièrement Mme. EL BAKKALI Hanane, Mme. WAHABI Zakia et tout ceux qui nous ont aidé pendant la réalisation de notre projet de fin d'année, pour les efforts qu'ils ont fournis afin de perfectionner notre formation et enrichir nos connaissances.

Nous souhaitons que ce travail soit à la hauteur du niveau estimé.

En fin, veuillez accepter, mesdames et messieurs les membres du jury, toutes nos reconnaissances.

Résumé

L'utilisation des infrastructures de clé publique (PKI) et des techniques cryptographiques sous-jacentes est devenue une nécessité pour sécuriser les échanges électroniques sur Internet. Le nombre croissant d'Autorités de Certification (CA : Certification Authorities) approuvées par les navigateurs actuels en est un témoin éloquent. Néanmoins, ces CA n'échappent point aux risques de fraude et autres menaces d'attaque. Donc , l'utilisateur doit évaluer la signature , le chemin de la certification et même la politique de certification qui est appliquée lors de l'émission du certificat . C'est dans ce cadre que s'inscrit notre projet qui permet de développer un outil pour la formalisation des CP afin de faciliter leur évaluation.

KEY WORDS :

Infrastructure de clé publique (PKI) – Autorités de Certification (CA) – Signature – Politique de certification (CP) – Certificat.

Abstract

The growing number of Certificate Authorities (CAs) assigned to the Public Key Infrastructure (PKI) and approved by current browsers poses serious security concerns. The user must therefore evaluate the signature, the certification path and even the certification policy that is applied when issuing the certificate. Our project makes it possible to develop a tool for the formalization of the CP in order to facilitate their evaluation.

KEY WORDS :

Certificate Authorities (CA) – Public Key Infrastructure (PKI) – Certification – Certification Policy (CP) – Signature.

Liste des abréviations

AC	Autorité de certification
CPS	La déclaration de pratique de certification
CRL	Liste de révocation de certificats
LSL	Les niveaux de sécurité locaux
PKC	Certificat de clé publique
PKI	Public Key Infrastructure
PMC	Perceptron multicouche
TLS	Transport layer structure
PC	Politique de certification
AA	Autorité administrative
ACR	Autorité de certification racine
AE	Autorité d'enregistrement
AH	Autorité d'horodatage
TPC	Tierce Partie de Confiance

Table des figures

1.1	diagramme de GANT	4
2.1	Organisation de la PKI	7
3.1	Structure d'un réseau de neurones artificiels	12
3.2	La régression linear et logistique	13
3.3	La régression linear et logistique	13
3.4	classification de notre résultat	14
3.5	La fonction sigmoid graphique	14
4.1	syntaxe du fichier JSON	16
4.2	PC en JSON format	17
4.3	Pandas logo	18
4.4	NLTK commande d'installation	19
4.5	importation de NLTK	20
4.6	tokénize les champs de la PC	20
4.7	importation de la PC	21
4.8	Browse fichier	21
4.9	champs d'entrée	22
4.10	champs d'entrée	22
4.11	Les fichiers correspondant à chaque champs	23
4.12	Validation de la PC avec NLTK	23
4.13	Vectorization des données	24
4.14	Les fichiers des données	24
4.15	Vectorization des données	25
4.16	Prédiction de notre modèle	25
4.17	Evaluation de chaque champs	26
4.18	Interface d'entrée	26
4.19	Evaluation globale de notre PC	27

Table des matières

Introduction générale	1
1 Contexte général du projet	2
1.1 Introduction	3
1.2 Problématique	3
1.3 Objectifs	3
1.4 Planification	4
2 Généralité sur l'infrastructure de clé publique (PKI)	5
2.1 Définition de la PKI	6
2.2 Organisation d'une PKI	7
2.3 Les acteurs d'une infrastructure de gestion de clés	7
2.4 Les certificats	9
2.5 Les composants d'une PKI	10
3 Introduction aux réseaux de neurones	11
3.1 Réseau de neurone	12
3.1.1 Définition	12
3.1.2 Structure d'un RNA	12
3.1.3 Logistic Regression	12
3.1.4 Fonction sigmoïde (fonction logistique)	13
3.1.5 Limite de décision	14
4 Réalisation d'une solution de formalisation d'une CP	15
4.1 Conception du document PC avec JSON	16
4.1.1 JSON	16
4.1.1.1 Définition	16
4.1.1.2 Les caractéristiques	16
4.1.1.3 Les avantages	16
4.1.1.4 La syntaxe	16
4.1.2 Modèle proposé d'une PC en JSON format	17
4.2 Environnement de développement	17
4.2.1 Language de développement :	17
4.2.2 Pycharm :	17
4.3 Les bibliothèques utilisés	18
4.3.1 Numpy	18
4.3.2 Pandas	18
4.3.3 Tkinter	19
4.3.4 Nltk	19
4.3.4.1 Définition	19
4.3.4.2 installation de NLTK	19

4.3.4.3	Travailler avec NLTK	20
4.4	Réalisation	21
4.4.1	Importation de la politique de sécurité	21
4.4.2	sécurité de champs de l'entrée	21
4.4.2.1	Exécution de code à distant	21
4.4.2.2	inclusion de fichiers locaux	22
4.4.3	Evaluation de PC à l'aide d'une disribution fréquentielle	23
4.4.4	Evaluation de la PC à l'aide d'un algorithme d'apprentissage automatique	24
4.4.4.1	Vectorization des champs	24
4.4.4.2	Vectorization des réponses de chaque champs :	24
4.4.4.3	Apprentissage de notre modèle	25
4.4.4.4	Validation finale de notre PC	26

conclusion		28
-------------------	--	-----------

Introduction générale

De nos jours, les infrastructures à clés publiques (PKI) sont devenues une partie intéressante dans les architectures de sécurité de chaque entreprise. Une PKI se focalise sur de nombreux aspects de gestion de la sécurité, ainsi que le service en tant que facilitateur pour un nombre croissant d'applications de sécurité standard et personnalisées. La plupart des protocoles standards pour l'e-mail sécurisé, l'accès Web, les réseaux privés virtuels et les systèmes d'authentification des utilisateurs à connexion unique utilisent des certificats de clé publique et nécessitent donc une forme de PKI. La compréhension de PKI est devenu un aspect technique très précieux pour tout responsable informatique, CIO, administration de sécurité système ou développeur de protocole d'application vu l'évolution rapide des PKI dans plusieurs entreprises. Dans X.509 PKI, l'autorité de certification (CA) émet un certificat, qui présente une preuve pour confirmer l'identité de son titulaire, en utilisant ses propres règles qui sont définies dans la politique de certification (CP) et la déclaration de pratique de certification (CPS).

Certes, une autorité de certification peut être attaquée ou encore utilisée pour le but de délivrer des certificats frauduleux et une fausse signature. Ainsi, le niveau de confiance de ces certificats se remet en question. En outre, l'utilisateur du certificat (RP : Relying party) devrait vérifier la fiabilité de ce dernier afin de l'accepter ou non. Elle doit vérifier sa signature, son chemin de certification du certificat jusqu'au certificat-racine. De même, elle doit lire le CP / CPS que l'autorité de certification suit pendant le cycle de vie du certificat. Pratiquement, elle leur est difficile d'évaluer le CP / CPS qui est technique et long. Par conséquent, la partie utilisateur a besoin d'un mécanisme automatisé pour prendre une décision de confiance sur un certificat reçu. C'est dans ce cadre que s'inscrit notre projet qui permet de formaliser politique de certificat.

