ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : 3222

THESE DE DOCTORAT

Structure de Recherche : Équipe Intelligent Processing and Security of Systems
Discipline : Informatique
Spécialité : Sécurité Informatique, Big Data et Machine Learning

Présentée le 06/07/2019 par :

SAMIRA DOUZI

Deep Learning Systèmes De Detection des Intrusions Malveillantes

Devant le jury :

Yvon KERMARREC	**PES**	Institut Mines Telecom, Brest- France	**Président**
Bouabid EL OUAHIDI	**PES**	Faculté des Sciences, Université, Mohammed V, Rabat	**Directeur de thèse**
Fouzia OMARY	**PES**	Faculté des Sciences, Université Mohammed V, Rabat	**Rapporteur/Examinateur**
Abderrahim MARZOUK	**PES**	Faculté des Sciences et Techniques, Université Hassan I, Settat	**Rapporteur/ Examinateur**
Oussama Mohammed REDA	**PH**	Faculté des Sciences, Université Mohammed V, Rabat	**Rapporteur / Examinateur**
Mohammed BOULMALEF	**PES**	Ecole du Numérique et d'Informatique, Université Internationale, Rabat	**Examinateur**

Année Universitaire : 2018/2019

✉ Faculté des Sciences, avenue Ibn Battouta, BP. 1014 RP, Rabat –Maroc
☎ 00212(0)5 37 77 18 76 🖷 00212(0)5 37 77 42 61 ; http://www.fsr.Um5.ac.ma

Dédicaces

Je dédie ce travail
A la mémoire de mon père qu'Allah lui accorde sa Miséricorde,
A ma chère maman que Dieu te protège et te garde auprès de moi,
A mon mari Khalid qui m'a supporté tout au long de ce travail.
A mes filles Douaa et Khadija,
A ma sœur Khadija,
A mes frères Hassan, Hamid et Adil,
A mes amies Fatima et Ibtissam
A tous mes proches.

Remerciements

Cette thèse a été effectuée sous la direction du Professeur Bouabid EL OUAHIDI au sein du Laboratoire Intelligent Processing and security of Systems (IPSS), Faculté des Sciences, Université Mohammed V de Rabat . Nombreux sont ceux que je voudrais remercier pour m'avoir aidé, soutenu ou accompagné durant ces années de thèse. C'est pour leur montrer toute ma gratitude et reconnaissance que je dédie cette page.

Je tiens à exprimer ma plus profonde reconnaissance à mon Professeur Bouabid EL OUAHIDI, Professeur de l'Enseignement Supérieur à la Faculté des sciences de Rabat. Je vous remercie pour m'avoir accueilli dans votre laboratoire et d'avoir accepté de diriger ce travail de recherche. Ma considération est inestimable. Au cours de ces années, votre grande disponibilité, votre rigueur scientifique, votre enthousiasme et vos précieux conseils m'ont permis de travailler dans les meilleures conditions. La confiance que vous m'avez accordée ainsi que nos nombreuses discussions m'ont permis de progresser et de mieux appréhender les différentes facettes du métier d'enseignant-chercheur. Soyez assuré, Monsieur, de toute mon estime et de mon profond respect.

J'adresse mes sincères remerciements à tous les membres du jury pour l'intérêt qu'ils ont bien voulu porter à ce travail de thèse :

Je remercie vivement Monsieur Yvon KERMARREC ,Professeur de L'enseignement Supérieur à L'institut Mines Telecom, Brest- France, de l'honneur qu'il m'a fait en acceptant d'examiner ce travail et de présider le jury de ma soutenance. Soyez assuré ,Monsieur, de mon plus profond respect.

J'adresse aussi mes sincères remerciements à Madame Fouzia OMARY, Professeur de l'Enseignement Supérieur à la Faculté des Sciences de Rabat, de l'honneur qu'elle m'a fait en acceptant de juger ce travail et d'en

être le rapporteur.Veuillez Madame trouver ici l'expression de ma profonde reconnaissance.

Je remercie chaleureusement Monsieur Abderrahim MARZOUK, Professeur de l'Enseignement Supérieur à L'Université Hassan I , Faculté des Sciences et Techniques de Settat d'avoir accepté de juger ce travail. Je le remercie également de m'avoir fait l'honneur d'en être le rapporteur. Veuillez accepter mes plus sincères remerciements pour votre présence dans ce jury et soyez assuré, Monsieur, de tout mon respect et de ma profonde gratitude.

Je remercie également Monsieur Oussama Mohammed REDA , Professeur Habilité à la Faculté des Sciences de Rabat de m'avoir fait l'honneur d'en être le rapporteur. Veuillez , Monsieur, trouver ici l'expression de ma profonde reconnaissance.

Je suis très reconnaissante à Monsieur Mohammed BOULMALF, Professeur de l'Enseignement Supérieur à l'Université Internationale de Rabat. Je le remercie de m'avoir fait l'honneur d'examiner ce travail et pour tout l'intérêt qu'il lui a porté. Votre présence dans ce jury m'honore. Veuillez agréer, Monsieur, le témoignage de mon respect le plus profond.

Je voudrais également remercier les membres du Laboratoire Laboratoire Intelligent Processing and security of Systems (IPSS) qui m'ont encouragé et soutenu. Qu'ils trouvent ici l'expression de mes sincères remerciements, en particulier ma collègue Ibtissam Benchaji.

Ces avant-propos seraient incomplets sans un grand remerciement aux membres de ma famille, en particulier ma très chére mère, mon Mari Khalid Nabih, mes filles Douaa et Khadija , ma sœur Khadija et mes frères Hassan ,Hamid et Adil . Ce travail vous appartient à tous. Votre soutien sans limites, votre amour inconditionnel et vos encouragements continus étaient le souffle de vie qui m'a permis à chaque fois de régénérer mes forces et d'accomplir ce travail dans des conditions meilleures. Je pense également à mon amie Fatima Sifou qui m'a toujours soutenu et encouragé durant toutes ces années d'études et pour toutes les années de travail que nous avons passé ensemble. Je lui exprime ma profonde sympathie et je lui souhaite beaucoup de bien.

Je ne saurais terminer ces remerciements sans remercier vivement tous

mes instituteurs et institutrices, mes professeurs de collège, d'enseignements supérieurs qui ont vu en ma personne une étincelle d'espoir. Je remercie également toutes les autres personnes aimables et serviables qui m'ont soutenu et qui ont contribué de près ou de loin à l'accomplissement de ce travail.

Résumé

Malgré le développement important de la sécurité des systèmes informatiques, les solutions existantes ne peuvent pas défendre complètement les systèmes informatiques contre les menaces malveillantes. La plupart de ces attaques sont de petites variantes des Cyber-attaques connues et répertoriés, mais même des mécanismes avancés tels que les machines Learning rencontrent des difficultés pour détecter ces petites attaques mutantes au fil du temps.
Le succès de Deep Learning (DL) dans divers domaines a suscité l'intérêt de l'utiliser pour la detection des attaques, ce qui pourrait constituer un mécanisme résilient face à ces petites mutations ou à des nouvelles attaques.
Dans cette thèse, nos travaux se focalisent sur la sécurité des systèmes informatiques, en particulier, Nous nous intéressons au filtrage du courrier électronique et à la detection des intrusions malveillantes.
Nous avons proposé des nouvelles méthodes de filtrages des emails spam et phishing en utilisant des outils Deep Learning comme le model Neural Paragraph Vector-Distributed Memory (PV-DM), L'Auto Encoder (AE) et le Denoising Auto Encoder(DAE). Ces filtres anti spam et anti phishing nous ont inspiré par la suite l'élaboration d'un système de détection des intrusions Malveillantes en se basant sur les Modèles PV-DM et Fuzzy Logic.

Mots Clés : Big Data,Deep Learning,IDS, Cyber Security, Fuzzy Logic.

Abstract

We are at the dawn of the era of "Big Data". The growing volume of information generated by businesses, the rise of social media and the Internet are fueling exponential data growth. Companies based on technologies such as Microsoft, Yahoo, Amazon and Google have kept data in Exabyte or even more. Most of the information cannot be managed by traditional tools. On the other hand, the increasing dependence on computer systems, offers a large attack surface to attackers, having all kinds of motivations : financial theft, data theft, disruption, damage to the reputation or simply to have "epic lulz". The result is a landscape of threats ranging from highly sophisticated attacks to opportunistic cyber-criminality. Despite the significant development of network security, existing solutions cannot fully defend computer networks against malicious threats. Moreover, most of these attacks are small variants of the cyber-attacks known until now. This indicates that even advanced mechanisms such as traditional machine learning systems have difficulty detecting these small mutant attacks over time. In addition, the success of Deep Learning (DL) in various areas of Big Data has spurred many interests in the field of cyber security. The use of DL for attack detection in cyber space could be a resilient mechanism against small changes or new attacks due to its ability to extract high-level features. In this thesis, we propose Methods of filtering unwanted emails, commonly known as SPAM using Neural Networks and Deep Learning tools : Auto Encoder and Denoising Auto Encoder. Moreover, we have adopted this filter as an intrusion detection system that identifies suspicious actions based on the local and global context of the attack. Thus, we have proposed an Intrusion Detection system that differs from the above by integrating fuzzy logic concepts such as membership functions and fuzzy sets as well as fuzzy association rules.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction Générale

Le monde s'est transformé en monde numérique, où une énorme quantité de données est générée au jour le jour à partir de différents plates-formes qui ont donné naissance au Big Data. Le Big Data est utilisé dans presque tous les domaines tels que les entreprises, l'administration publique, la sécurité, la recherche scientifique, les soins de santé, l'Internet des objets (IoT), les recommandations des systèmes commerciaux ,les bourses, etc.

Le principal défi de cette énorme taille de données n'est pas seulement de la collecter, mais aussi de la gérer correctement et l'utiliser de manière efficace pour la prise de décision ou la prévision. En effet , le Big Data outre le volume qui s'explose , présente d'autres complications , généralement désignés par les quatre V : volume , variété , vélocité et véracité [2].

cet écosystème engendre une vaste surface d'attaques aux escrocs ayant diverses motivations (vol financiers , vol de données , perturbation, atteinte à la réputation etc.).C'est Bien pour cela que nous proposons les solutions Deep Learning pour la sécurisation des systèmes informatiques.

1 Le Big Data Analytique : Défis et Opportunités

Le concept générique de Big Data est de traiter des données d'entrée brutes non structurées ,complexes, de différentes tailles, et non supervisées,ou peuvent contenir seulement une petite partie de données supervisée. Par conséquent, cette variété de représentation des données soulève différents défis dans le Big Data pour extraire des informations utiles et structurées à partir des données non structurées et non catégorisées.
La Figure 1 illustre les types de données , les outils,les applications, et les défis de Big Data.

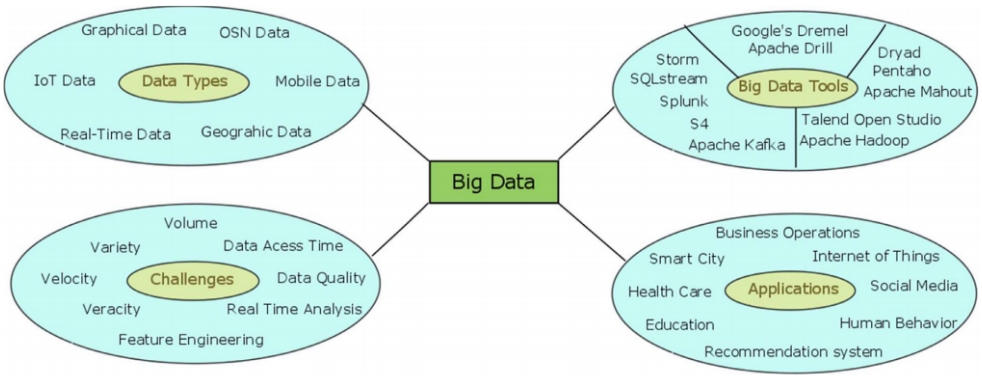

FIGURE 1: Classification de la littérature Big Data

Les méthodes traditionnelles ne permettent pas d'extraire des informations à partir de ces vo-

lumes considérable de données .Par conséquent,des techniques de machine learning ainsi que des machines très puissantes en calcul sont nécessaires pour gérer ce volume et cette variété de données. Le Deep Learning joue un rôle important dans l'analyse de Big Data [3] et il est largement utilisé dans différents domaines tels que les médias sociaux, les smart cities,l'internet des objets,le système de recommandation, et la cyber sécurité dernièrement.

2 Les réseaux de neurones

Les réseaux de neurones sont parmi les méthodes de Machine Learning les plus populaires. Leur unité de base c'est le neurone. Dans un réseau de neurones,les neurones se connectent les uns aux autres et chaque connexion a un poids.

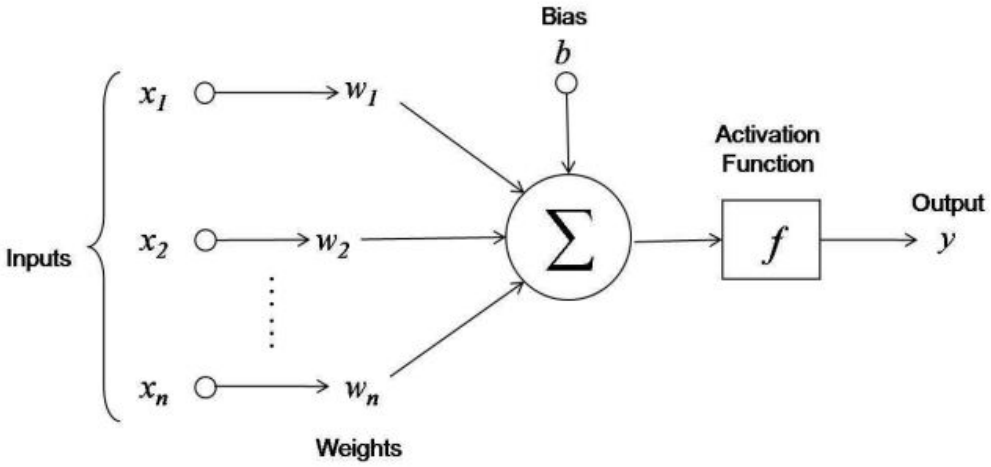

FIGURE 2: Schéma fonctionnel du réseau neurone

Dans un réseau de neurones (Voir Figure 1), les neurones se connectent les uns aux autres. Chaque connexion a un poids. Étant donné un ensemble de données X =$\{x_1, x_2, ..x_n\}$ et que les poids correspondants sont $w_1, w_2, ..w_n$ alors $\sum$ est :

$$\sum = \sum_{1}^{n} w_i x_i + b \tag{1}$$

Après la fonction d'activation est appliquée à $\sum$ pour donner la sortie y. (La fonction d'activation f peut être une fonction linéaire ou sigmoïde).

$$y = f(\sum) \tag{2}$$

Si cette sortie est différente de la sortie souhaitée, les poids sont ajustés à nouveau et ce processus se poursuit jusqu'à l'obtention de la sortie souhaitée. Ces poids de mise à jour se produisent conformément à l'algorithme de rétro-propagation.
Un réseau de neurones apprend en modifiant les poids de la connexion entre les neurones. Il existe trois types d'apprentissage, tels que **Apprentissage Supervisé, Apprentissage Non Supervisé** et **Apprentissage Par Renforcement**.

- ❄ Apprentissage Supervisé : le réseau fournira un vecteur de sortie en fonction du vecteur d'entrée. Ce vecteur de sortie est comparé au vecteur de sortie souhaité. S'il y a une différence, les poids vont être modifiés. Ce processus continue jusqu'à ce que la sortie réelle corresponde à la sortie souhaitée.

- ❄ Apprentissage Non Supervisé : le réseau identifie les modèles et les caractéristiques des données d'entrée et les relations entre les données d'entrée. Dans cet apprentissage, des

vecteurs d'entrée de types similaires se combinent pour créer des grappes. Lorsque le réseau obtient un nouveau modèle d'entrée, la sortie sera spécifiée en spécifiant la classe à laquelle appartient ce modèle d'entrée.

❄ Apprentissage Par Renforcement :accepte certaines réactions de l'environnement. Ensuite, le réseau modifie les poids.

2.1 Deep Learning

Le Deep learning est basé sur les réseaux de neurones, mais cette fois ci avec plusieurs couches cachées.Tout comme un réseau de neurones, l'algorithme Deep learning ou apprentissage profond va prendre en entrée un X afin de retourner un résultat y en sortie. La valeur d'entrée sera traitée et analysée au travers de nombreuses successions de neurones qui prennent en entrée les sorties des couches de neurones précédentes. L'origine du nom de l'apprentissage profond(Deep Learning) vient du fait que le maillage de neurones est complexe, on dit alors que la couche de neurones est alors « profonde ». Ce maillage peut être constitué de plusieurs milliers, voire de millions de neurones. Le concept du Deep Learning consiste à explorer un grand volume de données pour identifier automatiquement les modèles et extraire des caractéristiques de données complexes non supervisées sans implication humaine, ce qui en fait un outil important pour l'analyse de Big Data [4].

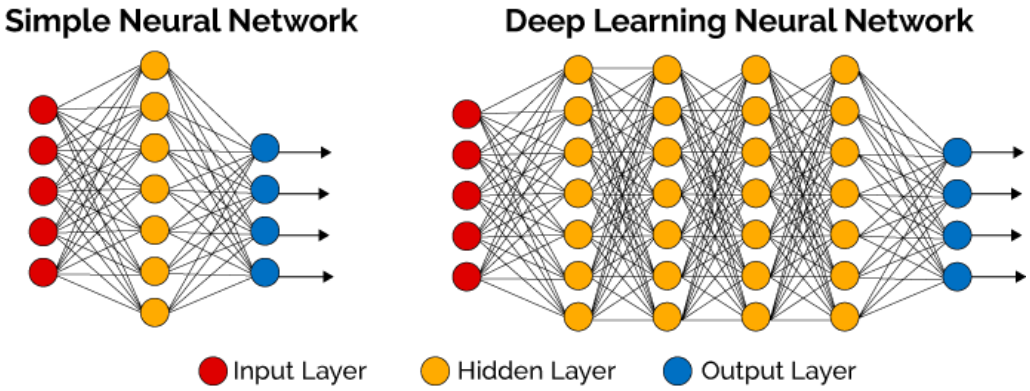

FIGURE 3: La différence entre un réseau de neurones simple et une architecture Deep Learning

Le Deep Learning utilise des techniques supervisées et non supervisées pour apprendre et extraire automatiquement les représentations de données. Il peut être utilisé pour résoudre les problèmes de Big Data (tels que la réduction et la compression,la récupération d'informations, etc.) de manière plus efficace. ce qui n'était pas possible avec les réseaux de neurones traditionnelles.Nous discutons ici trois architectures de Deep Learning utilisées dans notre sujet de recherche .

2.1.1 Auto Encoder

Les Auto Encoders sont des algorithmes d'apprentissage non supervisé à base de réseaux de neurones artificiels, qui permettent de construire une nouvelle représentation d'un jeu de données. Généralement, celle-ci est plus compacte, et présente moins de descripteurs, ce qui permet de réduire la dimensionnalité du jeu de données. la figure 2, montre L'architecture d'un autoencodeur qui est constitué de deux parties : l'Encodeur et le Décodeur.

L'Encodeur est constitué par un ensemble de couches de neurones, qui traitent les données afin de construire de nouvelles représentations dites "encodées". À leur tour, les couches de neurones du Décodeur, reçoivent ces représentations et les traitent afin d'essayer de reconstruire les données de départ. Les différences entre les données reconstruites et les données initiales permettent de mesurer l'erreur commise par l'Auto Encoder. L'entraînement consiste à modifier les paramètres

de l'Auto Encoder ,afin de réduire l'erreur de reconstruction mesurée sur les différents exemples du jeu de données.

La plupart du temps, on ne s'intéresse pas à la dernière couche du Décodeur, qui contient uniquement la reconstruction des données initiales, mais plutôt à la nouvelle représentation créée par l'Encodeur.

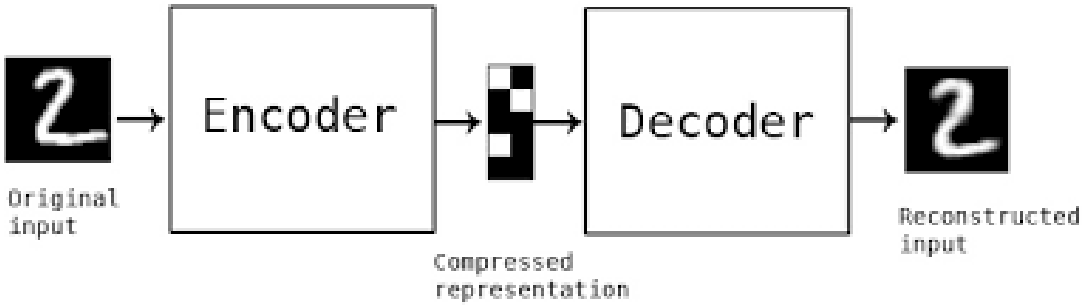

FIGURE 4: Architecture de l'Auto Encoder

2.1.2 Denoising Autoencoder

Le Denoising Auto Encoder (DAE) est une version stochastique d'un Auto Encoder classique. [5] L'idée derrière les Denoising Auto Encoder est simple : Afin de forcer la couche cachée de L'AE à découvrir des caractéristiques plus robustes, et l'empêcher de simplement apprendre l'identité, nous entrainons l'Auto Encoder à la reconstruction des entrée, d'une version corrompue de celle-ci. Intuitivement, Le Denoising Auto Encoder (DAE) effectue deux opérations :

- Coder les entrées.
- Essayer d'annuler l'effet d'un processus de corruption appliqué de manière stochastique à l'entrée de l'AE.

Ce processus ne peut être accompli qu'en capturant les dépendances statistiques entre les entrées.Par conséquent, DAE essaie de prédire les valeurs corrompues à partir des valeurs non corrompues. . Pour convertir un Auto Encoder en un Denoising Auto Encoder, il suffit d'ajouter une étape de corruption stochastique opérant sur les entrées (Figure 5). Les entrées peuvent être corrompues de nombreuses manières,en masquant par exemple de manière aléatoire quelques entrées en les rendant nulles.

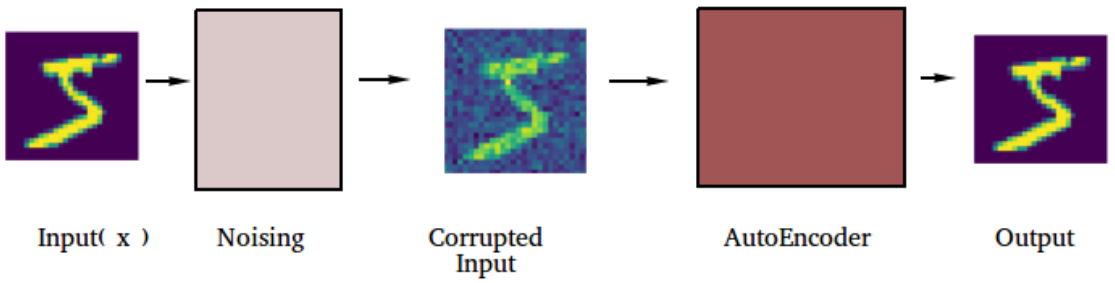

FIGURE 5: Architecture de Denoising Auto Encoder

2.1.3 Word Embedding

Le Word Embedding est le nom collectif d'un ensemble de techniques de modélisation de langage et d'apprentissage de caractéristiques dans Natural Language Processing (NLP), dans lesquelles les mots d'un vocabulaire sont transformés en des vecteurs de nombres réels. Word Embedding est capable de capturer le contexte d'un mot dans un document, la similarité sémantique et syntaxique, la relation avec d'autres mots, etc.

2.1.4 Word2vec

Word2Vec est l'une des techniques les plus populaires de Word Embedding , c'est un réseau de neurones à deux couches profondes formé pour reconstruire les contextes linguistiques des mots. Il prend en entrée un grand corpus de texte et produit un espace vectoriel, où chaque mot unique du corpus étant associé à un vecteur correspondant. Word2vec se base sur l'hypothèse :"qu'on peut prédire le sens d'un mot à partir de son contexte". Ainsi, si deux mots apparaissent à la même position dans deux phrases, ils sont très liés soit en sémantique, soit en syntaxe. Ainsi,Les vecteurs de mots sont positionnés dans l'espace vectoriel de telle sorte que, les mots qui partageant des contextes similaires soient situés à proximité les uns des autres dans l'espace[6].(Voir Figure 6).

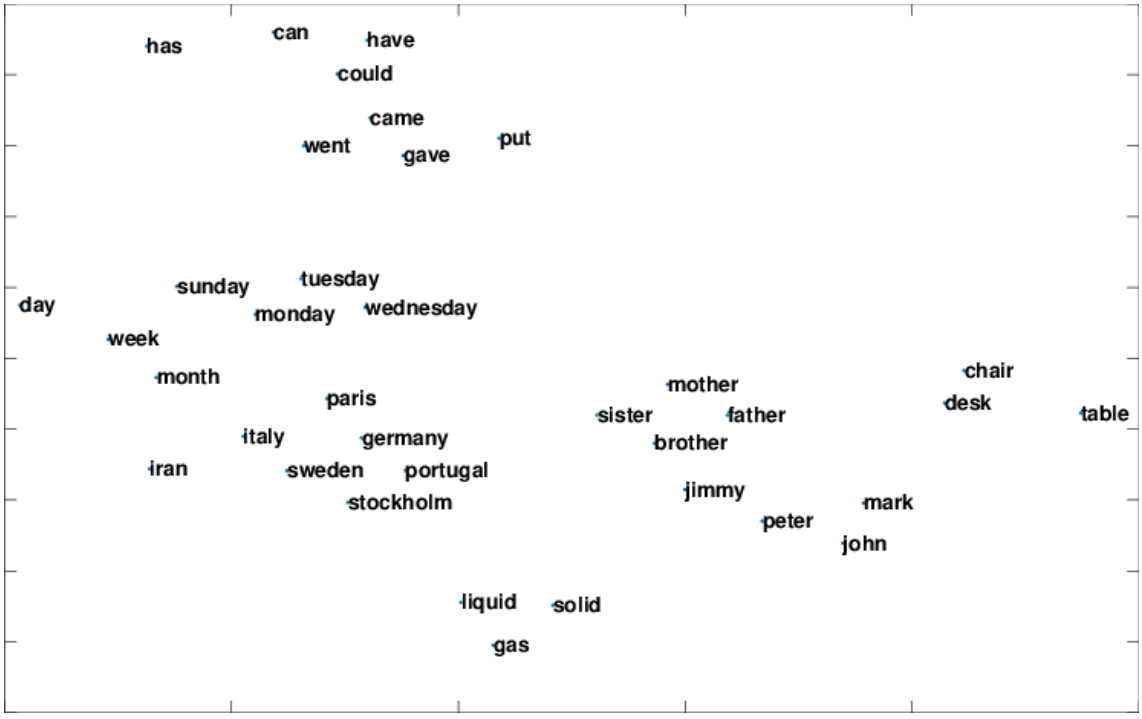

FIGURE 6: la répartition des mots dans l'espace vectoriel après l'application de Word 2vec

Word2vec utilise deux algorithmes pour produire une représentation distribuée de mots :

- le premier algorithme est appelé continuos Bag of Word (CBOW).La Figure 7 illustre l'architecture CBOW, appliquée à la phrase "Mon verre s'est brisé comme un éclat de rire". Dans cet exemple, le modelé est entrainé pour prédire le mot manquant ("comme"). les mots considérés comme trop communs, appelés aussi "mots outils" ou "stop word" sont éliminés des données fournies au réseau de neurones. La première couche du réseau projette chaque mot du contexte vers sa représentation vectorielle via une matrice W. Puis la

couche cachée analyse ces représentations vectorielles afin de tenter de prédire le mot central. A la fin de la phase d'entrainement, la matrice W contient les vecteurs représentant les différents mots de d'entrées.

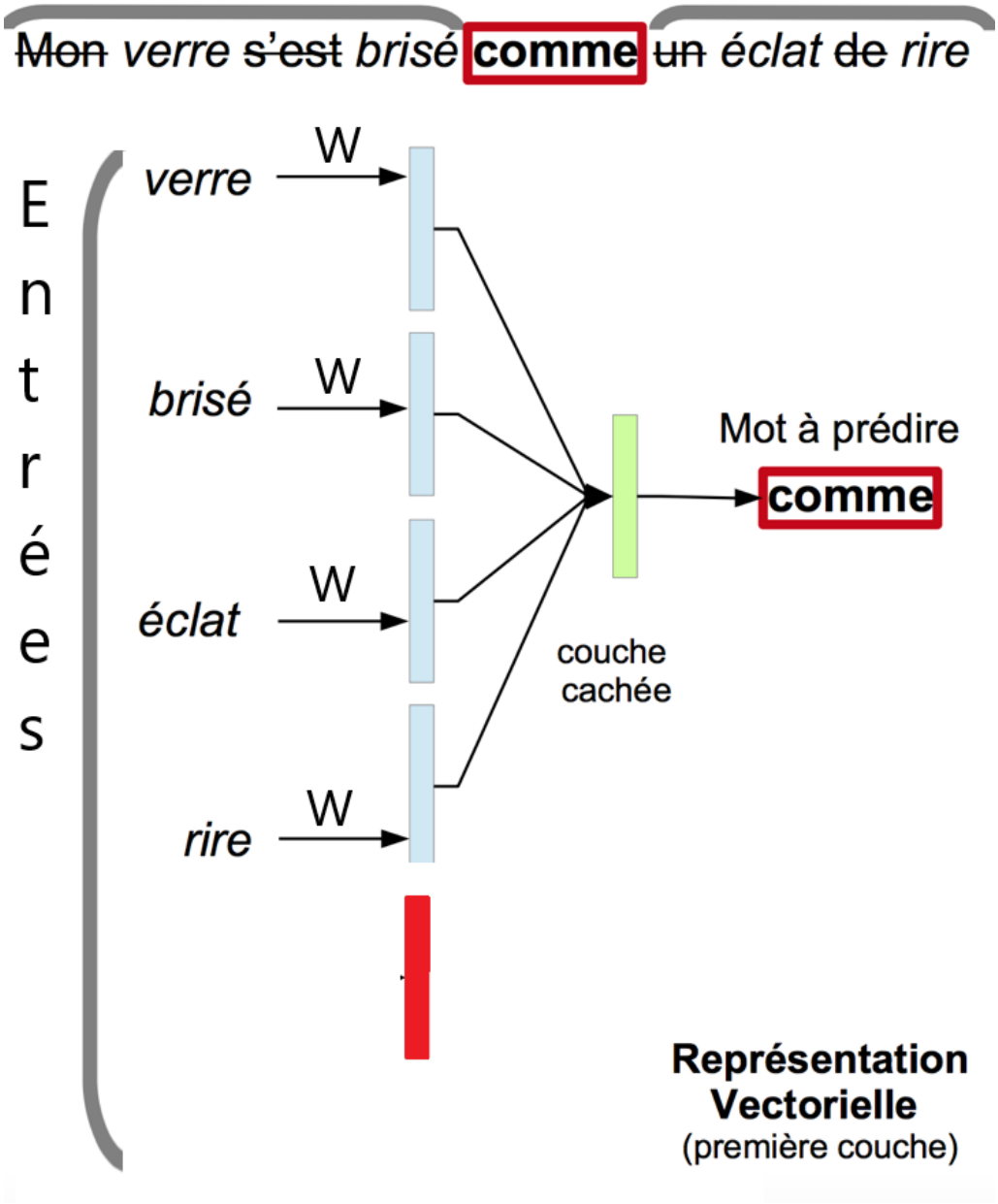

FIGURE 7: le framework de CBOW pour apprendre les vecteurs des mots

❄ le deuxième algorithme est appelé skip-gram, c'est l'image inverse du CBOW modèle. Le modèle skip-gram apprend en considérant un mot pour prédire le contexte environnant [7].La Figure 8 montre un exemple où le modèle prévoit les mots qui précédent et procèdent le mot France.

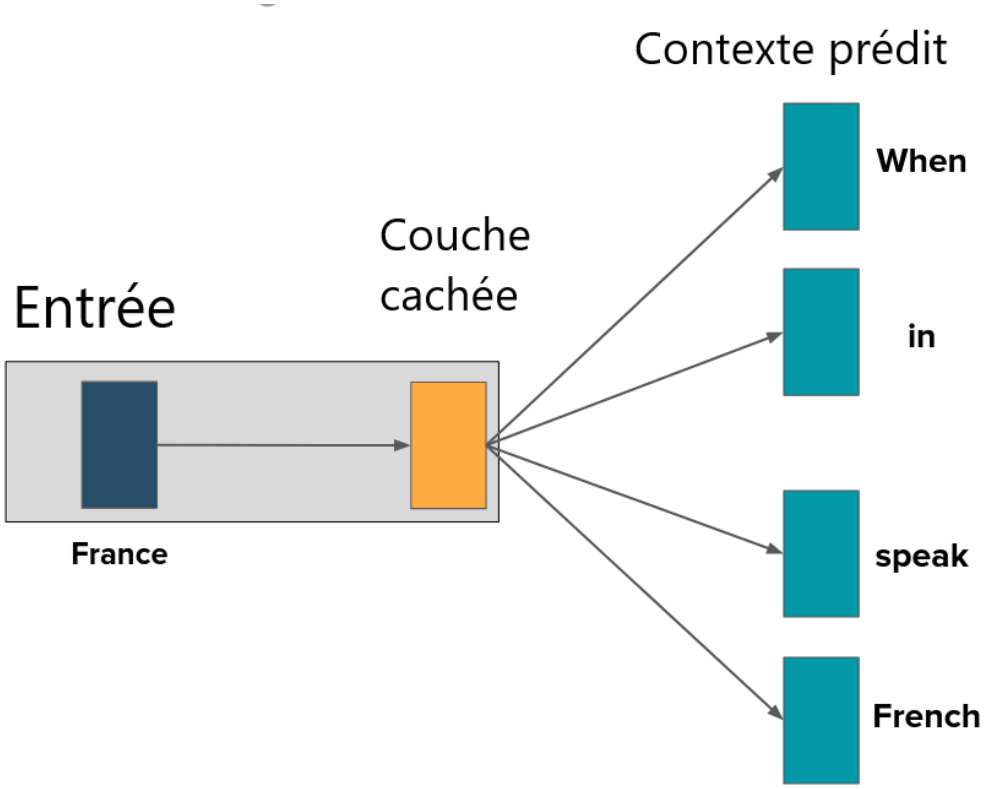

FIGURE 8: le framework skip gramme pour apprendre les vecteurs des mots

2.1.5 Paragraph2Vec

C'est une extension de Word2vec nommée Paragraph2Vec ou Doc2vec,elle est basée sur la même hypothèse que Word2vec dans le sens que la sémantique d'une phrase peut être prédit par son contexte.
Doc2vec permet de calculer la similarité sémantique entre deux documents et d'inférer des documents similaires de manière sémantique.Ainsi, si deux phrases apparaissent à la même position dans deux paragraphes, elles sont très liées soit de manière sémantique, soit de manière syntaxique.[8] Doc2vec utilise deux architectures pour produire une Paragraphe Vecteur :

- Le premier modèle est Paragraphe vecteur de Mémoire Distribué(PV-DM). Ce model est très similaire au modèle CBOW , la seule différence est l'ajout de la représentation du paragraphe(Le paragraphe contenant les mots du contexte) dans la liste des mots du contexte. Le paragraphe est transformé en vecteur via une matrice D. La concaténation du vecteur du paragraphe avec les vecteurs des mots du contexte sont utilisées pour prédire le mot manquant.(voir figure 9)