Dans ce rapport nous essayerons d'abord de donner une présentation générale du projet, puis, nous présenterons dans le deuxième chapitre l'infrastructure à clé publique. Ensuite, dans le troisième chapitre, nous donnerons un aperçu général sur les réseaux de neurones. Le quatrième chapitre portera sur la partie réalisation de notre projet, pour terminer par une conclusion et des perspectives.

Chapitre 1

Contexte général du projet

Dans ce premier chapitre nous commençons par présenter quelques grandes notions vues lors de notre documentation sur le sujet, puis nous entamerons la problématique pour arriver enfin à élaborer une description du projet.

1.1 Introduction

Dans ces dernières années l' utilisation des applications e-services (e-commerce , e-gouvernement ,e-health) augmente grâce à la confiance qu' accorde les utilisateurs à celles – ci. Cette confiance est généralement assurée par des technologies de sécurité comme l' utilisation d' une infrastructure PKI .

1.2 Problématique

La PKI permet de propager la confiance entre les bénéficiaires et les offreurs de services au moyen des certificats à clé publique et qui sont émis par une autorité de certification en appliquant les procédures décrites et définies dans une politique de certificat . Or , dans les réseaux ouverts , l' autorité de certification peut être attaquée et utilisée pour délivrer des certificats frauduleux , dans ce cas l' utilisateur(RELYING PARTY) doit vérifier la fiabilité d' une CA avant d' accepter un certificat émis par une autorité de certification . L' utilisateur doit donc évaluer la signature , le chemin de la certification et même la politique de certification qui est appliquée lors de l' émission du certificat . La CP est souvent longue et très technique , d' ou le besoin d' un mécanisme permettant l' automatisation et l' interprétation de celle- ci .

1.3 Objectifs

Les problèmes mentionnés précédemment peuvent être résolus et ceux en commençant tout d'abord par lire et comprendre le document CP dont la structure est conforme à la RFC 3647, ensuite on passe à l'extraction des rubriques nécessaire pour juger une CP pour pouvoir déterminer la fiabilité d'une CA et cela en développant un outils qui permet de formaliser une CP pour que cette dernière devient compréhensible par la machine et donc facile à évaluer .

1.4 Planification

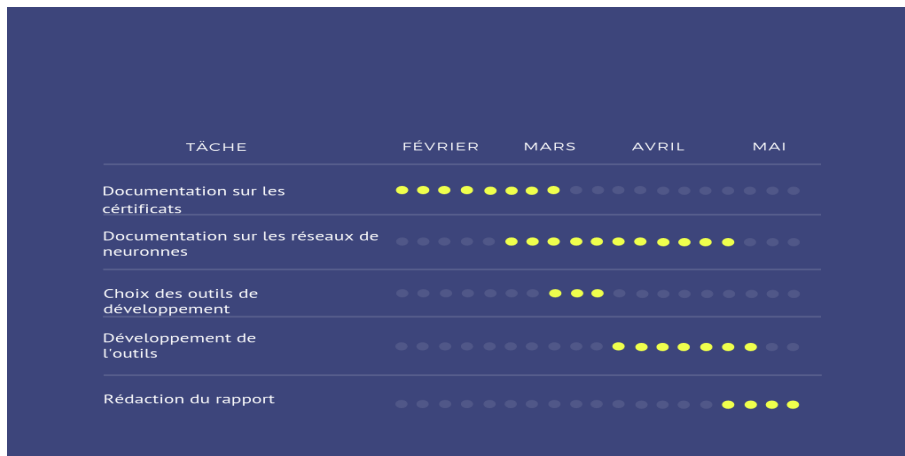


FIGURE 1.1 – diagramme de GANT

Chapitre 2

Généralité sur l'infrastructure de clé publique (PKI)

Dans ce deuxième chapitre nous commençons par présenter des généralités sur l'infrastructure à clé publique , puis nous entamerons le concept , le fonctionnement et les composants du PKI .

2.1 Définition de la PKI

PKI (Public Key Infrastructure) est un système de gestion des clefs publiques qui permet de gérer des listes importantes de clefs publiques et d'en assurer la fiabilité, pour des entités généralement dans un réseau. Elle offre un cadre global permettant d'installer des éléments de sécurité tels que la confidentialité, l'authentification, l'intégrité et la non-répudiation tant au sein de l'entreprise que lors d'échanges d'information avec l'extérieur [1].

Une infrastructure à clé publique utilise des mécanismes de signature et certifie des clés publiques qui permettent de chiffrer et de signer des messages ainsi que des flux de données, afin d'assurer les 7 services de sécurité :

- **Confidentialité** : vous utilisez une infrastructure à clé publique pour crypter les données stockées ou transmises.
- **Intégrité** : une infrastructure à clé publique vous permet de signer numériquement des données. Une signature numérique permet de vérifier qu'aucun utilisateur ou processus n'a modifié les données.
- **Authenticité** : une PKI fournit plusieurs mécanismes d'authenticité ; les données d'authentification passent par des algorithmes de hachage pour produire un résumé du message. L'empreinte du message est ensuite signée numériquement à l'aide de la clé privée de l'expéditeur, prouvant que le résumé du message a été produit par lui (la clé privée est unique).
- **Non-répudiation** : lorsque les données sont signées numériquement, la signature numérique fournit une preuve de l'intégrité des données signées et une preuve de l'origine des données. Un tiers peut alors vérifier l'intégrité et l'origine des données à tout moment. Cette vérification ne peut pas être réfutée par le propriétaire du certificat qui a numériquement signé les données.
- **Disponibilité** : vous pouvez installer plusieurs autorités de certification dans votre hiérarchie de CA pour émettre des certificats. Si une autorité de certification n'est pas disponible dans la hiérarchie de CA, un autre CA peut délivrer un certificat.
- **La traçabilité (ou « preuve »)** : garantie que les accès et tentatives d'accès aux éléments considérés sont tracés et que ces traces sont conservées et exploitables.
- **Le contrôle d'accès** : désigne les différentes solutions techniques qui permettent de sécuriser et gérer les accès physiques à un bâtiment ou un site, ou les accès logiques à un système d'information. On distingue ainsi le contrôle d'accès physique et le contrôle d'accès logique.

2.2 Organisation d'une PKI

Dans une infrastructure à clé publique, et pour obtenir un certificat numérique, l'utilisateur doit tout d'abord faire une demande auprès de l'autorité d'enregistrement (AE). Cette dernière s'occupe de la génération d'un couple de clé (clé publique, clé privée), ensuite, elle envoie la clé privée au client, applique une procédure et des critères qui sont définis par l'autorité de certification (AC), qui certifie la clé publique et appose sa signature sur le certificat, parfois fabriqué par un opérateur de certification [2].

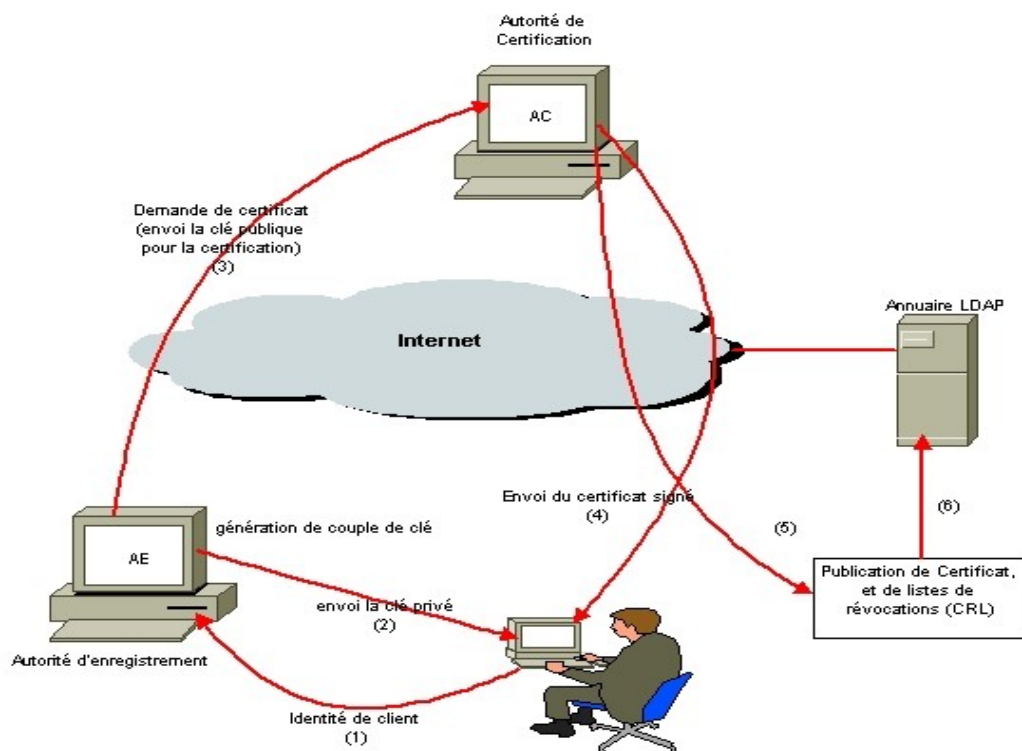


FIGURE 2.1 – Organisation de la PKI

2.3 Les acteurs d'une infrastructure de gestion de clés

Il existe plusieurs acteurs qui contribuent à l'infrastructure de gestion de clés :

- **Administrateur** : un administrateur met en œuvre les politiques de certification et déclarations des pratiques de certification de l'IGC au sein de la composante qu'il administre. Il est responsable de l'ensemble des services rendus par cette composante.[3]
- **Autorité administrative (AA)** : composante de l'IGC qui définit et fait appliquer les politiques de certification et les déclarations des pratiques de certification par l'IGC. [3]
- **Autorité de certification (AC)** : autorité chargée par un ou plusieurs utilisateurs

de créer et d'attribuer les certificats. Cette autorité peut, facultativement, créer les clés d'utilisateur [9594-8]. [4]

- **Autorité de certification racine (ACR)** : AC prise comme référence par une communauté d'utilisateurs (incluant d'autres AC). Elle est un élément essentiel de la confiance qui peut lui être accordée dans un contexte donné.[1]
- **Autorité d'enregistrement (AE)** : composante de l'IGC qui vérifie les données propres au demandeur ou porteur de certificat ainsi que les contraintes liées à l'usage d'un certificat, conformément à la politique de certification. L'AE est une composante optionnelle de l'IGC qui dépend directement d'au moins une autorité de certification.
- **Autorité d'horodatage (AH)** : composante de l'IGC qui délivre des contremarques de temps sur des données qui lui sont présentées. L'AH définie dans ce document ne fournit ses services que pour le compte de l'IGC en interne.[3]
- **Composante de l'IGC** : plate-forme constituée d'au moins un poste informatique, une application, un moyen de cryptologie et jouant un rôle déterminé au sein de l'IGC. Une composante peut être une AC, une AE, une TPC, une AH, etc. [4]
- **Contrôleur** : personne désignée par une autorité compétente et dont le rôle est de procéder de manière régulière à des contrôles de conformité de la mise en œuvre des politiques de certification, des déclarations des pratiques de certification et des services effectivement fournis par la composante de l'IGC.[1]
- **Exploitant** : personne travaillant pour le compte de l'IGC et disposant de droits d'accès à une autorité associés aux rôles qui lui sont attribués.
- **Ingénieur système** : il est chargé de la mise en route, de la configuration et de la maintenance technique de la plate-forme informatique de la composante. Il assure l'administration du système et du réseau de cette plate-forme.
- **Opérateur** : l'opérateur d'une composante (AE, AC, AH ou TPC) réalise l'exploitation des services offerts par la composante, dans le cadre de ses attributions. Il est chargé de lancer l'exécution des fonctions cryptographiques.
- **Responsable de sécurité** : le responsable de sécurité est responsable de l'application de la politique de sécurité physique et fonctionnelle d'une composante de l'IGC et de son environnement. Il gère les contrôles d'accès physiques à la plate-forme de la composante, et est chargé de mettre en œuvre la politique de sécurité régissant la composante.
- **Tierce Partie de Confiance (TPC)** : organisme gérant pour le compte d'autrui des conventions secrètes de moyens ou de prestations de cryptologie permettant d'assurer des fonctions de confidentialité [L90-1170].
- **Utilisateur final** : toute entité (utilisateur humain, organisme ou entité des technologies de l'information) détenant un certificat de clé publique généré par une composante

de l'IGC. L'utilisateur final est appelé demandeur de certificat lorsqu'il effectue une demande de certificat auprès d'une AE. Il est appelé porteur de certificat dès l'instant où il dispose d'un certificat émis par l'IGC. Un utilisateur final ne peut émettre de certificats pour le compte d'autres entités (composantes ou utilisateurs finaux).

- **Utilisateur de certificat** : toute entité (utilisateur humain, organisme ou entité des technologies de l'information) utilisant un certificat de clé publique à des fins de vérification de signature numérique. Un utilisateur de certificat ne détient pas forcément de certificat propre.

2.4 Les certificats

Une organisation de confiance certifie les identités (c'est à dire qu'elle garantit la correspondance entre les clés et les identités).[8] L'information certifiée est transportée dans un certificat électronique qui est à son tour délivré et signé par cette organisation. Un certificat électronique de confiance est donc le résultat de la combinaison de clé et de l'organisation. Un certificat représente un lien de confiance entre :

- Une identité
- Une clé publique
- Une période de validité
- Un usage

Le standard X.509 est celui sur lequel les certificats numériques utilisent souvent, et il comporte un certain nombre de champs (non-exhaustif) [1] :

- **Certificate signature Value** : Valeur de signature du certificat
- **Certificate signature algorithm** : algorithme et niveau de chiffrement utilisé
- **Version** : Version du certificat
- **Serial Number** : Numéro de série dans l'AC
- **Issuer** : AC validante
- **Validity** : durée de validité
- **Subject** : Distinguished Name (DN), champ servant généralement à nommer le certificat. Il est donc important de bien choisir son DN :
 - Il doit être lisible par l'homme en respectant quelques restrictions des bonnes pratiques (pas d'email, pas de rôle)
 - L'identifiant unique interne d'un certificat et (Issuer + Serial Number + version), mais est peu lisible. Ce rôle est rempli dans chaque organisation par le DN
- **Subject Public Key** : clé public

- **Subject Public Key algorithm** : algorithme et niveau de chiffrement utilisé.
- **Extensions** : information critique et non critique sur le nombre d'AC, sur les points de distribution et :
 - Certificates Key Usage : Authentification, chiffrement, signature, CRL
 - Certificate Extended Key Usage : Horodatage, OCSP, EFS