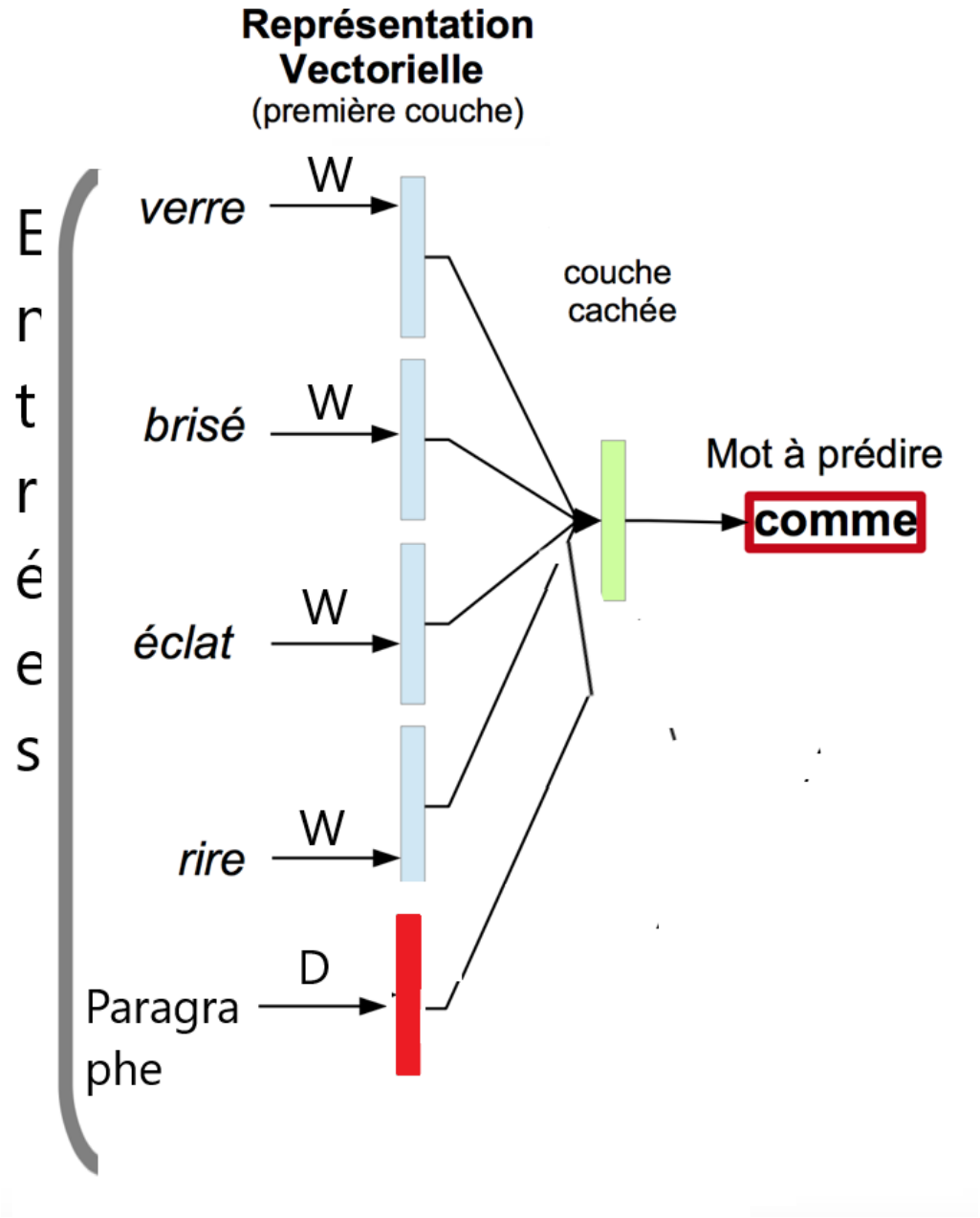

FIGURE 9: le framework PV-DM pour apprendre les vecteurs des paragraphes

❄ le deuxième modèle est le Paragraphe Vecteur Distributed Bag of Word (PV-DBOW). Ce modèle diffère légèrement du modèle PV-DM, il ignore les mots de contexte de l'entrée, mais force le modèle à prédire les mots échantillonnés de manière aléatoire à partir du paragraphe. La figure 10 montre un exemple où le vecteur du paragraphe est entrainé pour prédire un échantillon des mots.

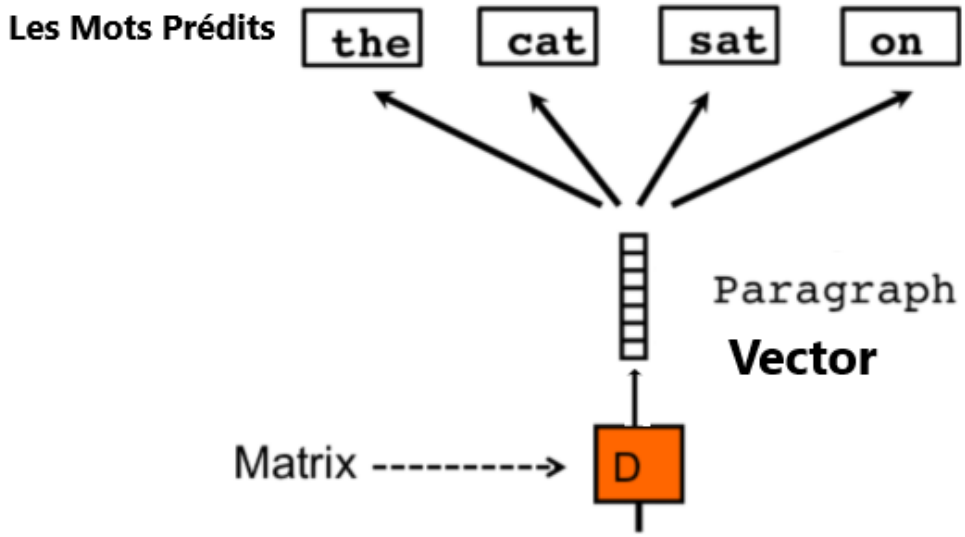

FIGURE 10: le framework PV-DBOW pour apprendre les vecteurs des paragraphes

3 Big Data Analytique et Cyber Security

3.1 La cyber-sécurité

L'avènement des appareils connectés et l'omniprésence d'Internet ont ouvert la voie aux intrus qui attaquent les systèmes informatiques, ce qui entraîne les cyber-attaques, pertes financières et vols d'informations.

Par conséquent, l'analyse de la sécurité des systèmes est devenue un sujet de préoccupation important et a retenu l'attention des chercheurs [9], en particulier dans le domaine de la détection des anomalies , qui est considéré comme crucial pour la sécurité des systèmes informatiques.
La cybersécurité vise à protéger les infrastructures des technologies de l'information et de la communication (TIC) (c'est-à-dire matériel informatique, logiciels, réseaux et données) contre les accès non autorisés et les perturbations .

Comme le nombre de services d'internet a augmenté,la taille de trafic données sur le réseau est devenue si grande et complexe, qu'il est très difficile de traiter avec les outils de traitement de données traditionnels. En plus,les entreprises produisent une quantité énorme de données difficile de stocker et à analyser. Par exemple,Cloud Security Alliance a signalé en 2013 qu'une entreprise aussi grande que HP générait environ un trillion d'événements de sécurité par jour.[1]

La détection rapide et efficace des intrusions est un problème très complexe en raison de la nature vaste et complexe des données de trafic réseau. Un système de détection des intrusions cyber-sécurité nécessite généralement un stockage et un traitement efficaces en temps réel de la grande taille des données du trafic réseau, et une analyse permettant d'identifier le trafic réseau malveillant. Cependant, des enquêtes préliminaires ont révélé que les approches existantes pour détecter les anomalies dans le réseau ne sont pas assez efficaces, en particulier en temps réel. Par conséquent, il est crucial de proposer un cadre qui gère efficacement le traitement de données volumineuses en temps réel et détecte les anomalies dans les systèmes informatiques. [10] [11].

3.2 Big Data Analytique - Cyber Security

Une enquête industrielle à grande échelle indique qu'un nombre croissant d'organisations ont pris conscience du l'importance et la valeur des technologies Big Data pour la protection de leurs infrastructures TIC contre les cyber-menaces[12] [14] . Les systèmes Big Data Analytique-Cyber

Security exploitent les technologies Big Data pour analyser les données d'événements de sécurité afin de protéger les systèmes informatiques, les ordinateurs et les données de l'entreprise des cyber attaques [13] et pour :

- Collecter des données sur les événements de sécurité.
- Peut gérer et analyser de manière efficace une grande quantité de données d'événements de sécurité.
- Opérer sur de grands volumes de données semi-structurés et non structurées (par exemple, du texte dans des fichiers journaux) pour extraire des informations précieuses sur la sécurité à partir des données.
- Révéler des modèles cachés et détecter les attaques lentes.
- Sélection et extraction de caractéristiques à partir des données collectées.
- Fournir une vue consolidée des informations de sécurité,etc.

La figure 8 montre quelques Outils de Big Data Analytique de cyber security.

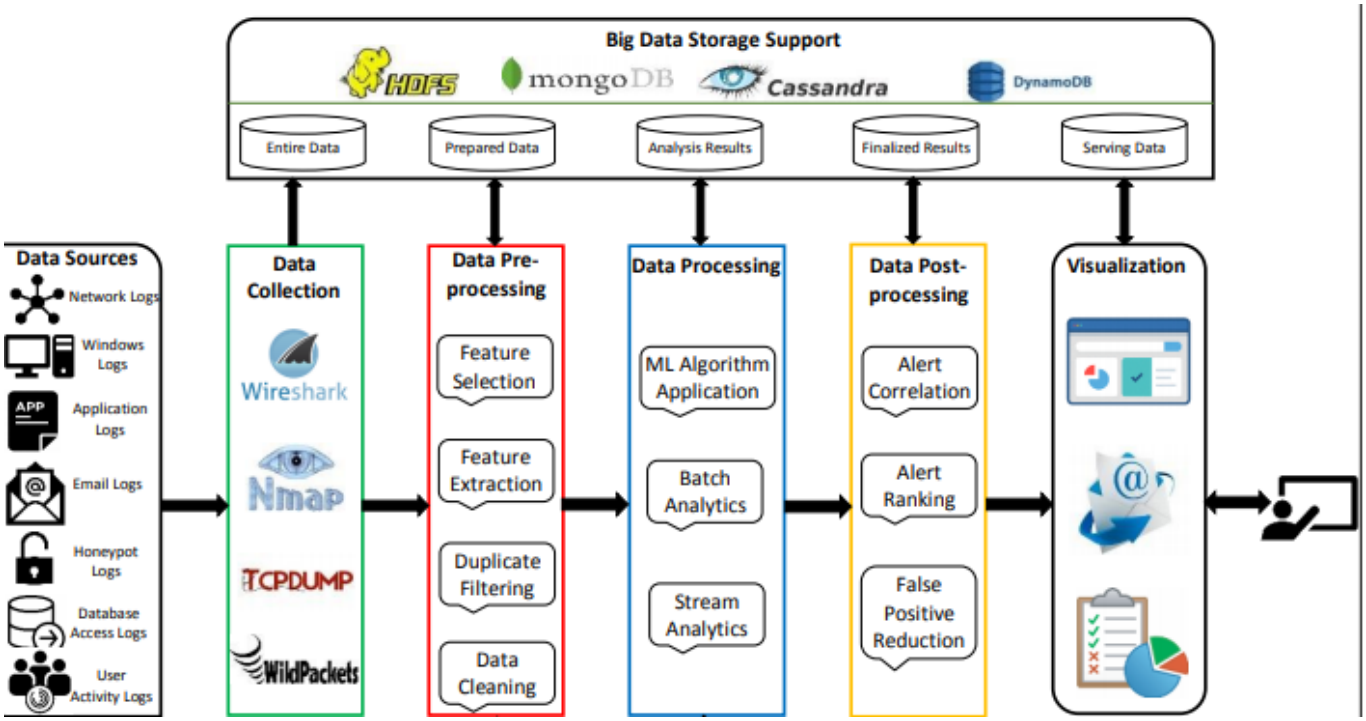

FIGURE 11: L'architecture des outils de Big Data Analytique pour la cyber sécurité.

4 Fuzzy Logique

Au lieu de labelliser des éléments comme étant 0 ou 1 comme dans la logique classique, la théorie des ensembles flous attribue des valeurs allant de 0 à 1 . Zadeh [15] déclare que Fuzzy Logic a la capacité de prendre des décisions rationnelles dans un environnement imprécis, incertain et incomplet. Les informations collectées concernant le trafic d'un réseau informatique s'inscrivent dans ce contexte, pour les raisons suivants :

- la détection d'intrusion implique de nombreux attributs numériques qui sont collectés et mesurés par des statistiques, ce qui peut entraîner des erreurs de détection élevées.
- Deuxièmement, il n'y a pas de frontière clairement définie entre les comportements normaux et anormaux en sécurité informatique.

Ceci justifie l'utilisation du Fuzzy logic pour la detection des intrusions malveillantes ,afin de tenir compte de ces incertitudes. Un fuzzy système d'inférence a pour but de mapper une entrée sur une sortie en appliquant le fuzzy raisonnement dans un processus en trois étapes (Figure 12) qui consiste à :

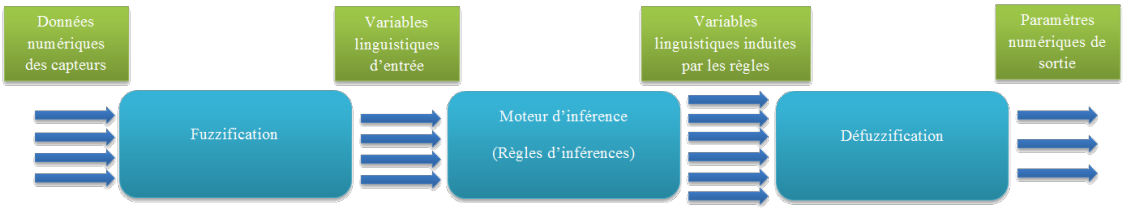

FIGURE 12: L'architecture d'un fuzzy système d'inférence.

❄ La Fuzzification : c'est l'étape qui consiste à la quantification floue des valeurs réelles d'une variable. Il y a un processus de fuzzification différent pour chaque variable numérique. Le but est d'obtenir une variable linguistique avec différentes valeurs linguistiques. Dans cette partie, on utilise les principes des sous-ensembles flous.
Prenons par exemple une température en degré Celcius provenant d'un capteur. On veut transformer cette donnée numérique en variable linguistique. On peut trouver plusieurs variables linguistiques qualifiant une température : *chaud, froid, très froid, tempéré, très chaud, etc* .
On peut utiliser plusieurs variables linguistiques pour caractériser un seul type de données. Dans cet exemple nous choisissons trois variables linguistiques pour qualifier la température : chaud, froid et tempéré. Pour cela, il faut créer une fonction d'appartenance pour chaque variable. Comme ces fonctions d'appartenances qualifient un même type de données, on peut les représenter sur le même graphique.Figure 13. Si par exemple le capteur renvoie la température T =17°C , après fuzzification, T sera chaude à 20%, tempérée à 60% et froide à 0% .voir Figure 14.

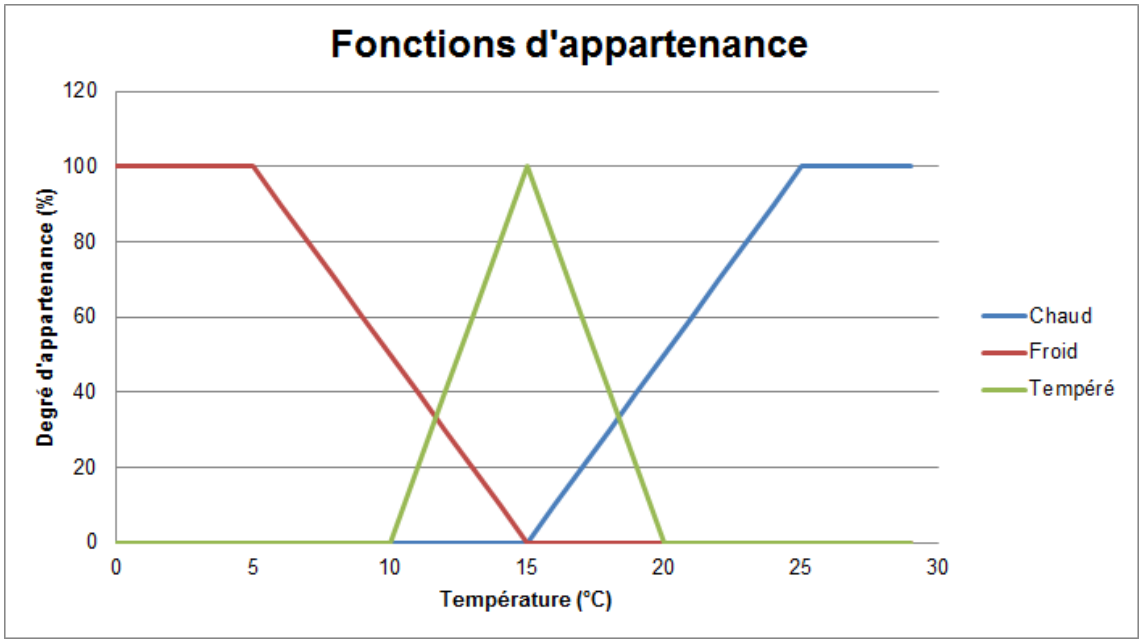

FIGURE 13: Fuzzification de la donnée Température.