2.5 Les composants d'une PKI

L'infrastructure de gestion de clé se compose généralement en quatre parties qui sont les suivantes [2] :

- **Autorité de certification (AC)** : en charge de la signature des demandes de certificats (CSR) et des listes de révocations (CRL)
- **Autorité d'enregistrement (AE)** : en charge de la vérification et de la délivrance des identités
- **Autorité de dépôt** : stockage des certificats et des listes de révocation
- **Sujet** : utilisateur ou système, sujet de l'identité appelé aussi entité terminale (EE)

On peut rajouter d'autres éléments comme suite :

- **Autorité de séquestre** : stockage centralisé des clés de chiffrement à des fins de recouvrement
- **Une entité d' enrôlement** : Comptoir pour faire les demandes

L'infrastructure de gestion de clé suit une hiérarchie pyramidale qu'on voit comme suite :

- **Au sommet une AC « Racine »** : qui est la base de la confiance
- **Des AC intermédiaires** :
 - Elles sont certifiées par l'AC racine
 - Elles peuvent délivrer des certifications à d'autres AC intermédiaires ou opérationnelles
- **Des AC opérationnelles** :
 - Elles sont certifiées par l'AC racine ou par une AC intermédiaire
 - Elles peuvent délivrer des certifications aux entités terminales

Chapitre 3

Introduction aux réseaux de neurones

Dans ce chapitre nous essayerons d' introduire la notion des réseaux de neurones comme on va les utiliser lors de la réalisation du projet .

3.1 Réseau de neurone

3.1.1 Définition

Un réseau neuronal est l'association, en un graphe plus ou moins complexe, d'objets élémentaires, les neurones formels. Les principaux réseaux se distinguent par l'organisation du graphe (en couches, complets. . .), c'est-à-dire leur architecture, son niveau de complexité (le nombre de neurones, présence ou non de boucles de rétroaction dans le réseau), par le type des neurones (leurs fonctions de transition ou d'activation) et enfin par l'objectif visé : apprentissage supervisé ou non, optimisation, systèmes dynamiques...[5]

3.1.2 Structure d'un RNA

La figure suivante montre la structure d'un neurone artificiel. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amonts. A chacune de ces entrées est associée un poids w abréviation de weight (poids en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associée un poids [5].

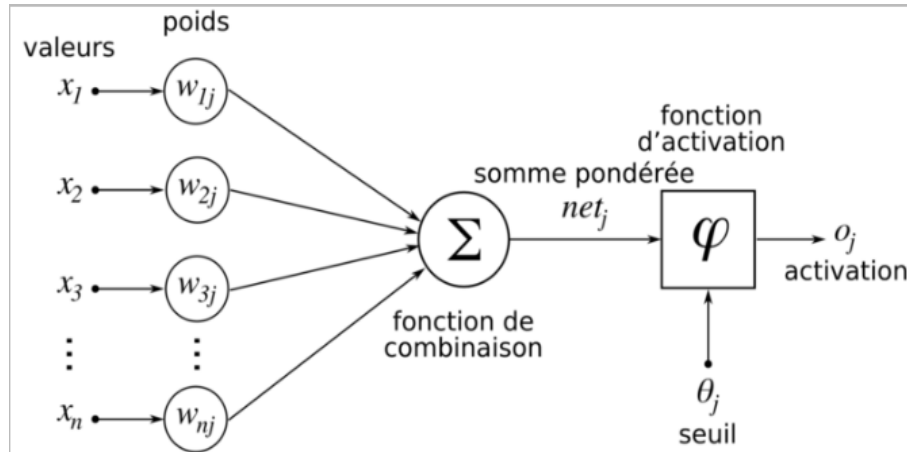


FIGURE 3.1 – Structure d'un réseau de neurones artificiels

3.1.3 Logistic Regression

est un algorithme de classification utilisé pour attribuer des observations à un ensemble discret de classes. Certains des exemples de problèmes de classification sont les spams par courrier électronique ou non, les transactions en ligne frauduleuses ou non, les tumeurs malignes ou bénignes. La régression logistique transforme sa sortie en utilisant la fonction logistique sigmoïde

pour renvoyer une valeur de probabilité.

La régression logistique est un algorithme d'apprentissage automatique utilisé pour les problèmes de classification. Il s'agit d'un algorithme d'analyse prédictive basé sur le concept de probabilité [8].

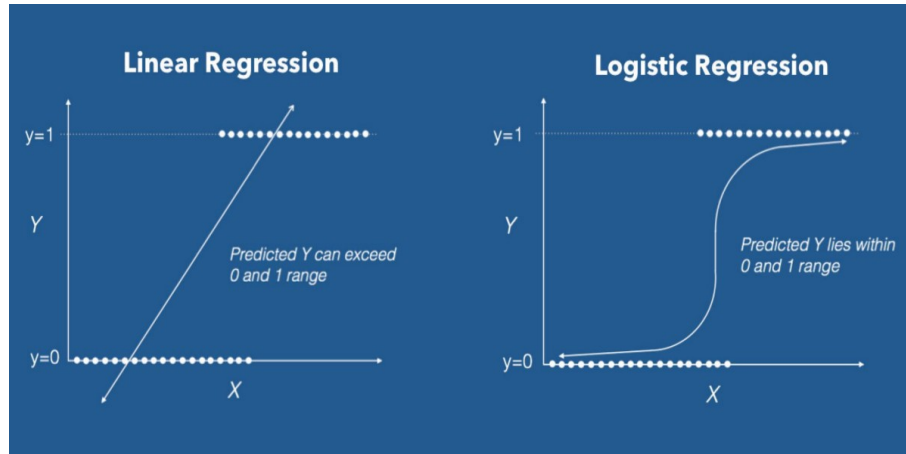


FIGURE 3.2 – La régression linear et logistique

3.1.4 Fonction sigmoïde (fonction logistique)

L'algorithme de régression logistique utilise également une équation linéaire avec des prédicteurs indépendants pour prédire une valeur. La valeur prédite peut être n'importe où entre l'infini négatif et l'infini positif. La sortie de l'algorithme doit être une variable de classe, c'est-à-dire 0-non, 1-oui. Par conséquent, nous réduisons le résultat de l'équation linéaire dans une plage de $[0,1]$. Pour écraser la valeur prédite entre 0 et 1, nous utilisons la fonction sigmoïde.[5]

$$h = g(z) = \frac{1}{1 + e^{-z}}$$

FIGURE 3.3 – La régression linear et logistique

3.1.5 Limite de décision

Notre fonction de prédiction actuelle renvoie un score de probabilité compris entre 0 et 1. Afin de le mapper sur une classe discrète (vrai / faux, accepté / non accepté), nous sélectionnons une valeur seuil ou un point de basculement au-dessus duquel nous classerons les valeurs dans la classe 1 et en dessous duquel nous classons les valeurs dans la classe 2[8].

$$\begin{aligned} p \geq 0.5, class &= 1 \\ p < 0.5, class &= 0 \end{aligned}$$

FIGURE 3.4 – classification de notre résultat

Pour la régression logistique avec plusieurs classes, nous pourrions sélectionner la classe avec la probabilité prédite la plus élevée.