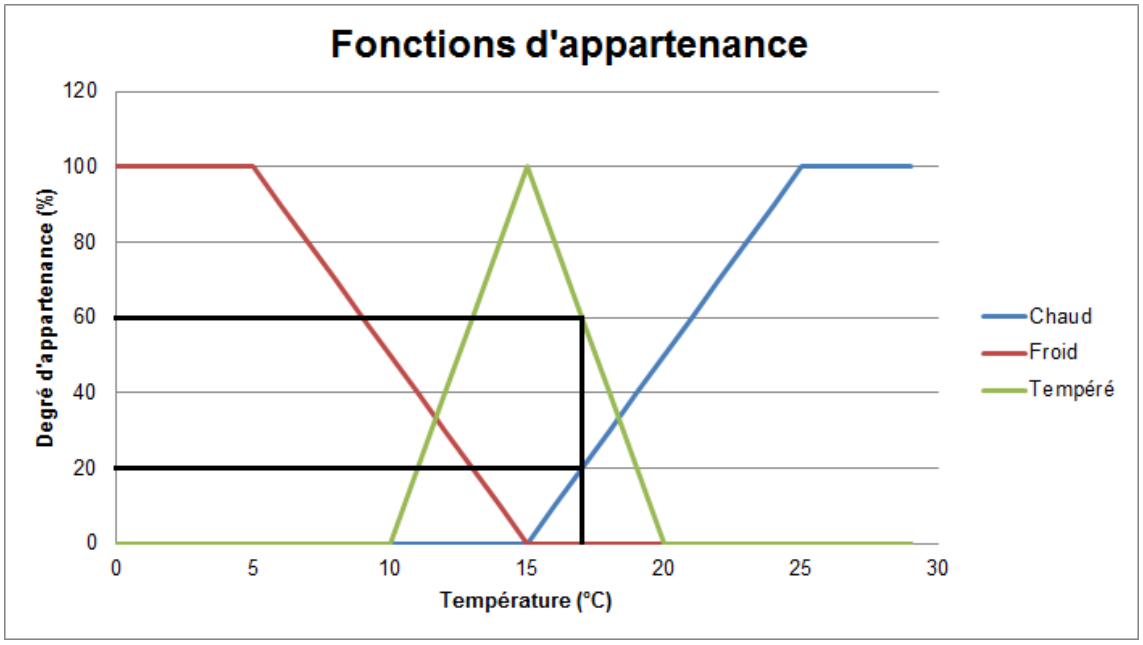

FIGURE 14: Degré d'appartenance de la Température 17°C.

❄ Les Règles d'inférences ou moteur d'inférence :est la deuxième partie du modèle. Le but est d'obtenir de nouvelles variables linguistiques à partir des autres variables linguistiques. Ces nouvelles variables linguistiques ont une relation directe avec les valeurs numériques qu'il faut obtenir à la fin du processus. Les règles sont sous forme :

SI **(X est A)**, *ALORS* **(Y est B)**.

Par exemple, " *SI* **la vitesse est grande ET la distance au feu est courte** *ALORS* **freine fort**" est une règle d'inférence valide.
Les règles d'inférences sont écrite par le concepteur du système flou en fonction de connaissance qu'il possède,ou générer par des programmes spécifiques (MATHLAB , A priori Algorithm , etc.). Une fois la liste de règles d'inférences est dressée , il suffit qu'appliquer chaque règles aux variables linguistiques calculés dans l'étape de fuzzification.(Figure 15)

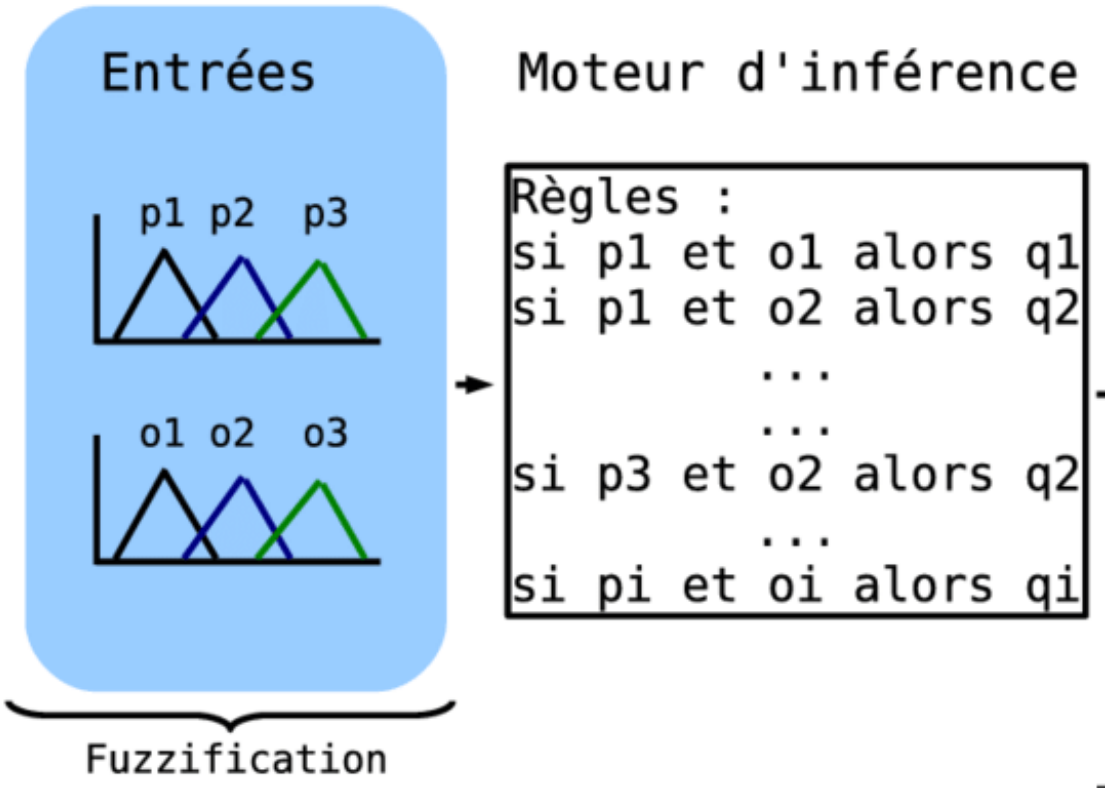

FIGURE 15: Règles d'inférences.

❄ La Défuzzification : est le processus de convertir une valeur floue en valeur nette. Le but est d'obtenir une variable numérique à partir des différentes valeurs linguistiques d'un variable linguistique. (Voir Figure.16)
Il existe plusieurs méthodes pour défuzzifier. Parmi les plus utilisés, on peut citer la méthode de la moyenne des maximas et la méthode du centre de gravité.
L'étape de défuzzification se déroule en deux étapes :

- Fusionner les règles d'inférences qui génèrent plusieurs valeurs de la même variable linguistique.
- Trouver la meilleure valeur quantitative des variables linguistiques en fonction des fonctions d'appartenances.

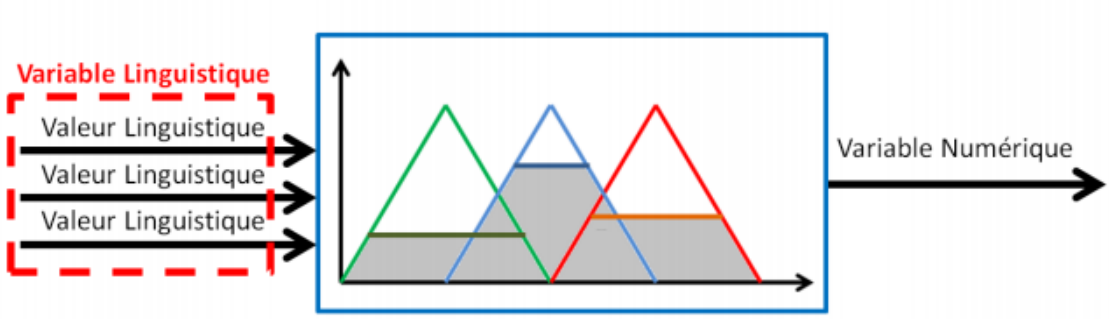

FIGURE 16: Défuzzification d'une variable linguistique de trois valeurs.

5 Les Travaux Accomplis

5.1 Un Filtre Anti Spam Basé sur Le Deep Learning et L'algorithme TF-IDF

Dans Le premier chapitre nous présentons l'algorithme PV-DM et son fonctionnement détaillé.Notre contribution dans ce chapitre est la proposition d'une nouvelle technique de filtrage du SPAM consistant à représenter chaque email en tant que vecteur distribué continu à l'aide du modèle Neural Paragraph Vector-Distributed Memory (PV-DM).

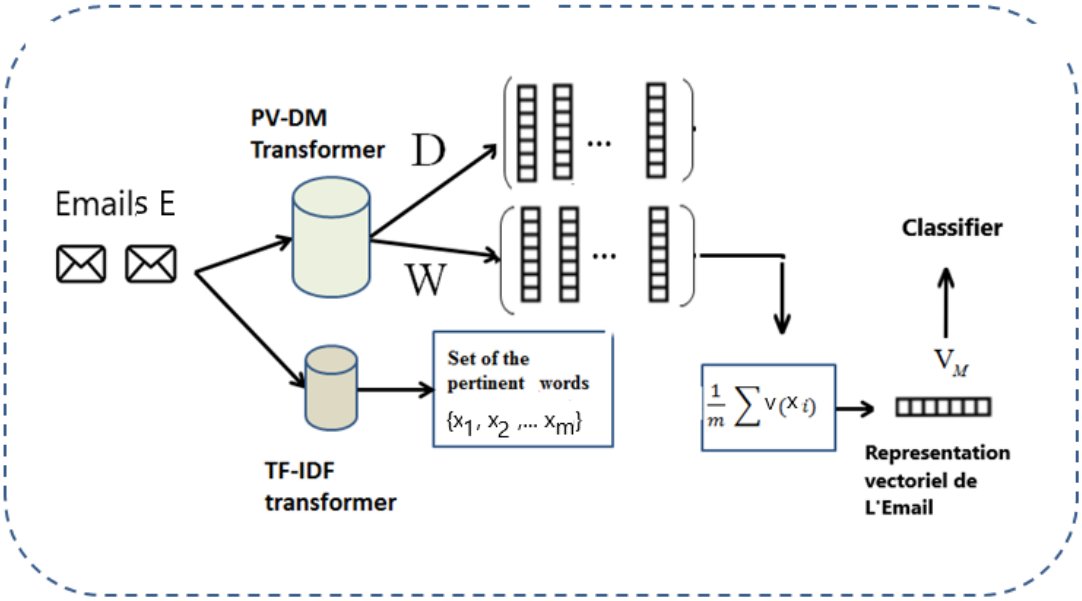

FIGURE 17: Structure de notre Filtre anti Spam

5.2 Extension du Filtre Anti-Spam pour la Filtration des Phishing

En dépit de ses performances, le filtre précédent développé est incapable de détecter un email légitime avec un URL malveillant (ce genre d'email est appelé phishing). D'où la proposition d'étendre ce filtre pour la détection de ces URL. Dans le contexte de phishing, la taille des données est considérable tant en nombre d'instances qu'en nombre de features des URLs considérés,par suite l'augmentation du temps requis pour effectuer la classification.De plus, souvent une partie de ces données ne contient que des informations redondantes, ou inutiles à la tâche de classification, rendant cette dernière plus complexe et non pertinente.Dès lors il est nécessaire de limiter le nombre de données pris en compte, de manière à en extraire l'information discriminante et pertinente améliorant la qualité du classification. En outre, très souvent les spammeurs créent de nouvelles techniques en changeant des features, pour contourner le filtre anti-phishing. Ces changements peuvent perturber le système de détection, créant ainsi un point d'incertitude pour le classificateur. Notre contribution dans ce chapitre est la proposition d'une nouvelle approche de détection et de prévention du phishing en se basant sur Auto Encoder(*AE*) pour réduire la dimension des données en sélectionnant les features pertinents, qui préservent la structure des données d'origine. Pour se prémunir contre les modifications des URLs malveillants, nous appliquons une fonction de corruption ϕ aléatoire sur les données pour introduire le bruit dans les données. Ainsi nous entrainons un Denoising Auto Encoder (*DAE*) à reconstruire les URLs originales et ainsi on réduira le temps de détection du phishing même en cas des changements.

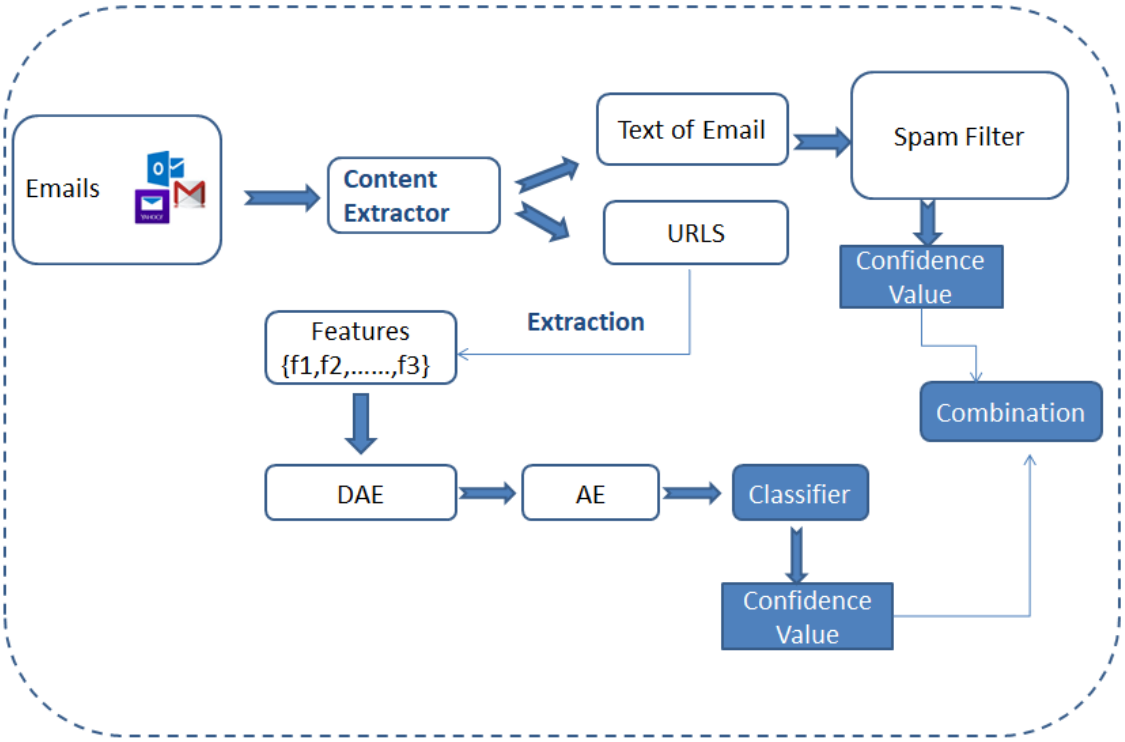

FIGURE 18: Architecture du Filtre anti phishing

5.3 Vers un IDS Learning via Le Model PV-DM et Mutual Information.

Dans ce chapitre nous explorons encore une fois les modèles Deep Learning à savoir PV-DM et l'algorithme de Big Data Analytiques Mutual Information sur des données non textuelles.Les raisons de notre motivation sont :

- une connexion peut présenter ou non une attaque en fonction du contexte dans lequel elle apparaît.

- une attaque peut impliquer plusieurs connexions. Ces dépendances ne peuvent pas être capturées que par les labels des connexions.

Notre contribution dans ce chapitre est la proposition d'un nouvel hybride IDS basé sur la sélection de caractéristiques au moyen de l'algorithme Mutual Information qui lit les paquets du réseau comme un langage naturel en utilisant l'algorithme PV-DM, et qui apprend à détecter le trafic malveillant en employant Random Forest et Logistique Régression en tant que des classificateurs.

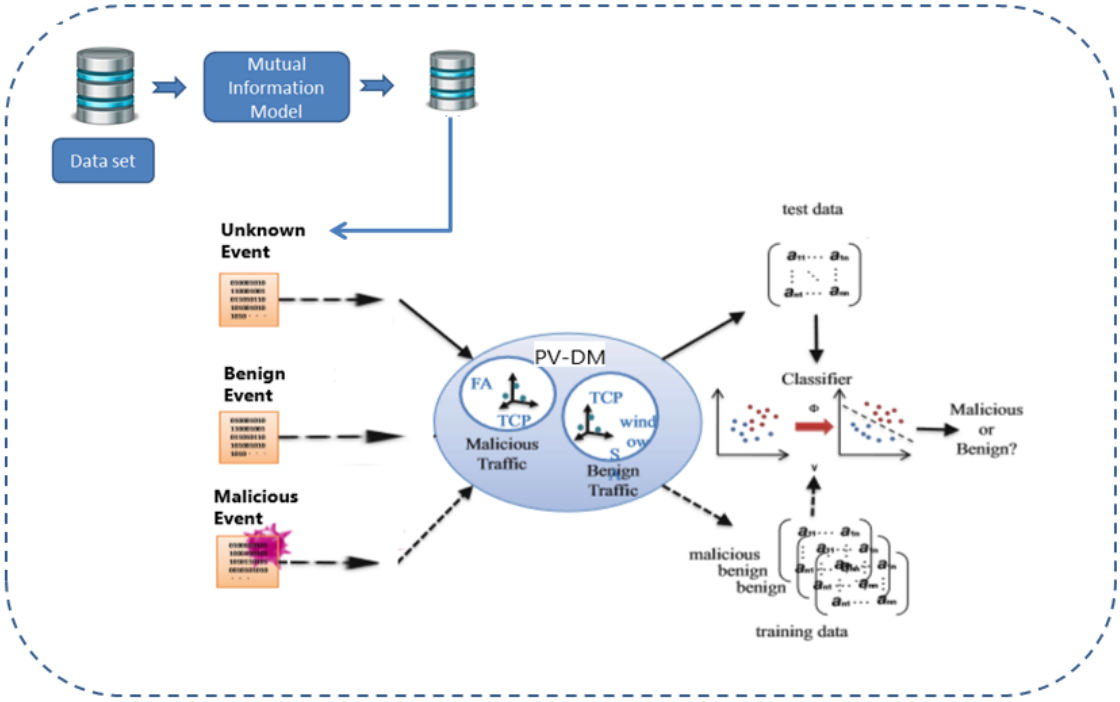

FIGURE 19: Architecture de IDS Learning

5.4 L'Extension de L'IDS Learning avec Fuzzy Logic et L'algorithme Weighted Fuzzy C Mean

Le Quatrième chapitre présente une nouvelle méthode pour la detection des Intrusion en utilisant le Fuzzy Logic(la logic floue) et la méthode du Big Data analytique : Weighted Fuzzy C Mean (WFCM). WFCM va servir à réduire les items impliqués dans l'induction de règles floue sans aboutir à une perte d'information considérable. De plus nous proposons une nouvelle formule de signifiance pour estimer l'importance d'une règle,en intégrant les poids des attributs dans la formule initiale . la motivation derrière ce travail est d'une part que le Fuzzy Logic a la capacité de prendre des décisions rationnelles dans un environnement imprécis, incertain et incomplet. D'autre part, les informations collectées concernant le trafic d'un système informatique s'inscrivent dans ce contexte .