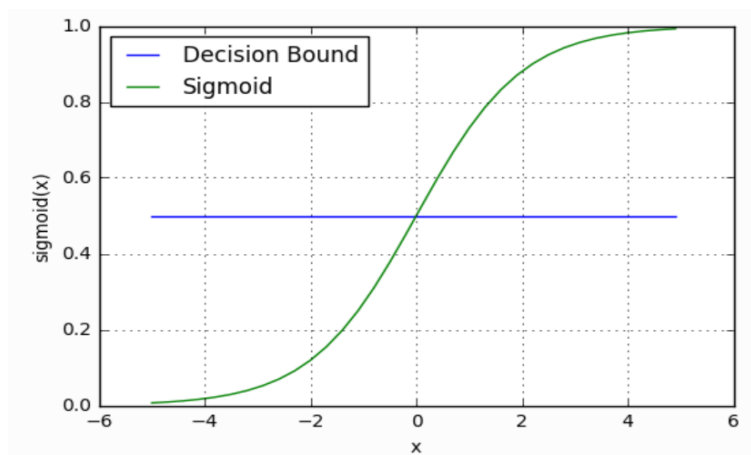


FIGURE 3.5 – La fonction sigmoïde graphique

Chapitre 4

Réalisation d'une solution de formalisation d'une CP

Dans ce chapitre nous allons présenter l'environnement de développement avec ses différents composants, pour exposer par la suite les étapes de réalisation du projet ainsi que le résultat obtenu .

4.1 Conception du document PC avec JSON

4.1.1 JSON

4.1.1.1 Définition

JavaScript Object Notation (JSON) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple. Créé par Douglas Crockford entre 2002 et 2005.

Des bibliothèques pour le format JSON existent dans la plupart des langages de programmation.

4.1.1.2 Les caractéristiques

Un document JSON comprend deux types d'éléments structurels :

- des ensembles de paires « nom » (alias « clé ») / « valeur »
- des listes ordonnées de valeurs

Ces mêmes éléments représentent trois types de données :

- des objets
- des tableaux
- des valeurs génériques de type tableau, objet, booléen, nombre, chaîne de caractères ou null.

4.1.1.3 Les avantages


Les avantages de JSON :

- La vitesse de traitement.
- La simplicité de mise en oeuvre. On n'a pas besoin de parser un fichier XML pour extraire des informations à travers le net, car JSON est reconnu nativement par JavaScript.
- Les contenus binaires peuvent être intégré et échangés sur le net avec une représentation textuelle spéciale avec une commande comme : `new Buffer(file).toString('base64')`.

4.1.1.4 La syntaxe

Les éléments de JSON sont :

- Un objet : contient d'autres objets ou des variables.
- Une variable scalaire : Number, String, Boolean.

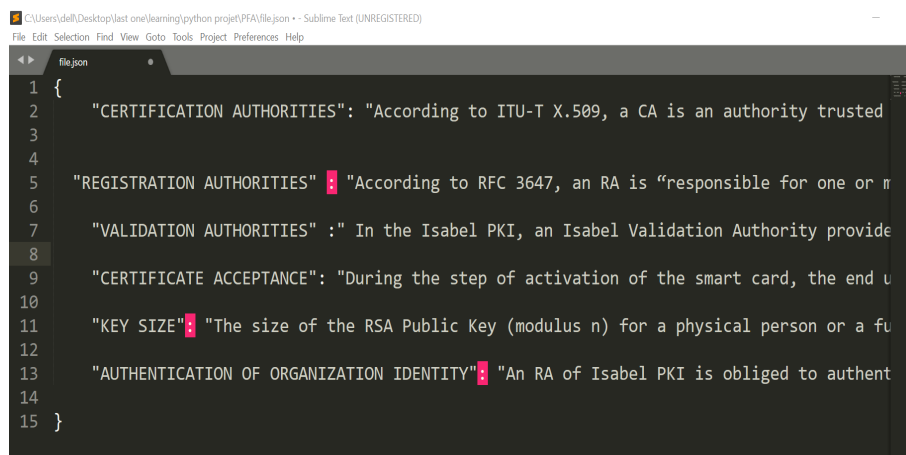


```
"nom" : "valeur"
```

FIGURE 4.1 – syntaxe du fichier JSON

4.1.2 Modèle proposé d'une PC en JSON format

On exposera ci-dessous des parties du modele de formalisation en JSON d'une politique de certification



```
1 {
2   "CERTIFICATION AUTHORITIES": "According to ITU-T X.509, a CA is an authority trusted
3
4
5   "REGISTRATION AUTHORITIES" : "According to RFC 3647, an RA is "responsible for one or m
6
7   "VALIDATION AUTHORITIES" : " In the Isabel PKI, an Isabel Validation Authority provide
8
9   "CERTIFICATE ACCEPTANCE": "During the step of activation of the smart card, the end u
10
11   "KEY SIZE" : "The size of the RSA Public Key (modulus n) for a physical person or a fu
12
13   "AUTHENTICATION OF ORGANIZATION IDENTITY" : "An RA of Isabel PKI is obliged to authent
14
15 }
```

FIGURE 4.2 – PC en JSON format

4.2 Environnement de développement

4.2.1 Language de développement :

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl. Le langage Python est placé sous une licence libre proche de la licence BSD4 ,fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux⁵, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

4.2.2 Pycharm :

PyCharm est un environnement de développement intégré utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.

4.3 Les bibliothèques utilisés

4.3.1 Numpy

NumPy est le paquet fondamental du calcul scientifique avec Python. Il contient entre autres :

- • un puissant objet tableau N-dimensionnel
- • fonctions sophistiquées (diffusion)
- • outils d'intégration de code C / C++ et Fortran
- • algèbre linéaire, transformée de Fourier et capacités de nombres aléatoires

Outre ses utilisations scientifiques évidentes, NumPy peut également être utilisé comme un conteneur multidimensionnel efficace de données génériques. Des types de données arbitraires peuvent être définis. Cela permet à NumPy de s'intégrer de manière transparente et rapide à une grande variété de bases de données. NumPy est sous licence BSD , ce qui permet sa réutilisation avec peu de restrictions.

4.3.2 Pandas

Pandas est une bibliothèque basée sur Python qui permet de travailler simplement et efficacement avec des données structurées. Jusqu'à présent, les versions se sont enchaînées rapidement mais celle-ci arrive plusieurs mois après la précédente. On peut donc espérer qu'un code développé pour cette nouvelle version demandera moins de maintenance.

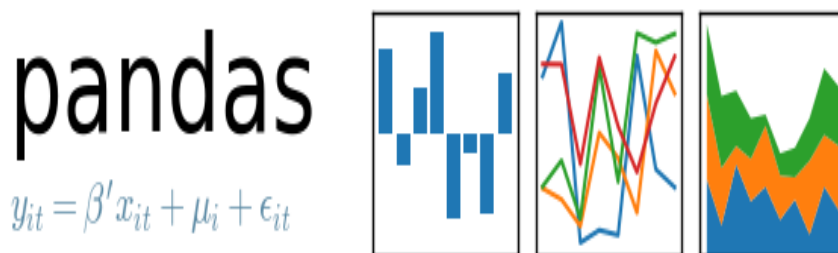


FIGURE 4.3 – Pandas logo

Les principales possibilités sont :

- récupérer des données depuis des fichiers CSV, tableaux Excel, des pages web, HDF5, etc.
- grouper, découper, alléger, déplacer, écrire les données.
- ces données peuvent être à une ou deux dimensions, avec des manques, ou encore temporelles avec ou sans périodicité

4.3.3 Tkinter

Tkinter (de l'anglais Tool kit interface) est la bibliothèque graphique libre d'origine pour le langage Python, permettant la création d'interfaces graphiques. Tkinter est un module de base intégré dans Python, normalement vous n'avez rien à faire pour pouvoir l'utiliser. L'un des avantages de Tkinter est sa portabilité sur les OS les plus utilisés par le grand public.