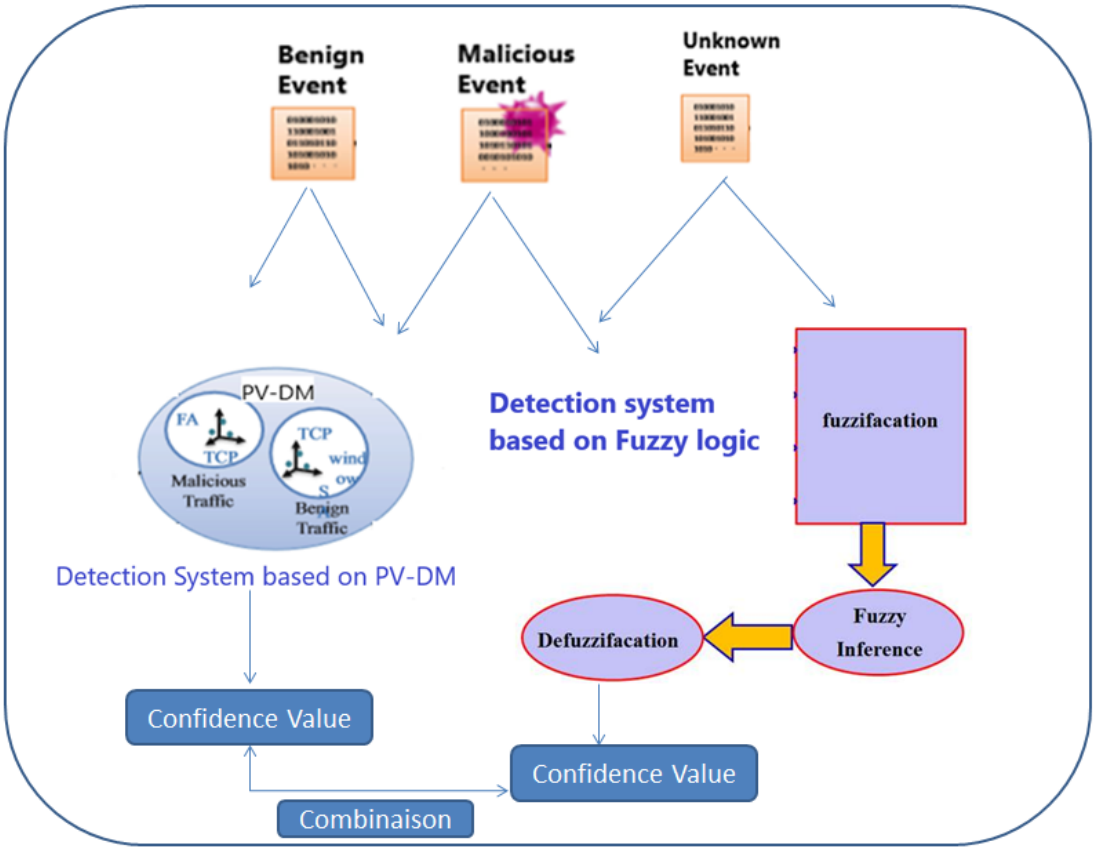

FIGURE 20: Architecture de l'extension IDS Learning

Bibliographie

[1] kaspersky lab report 2018, https ://securelist.com/kaspersky-security-bulletin-2018-statistics.

[2] Chen X-W, Lin X. Big data deep learning : challenges and perspectives. IEEE Access 2014 ;2 :514–25.

[3] https ://fr.wikipedia.org/wiki/Big_data.

[4] Jan.B et al. Deep learning in big data Analytics : A comparative study,journal of Computers & Electrical Engineering. 2017.

[5] Vincent, et al., Extracting and Composing Robust Features with Denoising Autoencoders, Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08), pages 1096 - 1103, ACM, 2008.

[6] Ketul Barot et al ,Using Natural Language Processing Models for Understanding Network Anomalies, IEEE 2016 .

[7] Mamoru Mimura(B) et Hidema Tanaka,Reading Network Packets as a Natural Language for Intrusion Detection, Information Security and Cryptology – ICISC 2017, pp.339-350.

[8] Q. Le et T. Mikolov, Distributed Representations of Sentences and Documents ,Proceedings of the 31 st International Conference on Machine Learning, Beijing,(2014).

[9] Dua, S. and X. Du, Data mining and machine learning in cybersecurity. 2016 : Auerbach Publications.

[10] Cardenas, A.A., P.K. Manadhata, and S.P. Rajan, Big data analytics for security. IEEE Security & Privacy, 2013. 11(6) : p. 74-76.

[11] Cárdenas, A.A., P.K. Manadhata, and S. Rajan, Big data analytics for security intelligence. Available at https ://goo.gl/wxKqDV. 2013 : p. 1-22.

[12] KuppingerCole, B.a., Big Data and Information Security Report. Available at https ://goo.gl/tffZVv. 2016

[13] Chickowski, E., A case study in security big data analysis. Dark Reading, 2012. 9.

[14] Cybenko, G. and C.E. Landwehr, Security Analytics and Measurements. IEEE Security & Privacy, 2012.

[15] Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. Commun. ACM , 37 , 77–84.

Chapitre 2

Filtre Anti-Spam Basé sur Deep Learning et L'algorithme TF-IDF

1 Motivation

L'utilisation du courrier électronique continue de croître parallèlement aux autres méthodes de communication interpersonnelle. En 2017, le nombre total de courriers électroniques professionnels et personnels envoyés et reçus par jour a atteint 269 milliards. Le volume devrait continuer de croître à un taux annuel moyen de 4,4% au cours des quatre prochaines années pour atteindre 319,6 milliards d'ici la fin de 2021[1] voir les figures 1 et 2.
Cependant, le volume croissant du courrier électronique a entraîné l'apparition de problèmes causés par les courriers électroniques appelés Spam.
Le Spam est un message, généralement a caractère publicitaire que l'on reçoit de manière abondante sans avoir sollicité l'expéditeur. Actuellement le Spam par courrier électronique est le type de Spam le plus répandu et le plus gênant. Néanmoins il en existe d'autres types : par message de forum de discussion, par des fenêtres pop up etc. Le Spam est un problème qui non seulement affecte les usagers ordinaires d'internet, mais représente également un souci majeur pour les entreprises et les organisations. Selon Symantec Intelligence Report, le pourcentage global du trafic de courrier électronique défini comme Spam est 71,9% [2].
Les Spam véhicule en plus de la publicité pharmaceutique et pornographique etc, des tentatives d'intrusions, de détournements et de destructions (Virus, Backdoor, phishing etc) ,ce qui nuit à la fiabilité du courrier électronique [3].

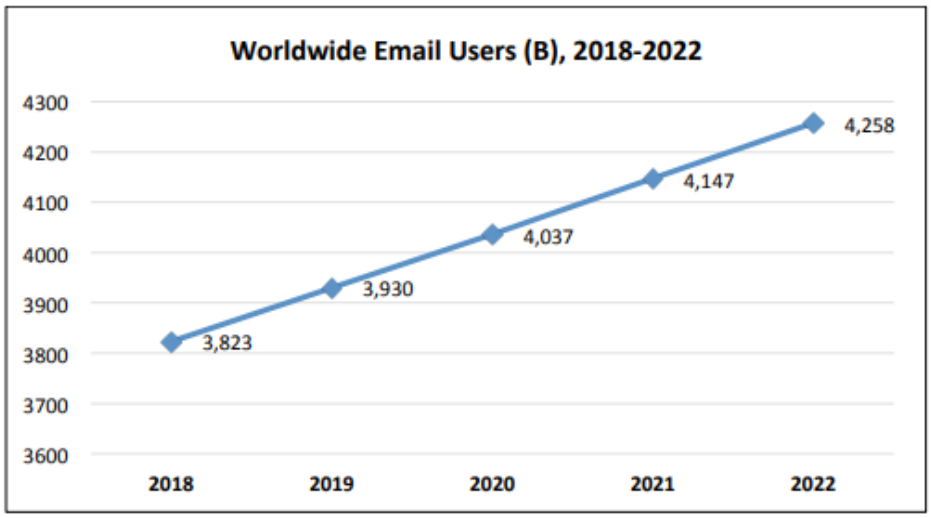

FIGURE 1: Le Nombre des Utilisateurs de courrier électronique À l'échelle mondiale

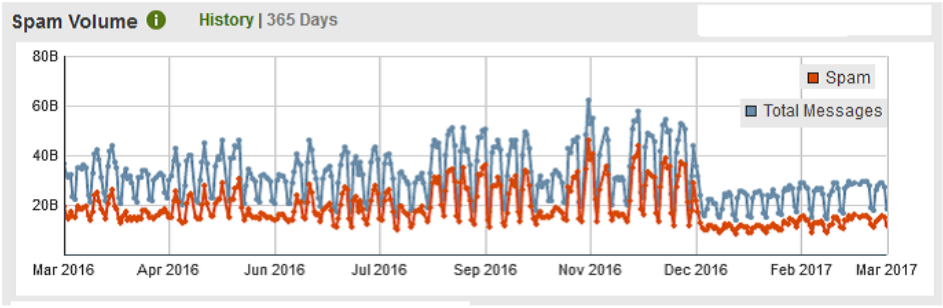

FIGURE 2: Volume global de Spam et de courrier électronique.

2 Les Types Des Spam

2.1 Le Spam avec texte

Les Spams avec du texte sont les Spams dont le contenu contient seulement du texte. On voit dans l'exemple suivant que le contenu de ce Spam est text/plain.

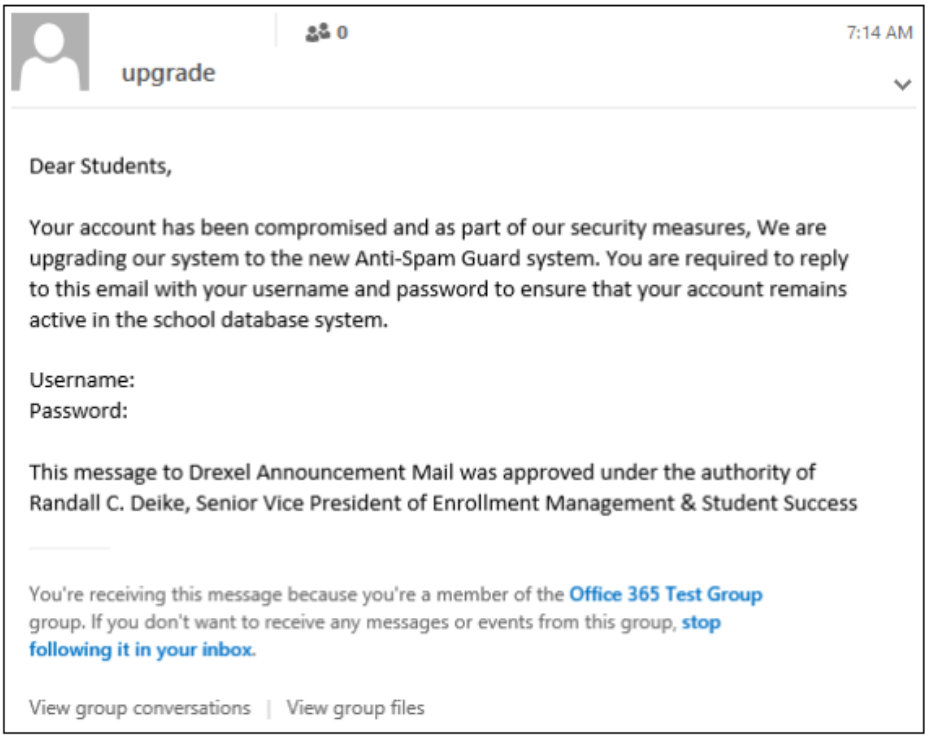

FIGURE 3: Spam texte

2.2 Le Spam avec des images

Le Spam avec des images est un email où le texte de son contenu est stocké au format GIF ou JPEG et affiché dans l'Email. Ceci empêche les filtres de Spam basées sur le texte de se baser sur les mots clés afin de détecter et de marquer ces messages en tant que Spam.

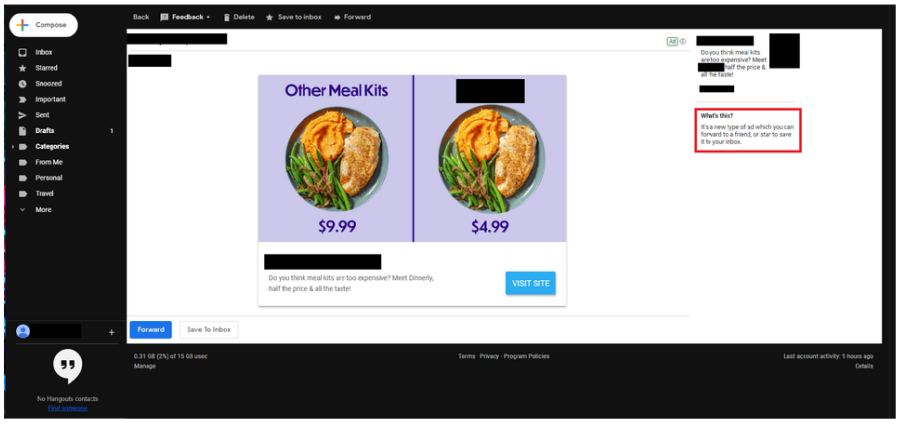

FIGURE 4: Spam image

2.3 Le Spam avec URL

C'est un email communément appelé Phishing ou L'hameçonnage. il consiste à l'origine en des attaques destinées à tromper la victime au moyen de messages électroniques fallacieux ou « maquillés » et de sites Internet frauduleux usurpant le nom d'une banque, d'un commerçant sur l'Internet ou d'une société de carte de crédit, afin d'amener par la ruse les utilisateurs de l'Internet à révéler leurs informations personnelles.

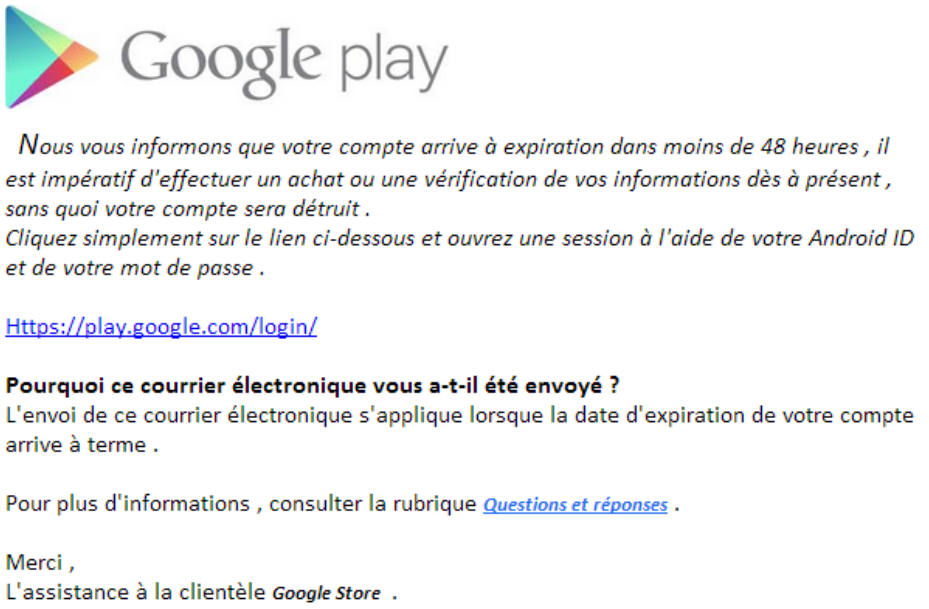

FIGURE 5: Exemple de phishing

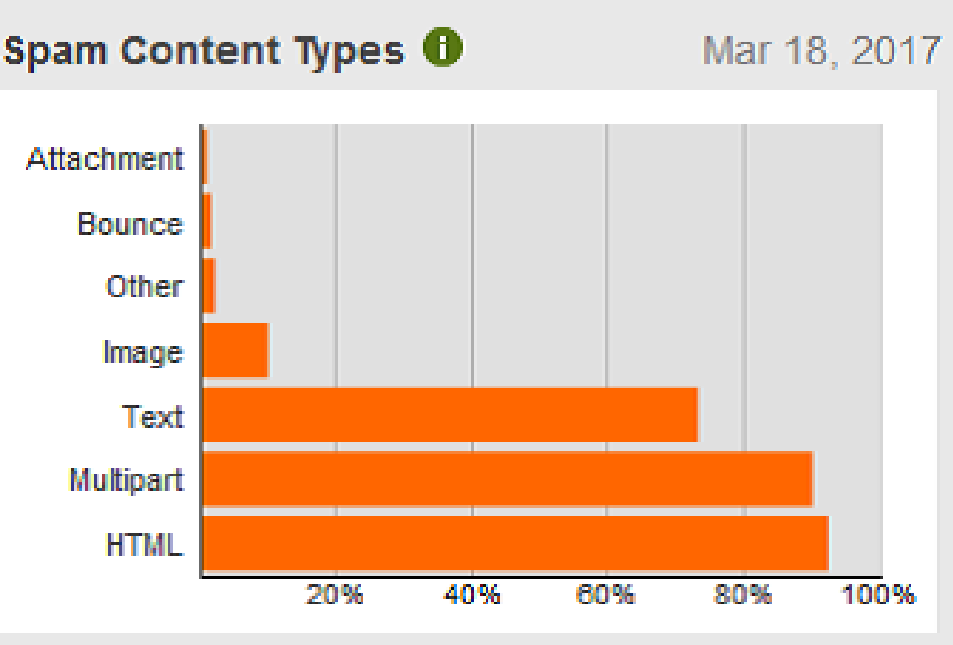

FIGURE 6: Les Types de Spam

3 Les Filtres Anti-Spam

Il existe différents types de filtres anti-spam portent sur différents éléments de L'Email tels que l'objet, le contenu ou encore l'expéditeur de l'Email.