4.3.4 Nltk

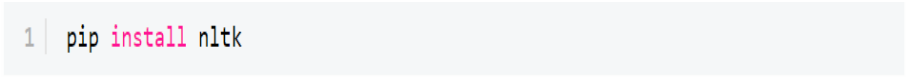
4.3.4.1 Définition

Natural Language Toolkit (NLTK) est une boîte-à-outil permettant la création de programmes pour l'analyse de texte. Cet ensemble a été créé à l'origine par Steven Bird et Edward Loper, en relation avec des cours de linguistique informatique à l'Université de Pennsylvanie en 2001. Il existe un manuel d'apprentissage pour cet ensemble titré Natural Language Processing with Python (en anglais).

4.3.4.2 installation de NLTK

Commençons par installer la librairie NLTK pour démarrer nos prochaines expérimentations en analyse du langage naturel.

Soyons fous! Son installation est assez simple. J'utilise le terminal de pycharm et je tape la commande suivante :



```
1 | pip install nltk
```

FIGURE 4.4 – NLTK commande d'installation

4.3.4.3 Travailler avec NLTK

1. chargement de nltk :

NLTK est un puissant package Python qui fournit un ensemble d'algorithmes de langages naturels variés. C'est gratuit, opensource, facile à utiliser, une grande communauté et bien documenté. NLTK comprend les algorithmes les plus courants tels que la tokenisation, le balisage partiel du discours, l'accrochage, l'analyse des sentiments, la segmentation des sujets et la reconnaissance des entités nommées.

```
#Loading NLTK
import nltk
```

FIGURE 4.5 – importation de NLTK

2. La tokenisation

La tokenisation est la première étape de l'analyse de texte. Le processus de décomposition d'un paragraphe de texte en petits morceaux tels que des mots ou des phrases s'appelle Tokenization. Le jeton est une entité unique constituant des blocs de construction pour une phrase ou un paragraphe.

3. La tokenisation des mots

Word Tokenizer divise le paragraphe de texte en mots. Chaque champs de notre politique de certification est sous forme d'un paragraphe ,alors on essaye de rendre le paragraphe sous forme d'une liste de mots

```
from nltk import wordpunct_tokenize
field=wordpunct_tokenize(word[key])
```

FIGURE 4.6 – tokénize les champs de la PC

4.4 Réalisation

4.4.1 Importation de la politique de sécurité

L'interface graphique suivante permet à un utilisateur d'importer la politique de certificat qu'on souhaite évaluer.

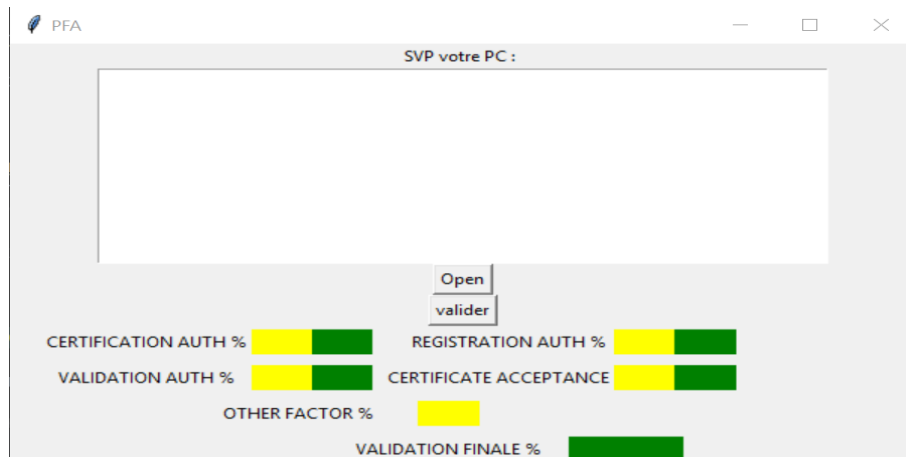


FIGURE 4.7 – importation de la PC

On clique sur « open » afin de parcourir les dossiers et récupérer le chemin de la politique de certification à évaluer.

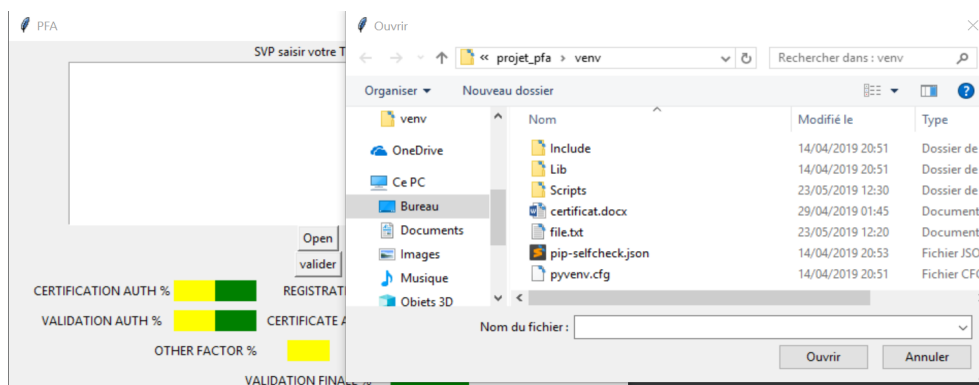


FIGURE 4.8 – Browse fichier

4.4.2 sécurité de champs de l'entrée

4.4.2.1 Exécution de code à distant

L'exécution de code à distance est la capacité d'un attaquant d'accéder au périphérique informatique de quelqu'un d'autre et d'y apporter des modifications, quel que soit son emplacement géographique.

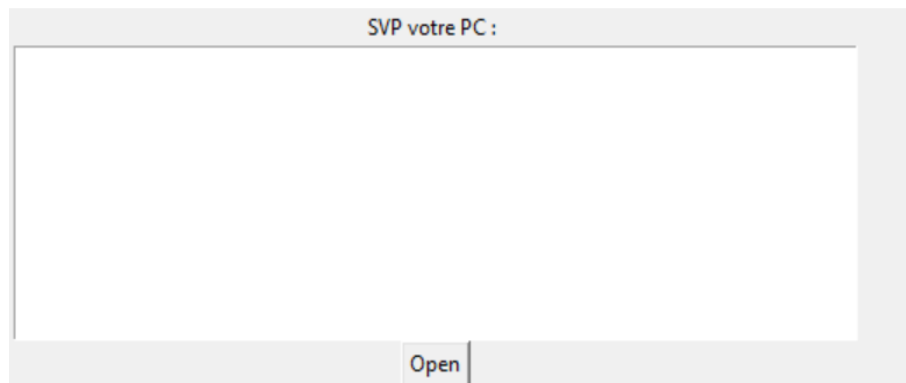


FIGURE 4.9 – champs d'entrée

Les vulnérabilités peuvent fournir à un attaquant la possibilité d'exécuter du code malveillant et de prendre le contrôle intégral du système affecté avec les privilèges de l'utilisateur exécutant l'application. Après avoir accédé au système, les attaquants tenteront souvent d'élever leurs privilèges. Le meilleur moyen de protéger notre application contre une vulnérabilité d'exécution de code à distance est de n'accepter que les fichiers comme entrée.