3.1 Le Filtre des Entêtes

Ce filtre s'applique uniquement à l'entête du message et ne s'attarde pas au contenu du mail. Cette technique présente l'avantage de pouvoir bloquer les Emails avant même que leur contenu ne soit envoyé. Le taux de faux négatifs dans ce type de filtrage est quasiment nul. Mais, Le taux

d'efficacité de ce type de filtre est peu suffisante()environ 50%)[4], vu que L'entête du message ne contient pas souvent assez d'informations pour pouvoir incriminer un Email .

3.2 Les Filtres de Contenu

Ce filtres sont base sur une L'une des approches les plus populaires utilisées dans les filtres anti-spam à savoir : la représentation Bag of Word (BoW) également connue sous le nom de modèle d'espace vectoriel [5].
cette approche construit une liste des mots clés à détecter afin de déterminer qu'un Email est un Spam. Par exemple, tous les Emails qui contiennent les mots : sexe, sexy, viagra, argent, money, drogue seront détectés comme Spam. Ce type de filtrage est très rapide, mais peux efficace pour les raisons suivants :

- les Spammeurs font souvent varier les mots clé afin d'éviter ces filtres. Par exemple, on retrouve V.I.A.G.R.A. ou encore Vi a gra au lieu du Viagra ou Fr33 à la place de Free.
- de plus , obtenir un nombre de faux positifs élevé [4]. Prenons par exemple le cas du mot clé sexe, il peut très bien être utilisé légitiment dans le cas d'une demande d'information complémentaire à un candidat « Nom : Prénom : Sexe : ».

4 Travaux Connexes

Pour faire au problème de Spam et remédier aux problèmes de BoW, plusieurs méthodes ont été proposées : Elisabeth Crawford et al [9], par exemple, ont proposé une représentation basée sur les phrases au lieu du BoW , pour augmenter les performances des classificateurs de courrier électronique.
Matthew Chang et Chung Keung Poon [10] ont étudié l'utilisation de phrases en tant que fonctionnalités de base pour la classification des emails. Ils utilisent trois classificateurs différents, à savoir Bayes, K-NN, et l'algorithme TF–IDF. Ils ont constaté que l'utilisation de phrases de la taille de deux donne généralement les meilleurs résultats de classement. Cependant le problème dans cette approche réside non seulement dans la grande dimensionnalité, mais aussi dans leur notion de phrases qui est limitée à une séquence de mots consécutifs de longueur fixe (max. 4 mots), tout en ignorant la sémantique des mots ou leurs contexte.
Woitaszek et al. [11] ont utilisé le classificateur SVM pour construire un système de classification automatisé permettant de détecter les emails commerciaux non sollicités. Dans leur étude, plusieurs ensembles d'échantillons de courriers ont été rassemblés pour créer des dictionnaires de mots trouvés dans les communications électroniques.
Johan Hovold [12] suppose qu'il est possible d'obtenir de très bonnes performances de classification en utilisant une variante de Naïve Bayes basée sur la position des mots.
Kanaris et al. [13] Utilisent n-grammes pour produire des mots plus robustes, pour le filtrage des emails. Sahami et al. (1998) [14] ont proposé un filtre de messagerie basé sur un classificateur Naïve Bayes amélioré. Les résultats expérimentaux ont confirmé l'amélioration escomptée au niveau de la précision et le rappel à la suite de l'adoption des phrases comme des fonctionnalités de classification.
Bien que ces méthodes de filtrage ont conduit à des résultats de detection satisfaisantes,ils souffrent toujours du problème de dimensionnalité ,et du taux de faux positifs élevé.[4]

5 Background

5.1 Bag of Word

La méthode Bag of Word (BoW) est une approche populaire rapide et facile à mettre en œuvre, et très utilisée en recherche d'information. Dans le modèle BoW, les documents sont représentés par des vecteurs dans laquelle chaque dimension correspond à un mot, générant ainsi des vecteurs avec une très haute dimensionnalité. C'est une représentation exclusivement lexicale qui repose sur la fréquence pour déterminer la valeur associée à chaque dimension du vecteur. Le principe de Bag of Word se résume en 4 phases :

❄ La tokenisation : c'est une étape indispensable en Bag of Word. Elle consiste à découper les phrases du jeu de données en des mots isolés (voir figure 6) . En python ,Scikit-Learn package aide dans cette étape avec ses fonctions de tokenisation en particulier *CountVectorizer()* et *TfidfVectorizer()*.

❄ La Lemmatisation : cette étape consiste à supprimer tout les chiffres,les ponctuations, symboles et les stopwords, et passer à la fin tout les mots restant en minuscule.

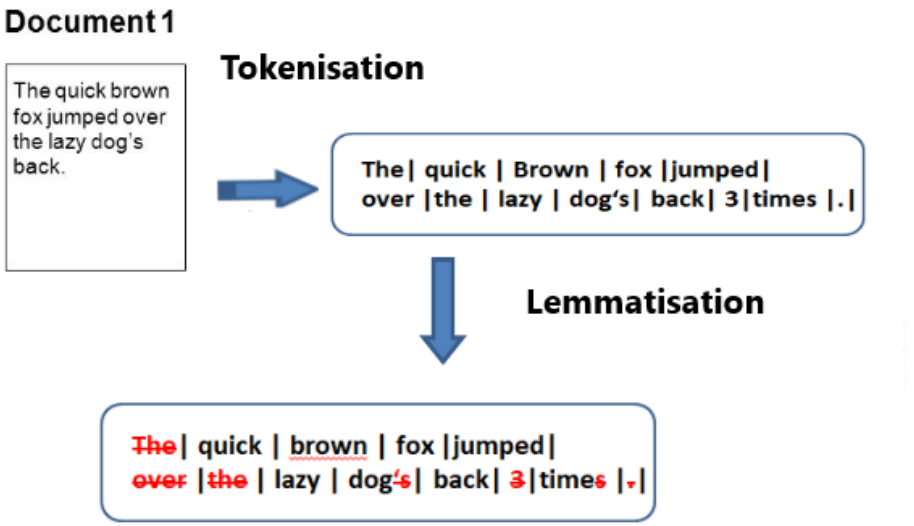

FIGURE 7: Exemple de la phase tokenisation et la Lemmatisation d'une texte.

❄ La constitution d'un Dictionnaire global qu'on appelle le Vocabulaire .Ce Dictionnaire est constitué par les mots (uniques) obtenus par la Lemmatisation .les mots dans le Vocabulaire seront ordonnés par ordre alphabétique.

❄ L'encodage : chaque mot du vocabulaire sera associé à à un vecteur binaire de dimension |V| (nommé one Hot Vecteur) , $v = (0,0,0...1,0...0)$ où le nombre 1 est à la position du mot dans le dictionnaire et $|V|$ est la taille du Vocabulaire.

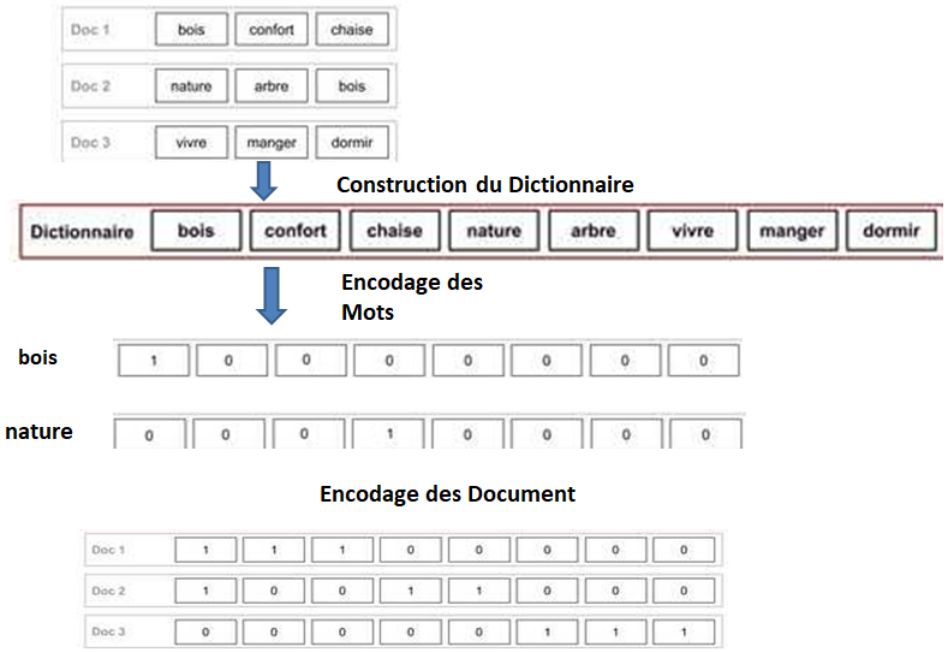

FIGURE 8: Exemple de construction du Dictionnaire et encodage des mots par le model BoW

L'algorithme Bag of Word est appliqué largement dans plusieurs applications à savoir le traitement du langage naturel , la récupération d'informations (Information Retrieval), la classification et la reconnaissance des images ,Vision par ordinateur etc.
Cependant BoW soufre de quelques limitations :

- Comme le Vocabulaire peut potentiellement atteindre des millions d'unités, les représentations vectorielle des mots seront hyper-dimensionnelle .

- Le model BoW ne tient pas compte de la similitude conceptuelle entre les termes. Par exemple, les mots "voiture" et "automobile" sont souvent utilisés dans le même contexte. or, les vecteurs correspondant à ces mots sont différents dans le modèle de BoW, ce qui pourrait conduire à des mauvaises performances au niveau de la classification.

- En modélisant des phrases en utilisant Bag of Word, l'ordre des mots dans la phrase n'est pas respecté. Ex : "Ceci est gratuit" et "Est-ce gratuit " ont exactement la même représentation vectorielle, et par conséquent, différents emails pourront avoir la même représentation puisque les mêmes mots sont utilisés.

- Enfin, le model BoW suppose que le dictionnaire initialement sélectionné sera toujours représentatif pour la classification d'un email. Or, il a été prouvé que cela est faux [4], car les spammeurs essaient continuellement de créer de nouveaux moyens de surmonter les filtres anti-spam. Par dissimuler certains termes qui sont très courants dans les messages de spam, par exemple, en écrivant «fr33 » au lieu de « free » ou « mon3y » au lieu de « money », comme tentative d'empêcher l'identification correcte de ces termes par les filtres anti-spam.

5.2 Word Embedding

L'idée clé du modèle de Word embedding est de représenter les concepts sous forme de vecteurs, les relations binaires sous forme de matrices et l'opération consistant à appliquer une rela-

tion à un concept sous forme de multiplication matrice-vecteur produisant une approximation du concept associé [15]. Le vecteur Word Embedding est calculé à l'aide de réseaux de neurones en procédant comme suit :

- ❄ La Couche d'entrées ou Input Layer : Dans cette couche , chaque mot du vocabulaire V est associé à son one hot vecteur de dimension $|V|$ (dimension de vocabulaire).

- ❄ La Couche de projection ou le Hidden layer : Dans cette couche le vecteur one-hot du mot est multiplié par une matrice $W \in R^{|V|*N}$. où W est le Word Embedding matrice initialisé aléatoirement , $|V|$ est la dimension du vocabulaire et N est la dimension du Word Embedding vecteur du mot qu'on veut calculer(généralement $20 \leq N \leq 100$). En parcourant tout le corpus, et à chaque étape t , le vecteur de mot cible et la matrice W sont mis à jour pour que les mots similaires se rapprochent dans l'espace vectoriel. Après de nombreuses étapes, les vecteurs deviennent significatifs, ce qui permet d'obtenir les mêmes vecteurs pour les mots similaires [19] [7].

- ❄ La Couche de sortie : Cette couche prend la sortie de la couche de Projection multiplié par une matrice W' (initialisée aussi aléatoirement) et crée une distribution de probabilité à l'aide d'une fonction Softmax sur les mots constituant le vocabulaire.

La figure 8 montre l'architecture générale de word embedding . où $X = \{x_1, ...x_|V|\}$ représente les vecteurs d'entrées , $H = \{h_1, ..h_N\}$ les projections des entrées dans la couche cachées et $Y = \{y_1 ...y_|V|\}$ représente les sorties calculées grâce à la fonction Softmax .

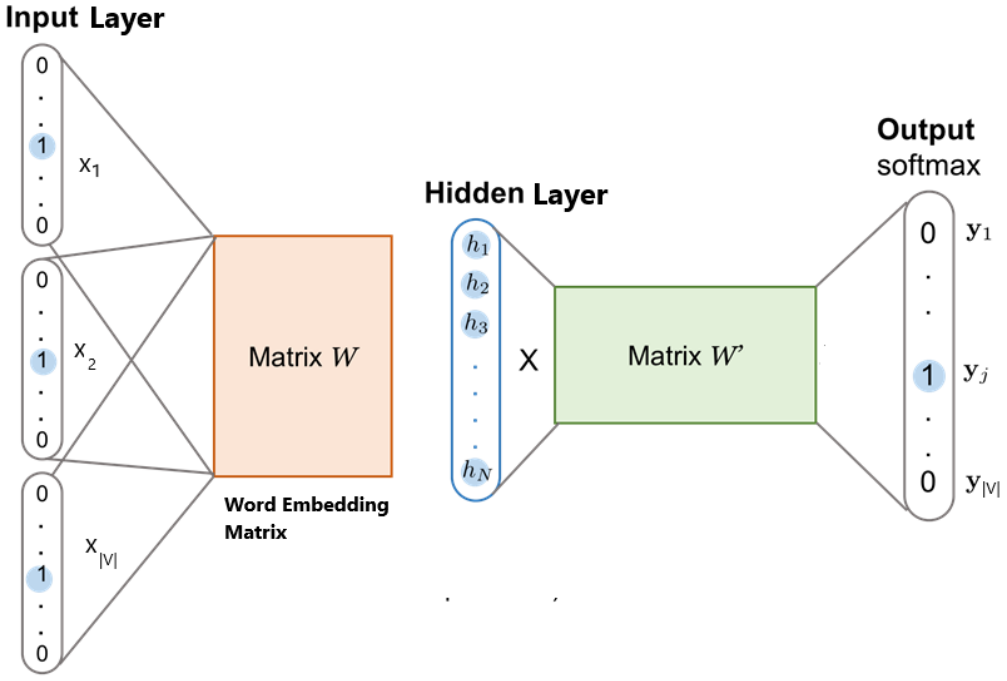

FIGURE 9: Architecture de word embedding

Les Word Embedding représentations possèdent des capacités surprenantes. Par exemple, on peut retrouver beaucoup de régularités linguistiques simplement en effectuant des translations linéaires dans cet espace de représentation [16] [17] [18]. Par exemple le résultat de vecteur("Washington") - vecteur("U.S.A") + vecteur("France") donne une position dont le vecteur le plus proche est vecteur ("Paris") voir Figure 9.

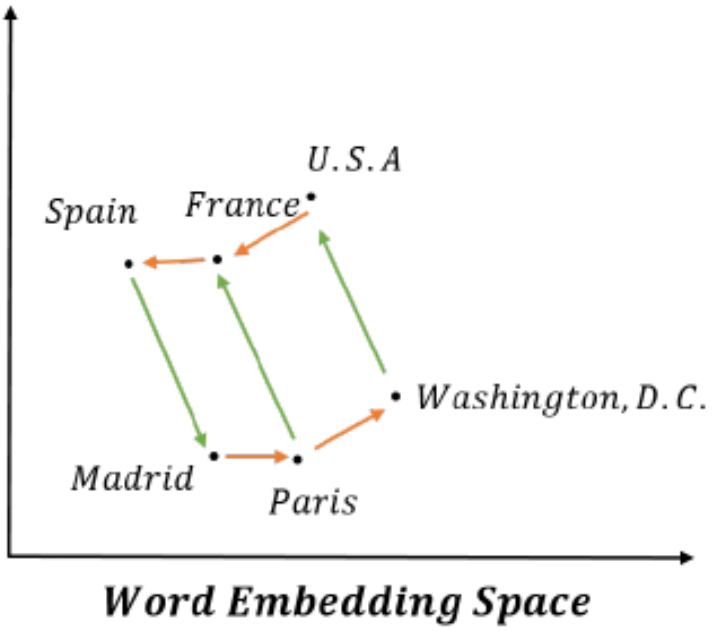

FIGURE 10: Exemple du calcul vectoriel dans l'espace de word Embedding

Le Word Embedding a montré aussi de bonnes performances dans divers domaines tels que la traduction machinale [16], la mesure de la similarité sémantique, la désambiguïsation de l'acronyme et la reconnaissance vocale [18].