4.4.2.2 inclusion de fichiers locaux

Une vulnérabilité d'inclusion de fichier permet à un attaquant d'accéder à des fichiers non autorisés ou sensibles disponibles sur le serveur Web ou d'exécuter des fichiers malveillants sur le serveur Web en utilisant la fonctionnalité "inclure". Cette vulnérabilité est principalement due à un mécanisme de validation d'entrée incorrect, dans lequel l'entrée de l'utilisateur est transmise au fichier, y compris les commandes sans validation appropriée. L'impact de cette vulnérabilité peut entraîner l'exécution de codes malveillants sur le serveur ou la révélation de données présentes dans des fichiers sensibles.

Le meilleur moyen de protéger notre application contre une vulnérabilité d'inclusion de fichier locale est de contrôler notre entrée, et filtrer les extensions pas de fichier image qui est généralement source des attaques.

```
1 def UploadAction(event=None):
2     filename = filedialog.askopenfilename(filetypes = (("pdf or word files", "*.docx"),
3     sais.insert(INSERT, filename)
```

FIGURE 4.10 – champs d'entrée

4.4.3 Evaluation de PC à l'aide d'une disribution fréquentielle

Pour chaque champs on trouve des fichiers qui contiennent différents types de réponses qu'il doit contenir chaque champs . Notre programme essaye de déterminer si chaque champs répond







Nom	Modifié le	Type	Taille
 AUTHENTICATION_OF.txt	23/05/2019 13:26	Document texte	0 Ko
 CERTIFICATE_ACCEPTANCE.txt	23/05/2019 13:21	Document texte	1 Ko
 CERTIFICATION_AUTHORITIES.txt	22/05/2019 03:36	Document texte	1 Ko
 file.txt	25/05/2019 14:25	Document texte	0 Ko
 REGISTRATION_AUTHORITIES.txt	23/05/2019 12:54	Document texte	1 Ko
 VALIDATION_AUTHORITIES.txt	23/05/2019 13:00	Document texte	1 Ko

FIGURE 4.11 – Les fichiers correspondant à chaque champs

à l'ensemble des réponses contenu dans chaque fichier , alors une estimation de probabilité de distribution est donnée par NLTK ,pour dire si chaque champs est bon ou non. A l'aide de

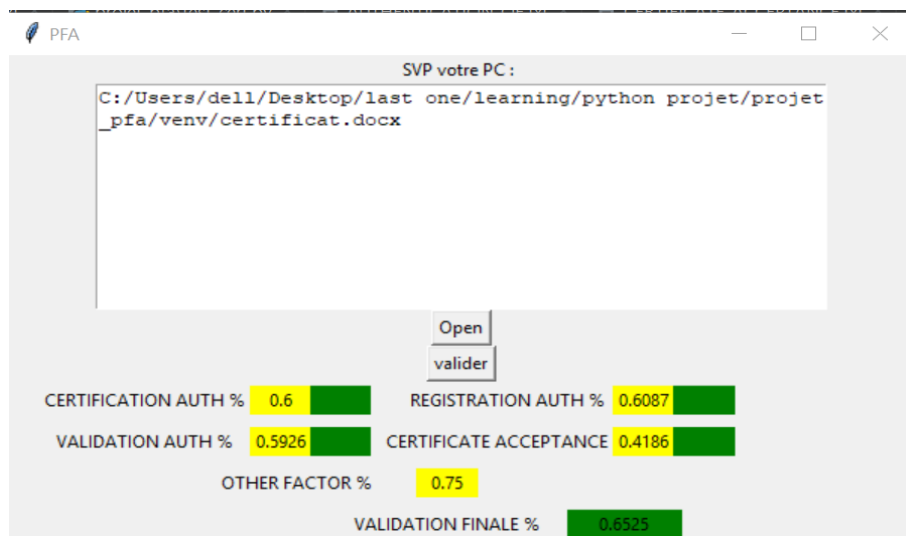


FIGURE 4.12 – Validation de la PC avec NLTK

score de chaque champs ,on serai capable de valider notre politique de certification et dire s'il est acceptable ou non.

4.4.4 Evaluation de la PC à l'aide d'un algorithme d'apprentissage automatique

4.4.4.1 Vectorization des champs

On divise d'abord la chaîne en jetons, qui sont des chaînes plus petites à peu près à des signes de ponctuation, des mots ou des parties de mots. Par exemple «CA verify subscriber's identities», il y a 5 jetons, dont chacun est vectorisé individuellement, ce qui donne une liste de vecteurs correspondant à chaque champ.

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
>>> sentences = ['According to ITU-T X.509, a CA is an authority trusted by one or more users']
>>> from sklearn.feature_extraction.text import CountVectorizer
>>> vectorizer = CountVectorizer(min_df=0, lowercase=False)
>>> vectorizer.fit(sentences)
CountVectorizer(analyzer='word', binary=False, decode_error='strict',
dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
lowercase=False, max_df=1.0, max_features=None, min_df=0,
ngram_range=(1, 1), preprocessor=None, stop_words=None,
strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
tokenizer=None, vocabulary=None)
>>> vectorizer.transform(sentences).toarray()
array([[1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 1, 2]],
      dtype=int64)
```

FIGURE 4.13 – Vectorization des données

4.4.4.2 Vectorization des réponses de chaque champs :







Nom	Modifié le	Type	Taille
 AUTHENTICATION_OF.txt	23/05/2019 13:26	Document texte	0 Ko
 CERTIFICATE_ACCEPTANCE.txt	23/05/2019 13:21	Document texte	1 Ko
 CERTIFICATION_AUTHORITIES.txt	22/05/2019 03:36	Document texte	1 Ko
 file.txt	25/05/2019 14:25	Document texte	0 Ko
 REGISTRATION_AUTHORITIES.txt	23/05/2019 12:54	Document texte	1 Ko
 VALIDATION_AUTHORITIES.txt	23/05/2019 13:00	Document texte	1 Ko

FIGURE 4.14 – Les fichiers des données

chaque fichier contient des réponses à vérifier par chaque champs de la politique de certification . après on transforme chaque phrase sous forme d'un vecteur.

```
reponse=["CA verify subscriber's (or user) identity"]
vectorizer.fit(reponse)
CountVecorizer(analyzer='word', binary=False, decode_error='strict',
dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
lowercase=False, max_df=1.0, max_features=None, min_df=0,
ngram_range=(1, 1), preprocessor=None, stop_words=None,
strip_accents=None, token_pattern='(?u)\\b\\w+\\b',
tokenizer=None, vocabulary=None)
vectorizer.transform(reponse)
<1x6 sparse matrix of type '<class 'numpy.int64'>'
with 6 stored elements in Compressed Sparse Row format>
r=vectorizer.transform(reponse)
r
<1x6 sparse matrix of type '<class 'numpy.int64'>'
with 6 stored elements in Compressed Sparse Row format>
r.toarray()
array([[1, 1, 1, 1, 1, 1]], dtype=int64)
```

FIGURE 4.15 – Vectorization des données

4.4.4.3 Apprentissage de notre modèle

Le modèle de classification que nous allons utiliser est la régression logistique, qui est un modèle linéaire simple mais puissant, mathématiquement parlant, qui est en fait une forme de régression entre 0 et 1 basée sur le vecteur caractéristique en entrée. En spécifiant une valeur de coupure (par défaut 0,5), le modèle de régression est utilisé pour la classification. Vous pouvez utiliser à nouveau la bibliothèque scikit-learn qui fournit le classifieur LogisticRegression :