5.3 Word2vec

Word2Vec [7] [19] est un modèle non supervisée basée sur le Word Embedding. Il utilise des perceptrons linéaires simples avec une seule couche cachée. L'idée est de compresser le corpus vers un dictionnaire de vecteurs denses de dimension bien inférieure choisie reflétant les relations entre les mots et leur contexte. le modèle Word2vec est le modèle le plus populaire de Word embedding '[8] [9], il existe en deux variantes : Continuos Bag of Word (CBOW), qui prédit un mot cible à partir de son contexte, et skip-gram, qui prédit les mots de contexte à partir d'un mot cible donné. La fonction de perte du modèle est minimisée lorsque le modèle produit des probabilités élevées pour les mots qui font partie du contexte du mot cible, et des probabilités faibles pour d'autres mots qui sont considérée comme bruit.(voir chapitre : Introduction Générale).

5.4 Le Model PV-DM

Le PV-DM est un algorithme développé par Google [5] [19]. C'est une technique basée sur les réseaux de neurones qui convertit les mots en des vecteurs réels, en tenant compte de l'ordre des mots, de telle sorte que les mots similaires sémantiquement et syntaxiquement sont étroitement positionnés dans l'espace. Ces vecteurs peuvent ensuite être utilisés comme caractéristiques pour des algorithmes de classification et d'apprentissage automatique.

Le PV-DM est inspirée du travail sur Word embedding , notamment le modèle CBOW du modèle Word2vec. L'idée centrale de PV-DM est qu'un paragraphe P peut être représenté comme un vecteur contribuant à la prédiction du mot manquant dans une phrase.
Étant donné un ensemble de mots, $x_{n+2}, x_{n+1}, x_{n-1}, x_{n-2}$ dans une paragraphe P où le mot x_n est manquant. PV-DM prédit le vecteur du mot absent $V(x_n)$ en tenant compte des autres mots vecteurs $v(x_{n+2}), v(x_{n+1}), v(x_{n-1}), v(x_{n-2})$ ainsi que du vecteur de paragraphe $V(P)$. Le vecteur de paragraphe $V(P)$ représente le contexte du mot que nous essayons de prédire. Voir Figure 10 .

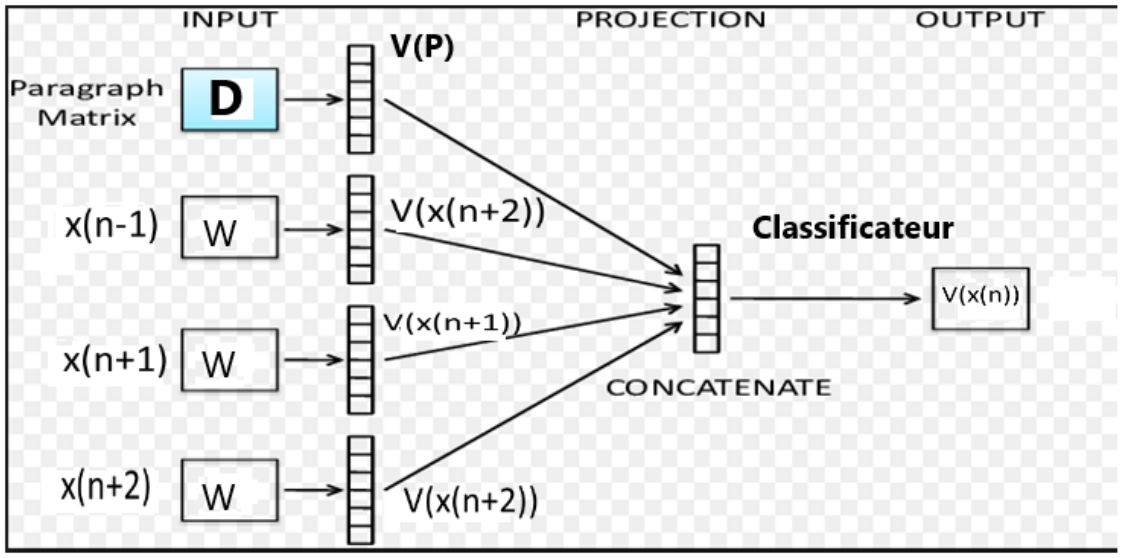

FIGURE 11: le framework de PV-DM pour l'apprentissage des vecteurs de paragraphes

Dans le modèle PV-DM, chaque paragraphe est représenté par un vecteur unique sous forme une colonne dans la matrice D et chaque mot est également représenté par un vecteur unique, sous forme une colonne dans la matrice W (Figure 10).Le vecteur de paragraphe et les vecteurs de mots sont ensuite concaténés, et la prédiction est effectuée via un classificateur, tel que Softmax, où nous avons :

$$p(v(x_n)|v(x_{n+2})...(x_{n-2}),P,W,D) = \frac{e^{y_{v(x_n)}}}{\sum_i e^{y_{v(x_i)}}} \quad (1)$$

Chaque $y_{v(x_i)}$ est calculé comme suit :

$$y = b + U * h(v(x_n)|v(x_{n+2}),....v(x_{n-2}),P,W,D) \quad (2)$$

où b et U sont les paramètres de Softmax et h est construit à partir de W et D.

Le PV-DM ou(le Paragraphe vecteur) permet de calculer la similarité sémantique entre deux documents et d'inférer des documents similaires sémantiquement. Certaines implémentations prennent également en charge l'inférence de l'incorporation de documents dans des documents non vus [9] .

5.5 Terme Fréquence -Inverse Document Fréquence (TF-IDF)

TF-IDF sont les acronymes de « Terme Frequency » et « Inverse Document Frequency ».c'est une méthode de pondération souvent utilisée en recherche d'information et en particulier dans la fouille de textes. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un Email, relativement à une collection ou un corpus des Emails. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le courrier. Il varie également en fonction de la fréquence du mot dans le corpus des Emails.

La méthode TF-IDF repose sur le schéma de pondération suivant :

- La Fréquence du terme TF(x_{ij}) : c'est simplement le nombre d'occurrences d'un terme x_i dans l'Email j.

- Fréquence inverse de document (IDF(x_i)) :La fréquence inverse de document est une mesure de l'importance du terme x_i dans l'ensemble du corpus. Dans le schéma TF-IDF, elle vise à donner un poids plus important aux termes les moins fréquents, considérés comme plus

discriminants. Elle consiste à calculer le logarithme de l'inverse de la proportion de Emails du corpus qui contiennent le terme :

$$\mathrm{IDF}(\mathrm{x_i}) = \log \frac{|\mathrm{E}|}{|\{\mathrm{e_j} : \mathrm{x_i} \in \mathrm{e_j}\}|} \tag{3}$$

où :

- |E| : nombre total de documents dans le corpus.
- $|\{e_j : x_i \in e_j\}|$: nombre de documents où le terme x_i apparaît.

❄ Finalement, le poids d'un terme dans un Email s'obtient en multipliant les deux mesures TF et IDF :

$$TF - IDF(x_{ij}) = \mathrm{TF}(\mathrm{x_{ij}}) \cdot \mathrm{IDF}(\mathrm{x_i}) \tag{4}$$

Dans la Figure 11 un exemple du calcul de TF-IDF des termes dans des documents différents.

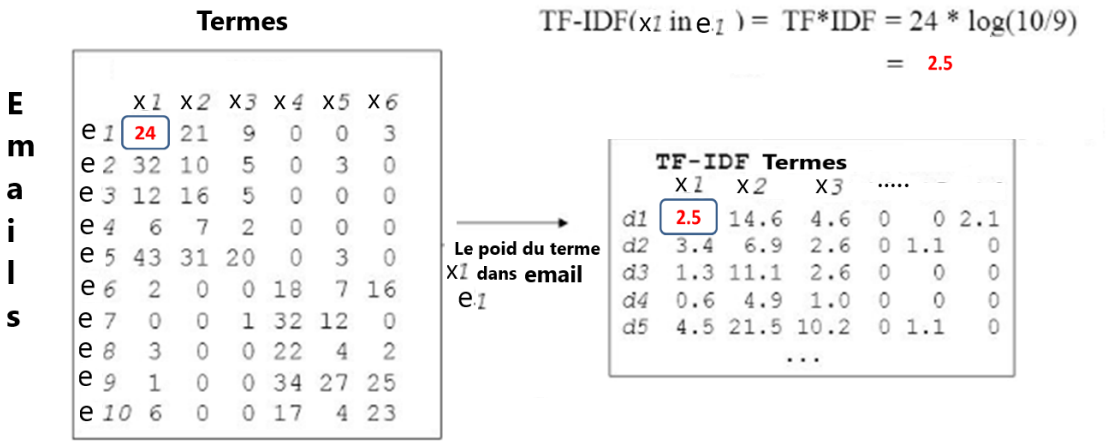

FIGURE 12: Un exemple du calcul du TF-IDF des termes

La mesure de TF-IDF va nous permettre d'évaluer l'importance d'un mot contenu dans un Email , relativement au corpus des Email utilisé dans notre expérience .

6 Solution Proposée

Notre solution vise à surmonter les désavantages de BoW, notamment le fait qu'il ignore l'ordre et la relation entre les mots. La présente étude propose la représentations d'un Email en se basant sur TF-IDF et PV-DM algorithme. La première méthode(TF-IDF) sélectionne les termes pertinent de chaque Email , et génère les vecteurs des termes en utilisant le contexte global du courrier électronique. Ce faisant, nous essayons de fournir une représentation qui capture les informations extraites à la fois du contexte local des mots et du contexte global de chaque courrier électronique.

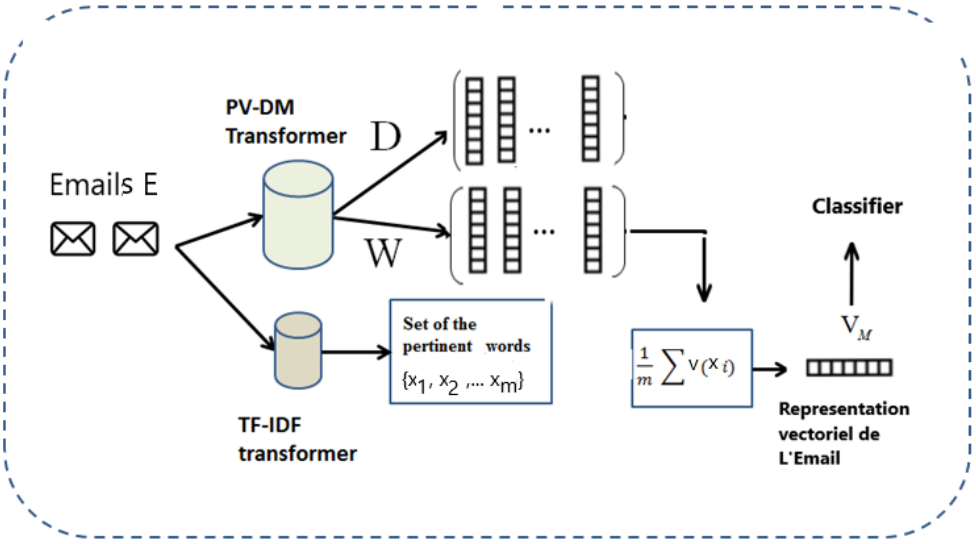

FIGURE 13: Architecture de notre solution

6.1 La phase D'entrainement

La phase d'entrainement de Notre approche consiste à construire la représentations vectorielles pour chaque Email à l'aide du modèle d'apprentissage profonde PV-DM et la méthode TF-IDF, en procédant comme suit (Voir la figure 12), Étant donné un corpus des Emails d'entrainement $E = \{e_1, e_2 ... e_n\}$:

❄ Application du modèle PV-DM sur E et Comme nous l'avons cité ci- dessus, le modèle PV-DM génère deux matrices :

- La Matrice D où chaque colonne représente un vecteur d'un courrier électronique.
- La Matrice W où chaque colonne est un vecteur qui représente un mot du courrier. Voir Figure 13.

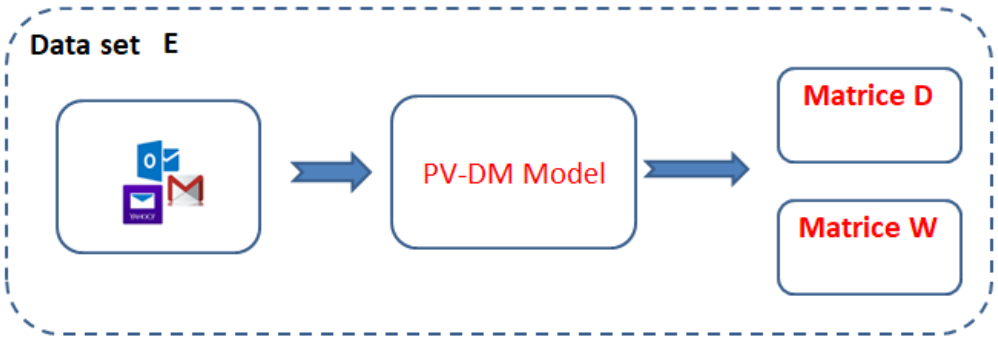

FIGURE 14: Application du PV-DM sur le corpus d'entrainement

❄ Application de la méthode TF-IDF sur le corpus d'entrainement pour capturer les mots pertinents ayant une valeur TF-IDF supérieur d'un seuil déterminé.Figure 14

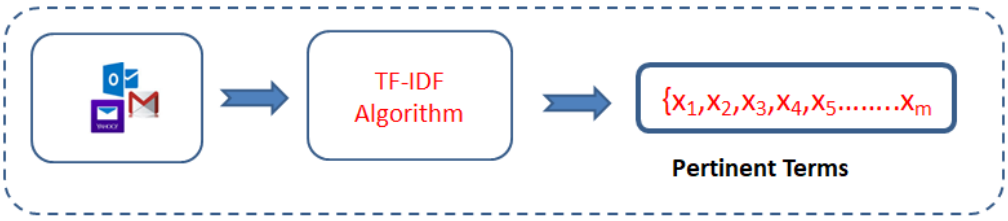

FIGURE 15: Application du TF-IDF sur le corpus d'entrainement

❄ Ensuite, nous allons extraire les représentations vectorielles correspondantes au termes sélectionnées par TF-IDF à partir de la matrice W .

❄ Enfin , nous calculons la moyenne arithmétique des vecteurs sélectionnées précédemment,le vecteur résultant est utilisé comme représentation vectorielle de l'Email nous appelons ce vecteur moyen V_M. Figure 15

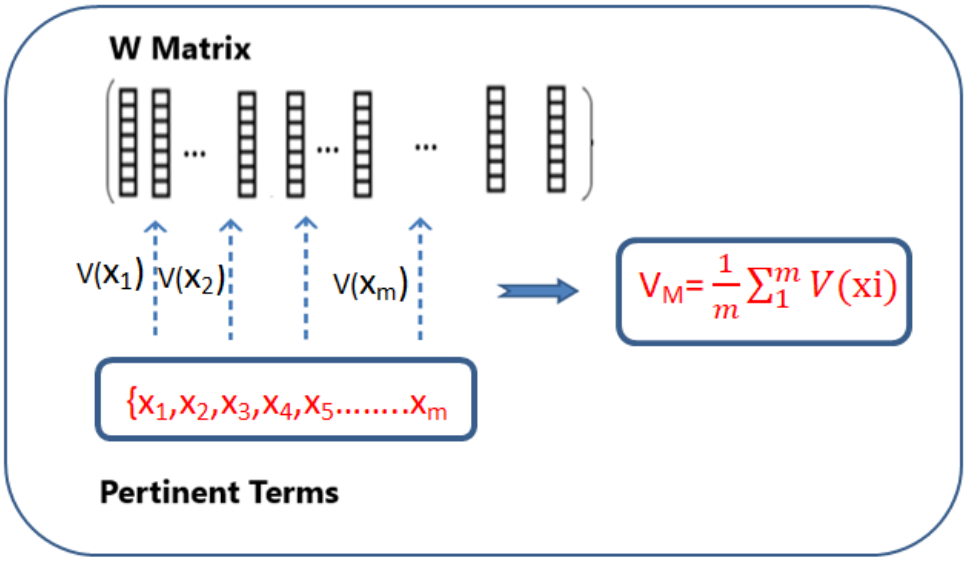

FIGURE 16: Calcul du représentation vectorielle d'un Email

Les représentations vectorielles des Emails résultantes seront les entrées d'entrainement des classificateurs comme *Logistic Regression , SVM* etc .

7 le Design Expérimental

7.1 Data Sets

Pour la classification, nous avons appliqué l'approche proposée sur deux bases de données. La première est la base de données **Enron** qui est utilisée dans plusieurs documents de recherche pour la classification des courriers électroniques [21] [22] [7] [23] . Elle est composé de 33 702 emails au total, nous fusionnons tous les messages d'Enron en un seul corpus. Ces emails sont divisés de manière aléatoire en un ensemble d'entrainement (26960 emails) et un ensemble de test (6742 emails).
Le deuxième ensemble de données que nous avons utilisé est le corpus de **Ling spam**, qui contient 2892 emails au total. Cet ensemble de données a été divisé comme suit : 2314 courrier en tant que base d'entrainement et 578 emails en tant que base de test.

	Training Data		Test Data	
Data set	Ham	Spam	Ham	Spam
Enron Data set	13237	13723	3308	3434
Ling Spam corpus	1930	384	482	96

TABLE 1: Répartition des bases de données pour l'expérience

7.2 Métriques de performance

Nous utilisons cinq métriques d'évaluation les plus couramment utilisés pour mesurer les performances de la méthode de filtrage proposée dans ce chapitre : Recall, Accuracy, Precision et F-score. Ces métriques sont calculées en utilisant les indices de la matrice de confusion (TP, FP, FN et TN) . Voir Tableau 2 :

La classe reel de l'email	classé comme spam	classé comme Ham
Spam	True Positif(TP)	False Negatif(FN)
Ham	False Positif(FP)	True Negatif(TN)

TABLE 2: Les Indices de la Matrice de Confusion

7.2.1 Métriques de Performances

❄ Recall : est défini comme la probabilité de classer correctement les emails de spam.