```
>>> from sklearn.linear_model import LogisticRegression

>>> classifier = LogisticRegression()
>>> classifier.fit(X_train, y_train)
>>> score = classifier.score(X_test, y_test)

>>> print("Accuracy:", score)
Accuracy: 0.796
```

FIGURE 4.16 – Prédiction de notre modèle

Vous pouvez voir que la régression logistique a atteint un impressionnant 79,6%, mais voyons comment ce modèle fonctionne par rapport aux autres ensembles de données que nous avons.

Dans ce script, nous effectuons et évaluons l'ensemble du processus pour chaque ensemble de données que nous avons

```
for key in word.keys():
    list = []
    fichier = key.split(' ')
    file = fichier[0] + '_' + fichier[1] + '.txt'
    dat = pd.read_csv(file, names=['sentence'], sep='_')
    list.append(dat)
    df = pd.concat(list)
    vectorizer = CountVectorizer(min_df=0, lowercase=False)
    vectorizer.fit([word[key]])
    y = [i for i in range(len(df['sentence'].values))]

    d = vectorizer.transform([word[key]]).toarray()
    sentences_train, sentences_test, y_train, y_test = train_test_split(df['sentence'].values, y,

    vectorizer.fit(sentences_train)
    X_train = vectorizer.transform(sentences_train)
    X_test = vectorizer.transform(sentences_test)

    classifier = LogisticRegression()

    classifier.fit(X_train, y_train)
    y_pred = classifier.score(X_train, y_train)
    classifier.fit(X_test, y_test)
    y_pred1 = classifier.score(X_test, y_test)
    score_total += y_pred1
    tab[valeur].set(round(y_pred1, 4))
```

FIGURE 4.17 – Evaluation de chaque champs

4.4.4.4 Validation finale de notre PC

L'interface graphique suivante permet à un utilisateur d'importer la politique de certificat qu'on souhaite évaluer .

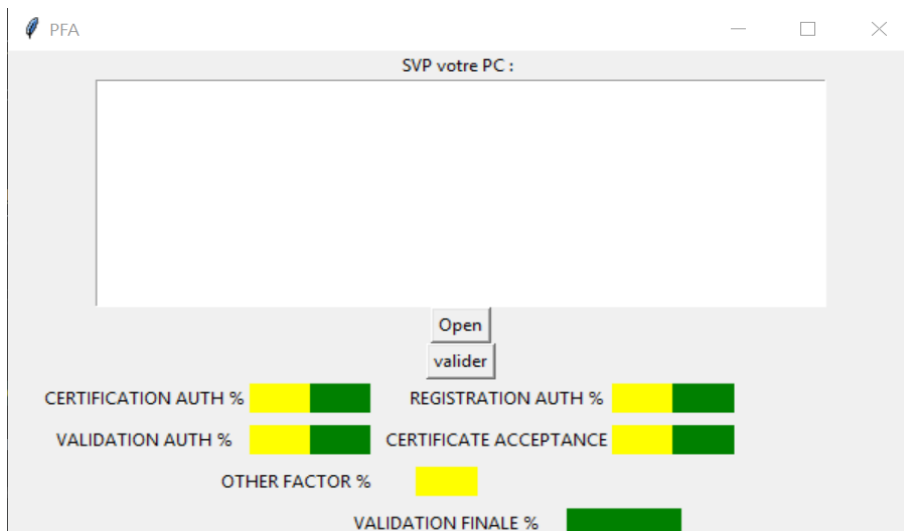


FIGURE 4.18 – Interface d'entrée

Après l'importation de PC , on l'évalue selon la score de chaque champs en vert et on détermine s'il est valide ou non :

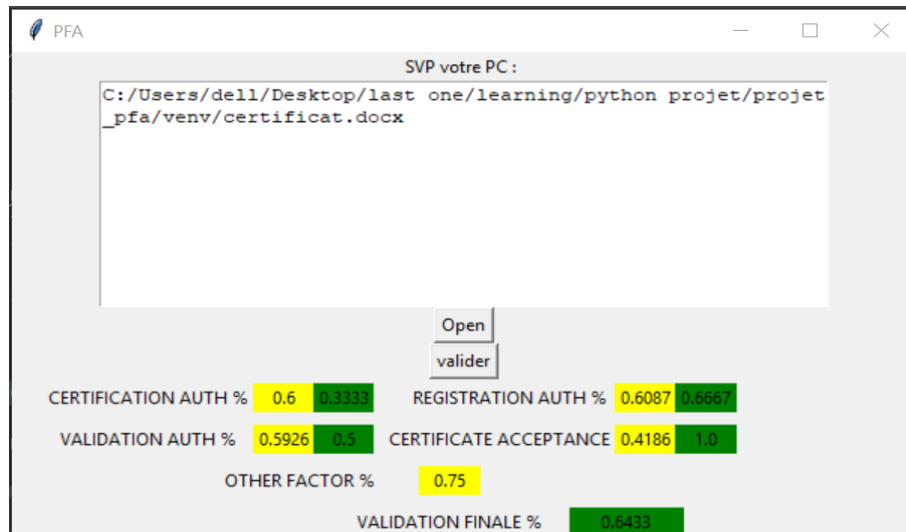


FIGURE 4.19 – Evaluation globale de notre PC

Finalement, notre PC est valide d'un degré de confiance égale 64,33% qui est généralement acceptable.

Conclusion Générale

L'infrastructure à clé publique permet donc d'assurer la sécurité de la communication électronique ainsi que la transaction dans un environnement ouvert vu que c'est une technologie utilisée pour la gestion et la distribution de clés et certificats publics. La structure de confiance syntaxique qui est connue sous le nom de modèle de confiance impact la propagation de la confiance dans cette infrastructure.

Notre projet propose donc une solution d'un mécanisme qui permet de lire et comprendre une politique de certification qui permet ensuite d'extraire les rubriques nécessaires pour pouvoir déterminer sa fiabilité et qui passe enfin à l'évaluer.

Comme perspectives de notre projet, on souhaite intégrer un moyen plus intelligent (une analyse syntaxique par exemple) pour pouvoir faciliter l'évaluation d'une politique de certification.

Bibliographie

- [1] MustaphaBenjada, PKI (PublicKeyInfrastructure) <https://www.securiteinfo.com/cryptographie/pki.sh>.
- [2] PhilippeLEGRIS, PKI\T1\textendashLesdéfinitions&concepts: <https://www.synetis.com/pki-les-definitions-concepts/>.
- [3] ThierryDulieu, ConceptsdebasedesPKI, <http://glasnost.entrouvert.org/pki.html>.
- [4] GroupeAdHocMessagerieSécurisée, PROCÉDURESETPOLITIQUESDECERTIFICATIONDECLÉS, <http://securinet.free.fr/annexe/pc2.pdf>.
- [5] <https://hackernoon.com/introduction-to-machine-learning-algorithms-logistic-regression>
- [6] PierreDeLoor, Réseauxdeneurones, <https://www.math.univtoulouse.fr/~besse/Wikistat/pdf/st-m-app-rn.pdf>.
- [7] ClaudeTouzet, LesRéseauxdeNeuronesArtificiels, http://www.touzet.org/Claude/Web-Fac-Claude/Les_reseaux_de_neurones_artificiels.pdf.
- [8] https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html,.