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

❄ Precision : mesure la précision de la méthode de filtrage pour classer correctement les spams

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

❄ Accuracy : c'est la capacité de la méthode de filtrage à classer correctement les emails légitimes et les emails indésirables.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

❄ F-score :une mesure populaire qui combine la Precision et le Recall en calculant leur moyenne harmonique. Cette mesure représente le fait qu'il est plus important de classer les courriers indésirables comme courrier désirable plutôt que de filtrer tout le courrier indésirable.

$$F-score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

❄ la Courbe Receiver Operating Characteristic de (ROC) : est un graphe bidimensionnel où l'axe des ordonnées represente le True Positive rate (sensitivity) et l'axe des abscisses represente False Positive rate (1-specificity). L'utilisation des courbes ROC présente l'avantage de ne pas être sensible aux modifications de la répartition des classes. Si le rapport entre les échantillons positifs et négatifs dans la base de test est différent de la relation trouvée dans la base d'entrainement, les courbes ROC restent les mêmes [23].

❄ AUC Area Under the Curve (AUC) : est un autre outil utilisé pour représenter l'efficacité d'un algorithme en fournissant une valeur scalaire, qui correspond à l'aire sous la courbe ROC. Plus l'AUC est élevé, plus l'algorithme est meilleur.

7.3 A propos de L'implémentation

Pour obtenir les matrices D et W on a utilisé le module Doc2vec de Genism toolkit [20]. Le module est implémenté en Python, et on l'a entraîné avec les paramètres suivants :

Paramètre	Value
Size	100
Window	5
Epoques d'entraînement	25
Taux d'apprentissage	0,025

TABLE 3: Les paramètres de PV-DM utilisés pour L'expérience

Pour la méthode TF-IDF nous avons utilisé l'implémentation de scikit-learn python Library de l'algorithme TF-IDF [21] avec les paramètres par défaut et le nombre de mots pertinent égal à 1000 pour la base Ling spam et 1500 mots pour la base Enron [22] .

8 Résultats Expérimentaux

Afin d'évaluer les performances de notre approche, nous avons comparé notre méthode avec les deux méthodes de représentation les plus connues : le modèle PV-DM et le modèle BoW. Les trois modèles sont entrainés avec différents classificateurs sur les bases de données Ling SPAM et Enron corpus. Nous avons aussi tracé les courbes ROC et calculé l'AUC scores réalisé par les différents classificateurs . Les résultats obtenus sont présentés dans les tableaux suivants :

8.1 Performances de notre Solution vis à vis Les Modèles PV-DM et BoW sur Ling Spam corpus

classifier . Name	AUC value Solution proposée %	AUC value PV-DM%	AUC value BoW %
SVM	0.9604	0.9097	0.5003
MPL	0.9578	0.9139	0.6627
KNN	0.9413	0.8663	0.5740
LR	0.9459	0.8997	0.7178

TABLE 4: AUCS de différents classificateurs entrainés avec notre Solution , PVDM , et BoW sur le corpus Ling spam

classifier	Model	Accuracy%	Precision%	Recall%	F-score%
SVM	Solution proposée	0.9827	0.9797	1	0.9897
	PV-DM	0.9619	0.9582	0.9979	0.9776
	Bow	0.8463	0.8648	0.9669	0.9130
KNN	Solution proposée	0.9827	0.9797	1.0	0.9897
	PV-DM	0.9756	0.9756	0.9937	0.9740
	Bow	0.8238	0.8685	0.9296	0.8980
LR	Solution proposée	0.9827	0.9797	1.0	0.9897
	PV-DM	0.9619	0.9618	0.9938	0.9805
	BoW	0.8342	0.8342	1.0	0.9096

TABLE 5: Comparaison entre les performances de notre Solution avec les modeles PV-DM et Bow entrainés et testés sur le corpus Ling Spam

8.2 Performances de notre Solution vis à vis les modèles PV-DM et BoW sur Enron Dataset

classifier	AUC value du Solution proposée%	AUC value de PV-DM%	AUC value de Bow%
SVM	0.9604	0.9097	0.6032
MPL	0.9792	0.9646	0.5198
KNN	0.9479	0.9344	0.6106
LR	0.9479	0.8979	0.5

TABLE 6: AUCS de différents classificateurs entrainés avec notre Solution, PVDM , et BoW sur le corpus Enron

classifier	Model	Accuracy%	Precision%	Recall%	F-score%
SVM	Solution proposée	0.9616	0.9655	0.9559	0.9607
	PV-DM	0.9111	0.9293	0.8863	0.9073
	Bow	0.5094	1.0	0.0006	0.0012
KNN	Solution proposée	0.9307	0.9435	0.9135	0.9283
	PV-DM	0.8667	0.8716	0.8540	0.8627
	BoW	0.5736	0.5626	0.5905	0.5762
LR	Solution proposée	0.9588	0.9644	0.9510	0.9576
	PV-DM	0.9027	0.9063	0.8942	0.9002
	BoW	0.7195	0.7599	0.6265	0.6868

TABLE 7: Comparaison entre les performances de notre Solution avec les modèle PV-DM et BoW entrainés et testés sur le corpus Enron

8.3 Discussion

Les résultats expérimentaux indiquent que le filtre proposé offre une amélioration significative en termes de tous les métriques de performances utilisés. Notre approche a surpassé les deux modèles PV-DM et BoW dans les deux bases de données Enron et Ling spam en dépit des différences de style du langage et de cohésion des messages entre les courriers électroniques des deux bases. En particulier les emails du corpus Enron contiennent beaucoup d'erreurs grammaticales et les textes des messages ne suivent généralement pas les conventions du langage formel, tandis que les emails de la base Ling spam sont généralement grammaticalement corrects et respectent les conventions de la langue anglaise. Et c'est à cause de cette hétérogénéité entre les deux bases, que les performances de BoW ont considérablement diminué sur l'ensemble de données d'Enron, et également on a remarqué qu'il y'a une disparité notoire de performances du Modèle PV-DM sur le même corpus , ce qui nous permet d'affirmer que notre modèle était le plus résistant à la dégradation de la langue et que la combinaison proposé de l'algorithme TF-IDF et le modèle PV-DM a augmenté le pouvoir discriminant de l'analyse des spams.

9 Conclusion

Dans ce chapitre, nous proposons une nouvelle approche pour le filtrage du spam. Ce filtre met l'accent sur la nature complémentaire des informations fournies par le contexte global et locale des termes les plus pertinents d'un courrier électronique. Notre filtre utilise le modèle de réseau de neurones PV-DM et le schéma TF-IDF pour avoir une représentation vectorielle pour chaque message. La classification finale est faite en entrainant des algorithmes Machine learning avec les représentations vectorielles obtenus.
Les résultats expérimentaux confirment clairement que les classificateurs entrainés avec notre méthode obtiennent les meilleurs résultats et surpassent les filtres basés PV-DM uniquement et BoW.

De plus, ils prouvent que le filtre proposée est plus résistant aux différences des système linguistique et de cohésion des messages. Les recherches futures étendront nos expériences à d'autres jeux de données,comme Spambase ,PU data set et ADCG SS14 Challenge 02. Nous allons aussi travaillé sur les phishing emails qui eux aussi sont des spams, mais avec des liens malveillants .Enfin, l'élaboration d'un filtre pour tous les types de spam constituera un autre défi.

Bibliographie

[1] Radicati.com,https ://site-stats.org/radicati.com/.

[2] Monthly Threat Report Symantec , https ://www.symantec.com/security-center/publications/monthlythreatreport.

[3] Ying, Kuo-Ching Lin, Shih-Wei Lee, Zne-Jung Lin, Yen-Tim,An ensemble approach applied to classify spam e-mails,Expert Syst. Appl. 37. 2197-2201.(2010)

[4] T. S. Guzella et W. M. Caminhas, A review of machine learning approaches to Spam filtering , Expert Systems with Applications, vol. 36, no 7, p. 10206-10222, (sept. 2009).

[5] Q. Le et T. Mikolov, Distributed Representations of Sentences and Documents ,Proceedings of the 31 st International Conference on Machine Learning, Beijing,(2014)

[6] Ronan Collobert,Jason Weston,A Unifed Architecture for Natural Language Processing : Deep Neural Networks with Multitask Learning,Proceedings of the 25th international conference on Machine learning,Pages 160-167 ,(July,2008).

[7] Tomas Mikolov,Kai Chen,Greg Corrado,„Jeffrey Dean,Efficient Estimation of Word Representations in Vector Space, arXiv :1301.3781v3 [cs.CL](7 Sep 2013) .

[8] Yoshua Bengio,Réjean Ducharme,Pascal Vincent,Christian Jauvin,A Neural Probabilistic Language Model,Journal of Machine Learning Research 3 1137–1155,(2003)

[9] Crawford, Elisabeth Koprinska, Irena Patrick, Jon, Phrases and Feature Selection in E-Mail Classification, Proceedings of the Ninth Australasian Document Computing Symposium 59-62,(December 2004).

[10] Matthew Chang, Chung Keung Poon,Using phrases as features in email classification,Journal of Systems and Software, Volume 82, Issue 6,Pages 1036-1045,(2009).

[11] M. Woitaszek, M. Shaaban and R. Czernikowski, Identifying junk electronic mail in Microsoft outlook with a support vector machine, Symposium on Applications and the Internet Proceedings, pp. 166-169.(2003)

[12] Johan Hovold,Naive Bayes Spam Filtering Using Word Position-based attributes and length-sensitive classification thresholds,booktitle=NODALIDA,p. 8,(2005).

[13] IOANNIS KANARIS, KONSTANTINOS KANARIS, IOANNIS HOUVARDAS,EFSTATHIOS STAMATATOS,WORDS VS. CHARACTER N-GRAMS FOR ANTI-SPAM FILTERING,International Journal on Artificial Intelligence Tools ,pp 1-20,(2006) .

[14] Mehran Sahami,Susan Dumais, David Heckerman, Eric Horvitz,A Bayesian Approach to Filtering Junk E-Mail,AAAI Workshop on Learning for Text Categorization,(July 1998).

[15] Alberto Paccanaro, Geoffrey Hinton,Learning Distributed Representations of Concepts using Linear Relational Embedding,IEEE Transactions on Knowledge and Data Engineering,Pages 232 - 244 ·(April 2001)

[16] Jacob Devlin, Rabih Zbib, Zhongqiang Huang,Thomas Lamar, Richard Schwartz,John Makhoul,Fast and Robust Neural Network Joint Models for Statistical Machine Translation,Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 1370–1380,(2014).

[17] Samy Bengio, Georg Heigold,Word Embeddings for Speech Recognition, Proceedings of the 15th Conference of the International Speech Communication Association, Interspeech (2014).

[18] Chao Li, Lei Ji, Jun Yan,Acronym Disambiguation Using Word Embedding, Twenty-Ninth AAAI Conference on Artificial Intelligence,p. 2,(2015).

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean,Distributed Representations of Words and Phrases and their Compositionality , arXiv :1310.4546v1 [cs.CL] Page 9.(16 Oct 2013).

[20] gensim : models.doc2vec – Doc2vec paragraph embeddings ,

[21] gensim-PyPI,https ://pypi.org/project/gensim/.

[22] sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 0.20.0 documentation.

[23] Data & Tasks - CDMC 2018 - Competition ,http ://www.csmining.org /cdmc2018/index.php ?id=5.

Chapitre 3

Extension du Filtre Anti-Spam Pour la Filtration Des Phishing

1 Motivation

Le phishing (ou encore hameçonnage en français) est une technique dite de "social engineering" ayant pour but de dérober à des individus leurs identifiants de connexion et les mots de passe ou leurs numéros de cartes bancaires, en se faisant passer pour une entité de confiance dans une communication électronique [1]. Les escrocs peuvent recueillir illégalement des renseignements en ligne ou tromper les victimes en leur envoyant des courriels semblant provenir d'organisations authentiques, comme des banques. La victime qui cliquerait sur le lien lancerait le téléchargement permettant ainsi l'infection de l'ordinateur et le vol des informations personnelles.
Le phishing ou le vol de données numériques est un problème croissant sur Internet au cours des dernières années touchant les entreprises et les individus. Le groupe d'anti-phishing (APWG) [4] [5] [6]. indique que le nombre total d'attaques par hameçonnage en 2016 était de 1 220 523, soit une augmentation de 65% par rapport à 2015.
Des attaques illustrant à quel point les attaques de phishing sont menaçantes : celle portée à Google et l'intrusion dans les systèmes de RSA Security [6], où les pirates ont pu accéder au serveur central de Google , et récupérer les clés virtuelles des clients de RSA via des phishing Emails.

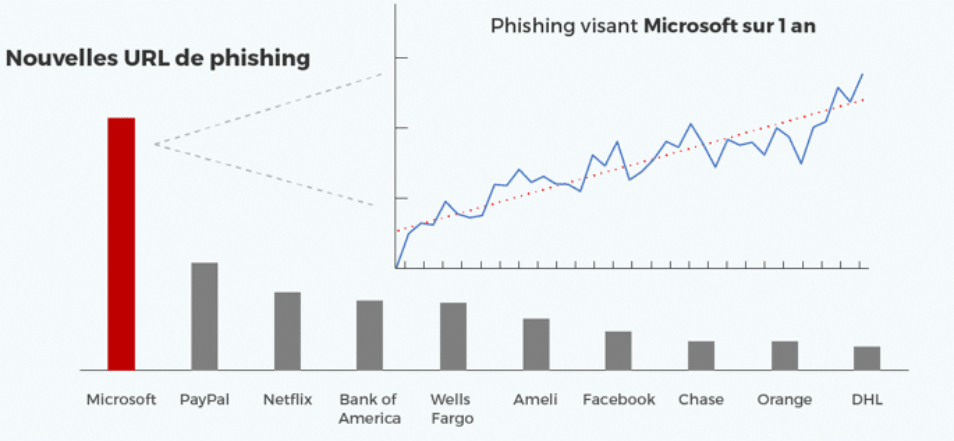

FIGURE 1: Les marques les plus visées par les attaques de phishing en 2018

La Figure 1 montre les marques les plus visées par les attaques de phishing ,et on peut remarquer que Microsoft reste de loin la victime préférée des phishers, Le principal raison de ces attaques est de récupérer des informations d'identification Office 365. Une seule combinaison

identifiant/mot de passe peut en effet permettre aux hackers d'accéder à une quantité phénoménale de fichiers confidentiels, de données et de contacts stockés dans les applications d'Office 365 comme SharePoint, OneDrive, Skype, Excel, CRM, etc. [8] Une attaque d'hameçonnage classique par courrier électronique pouvait typiquement se décrire comme suit :

- Étape 1 : L'hameçonneur envoie à sa victime potentielle un message électronique qui semble en apparence provenir de la banque de cette personne ou d'une autre organisation susceptible de détenir des informations personnelles. Dans cette tromperie, l'hameçonneur reproduit avec soin les couleurs, le graphisme, les logos et le langage de la banque par exemple.[7]

- Étape 2 : La victime potentielle lit le message électronique et mord à l'hameçon en donnant à l'hameçonneur des informations personnelles soit en répondant au message électronique soit en cliquant sur un lien et en fournissant l'information au moyen d'un formulaire qui a l'apparence de celui de la banque ou de l'organisation en question.[8].

- Étape 3 : Par le biais de ce faux site Internet ou du courrier électronique, les informations personnelles de la victime sont directement transmises à l'escroc.[9]

Dans le contexte de phishing , la taille des données a augmenté considérablement, tant en nombre d'instances qu'en nombre de features considérés, ce qui augmente le temps requis pour effectuer la classification.
En outre, les performances d'un classificateur dépendent fortement de la qualité de représentation des données à traiter, souvent une partie de celles-ci ne contient que des informations non pertinentes, redondantes ou inutiles à la tâche de classification. Dès lors il est nécessaire de limiter le nombre de features pris en compte de manière à en extraire l'information discriminante et pertinente améliorant la qualité du classification.
Dans nos travaux précédents [3] , nous avons proposé un filtre anti-spam qui tente de remédier à la faiblesse de l'approche Bag of Word (BoW), en prenant en compte le contexte et la relation entre les mots. Nous y sommes parvenus en proposant une méthodologie basé sur le Deep Learning modèle $PV-DM$, et l'algorithme de sélection de features $TF-IDF$. Les résultats expérimentaux ont prouvé que cette méthode est effective , et plus résistante au changement de style de la langue. Cependant,ce filtre ne peut pas détecter un courrier électronique avec un contenu légitime et un URL malveillant, ce qui nous motive à étendre le filtre anti spam pour la detection des phishing.

2 Les Types de Phishing selon le Contenu textuel de L'Email

- **Tirage Au Sort** : cette technique de phishing est très fréquente sur Internet, les faux tirages au sort incitent l'internaute à entrer sur des pages criminelles qui indiquent que l'on a gagné quelque chose généralement au tirage au sort.Plus particulièrement, la bannière du millionième visiteur est l'une des formes les plus courantes de ces formes d'arnaques.

- **Le Changement De Site** : présent plus particulièrement sur les sites commerciaux, le phishing bancaire de ce type incite l'internaute à entrer ses coordonnées bancaires ou ses contacts pour des raisons de changement d'architecture informatique (mises à jour de données, renouvellement des coordonnées etc).

- **Le Harponnage Sentimentale** : cette forme de phishing met ici la victime devant un cas de conscience. Le cybercriminel propose ainsi à l'internaute d'entrer ses coordonnées bancaires pour effectuer un don. En plus d'avoir transmis ses informations bancaires, la victime aura fait un don sur un formulaire fallacieux. D'autres techniques incitent à la transmission des coordonnées en promettant par exemple l'héritage d'une personne malade d'un syndrome incurable.

❄ **La menace** : C'est une Autre forme de phishing bancaire, certains harponneurs prennent l'apparence d'une enseigne officielle pour prévenir d'un risque de sécurité : Tentative d'intrusion, piratage de données etc.pour faire paniquer la victime et la pousser à remplir effectivement le formulaire fourni généralement en pièce jointe pour y confirmer son identité.

3 Les Techniques Anti Phishing

De nombreux travaux ont été menés pour mettre au point diverses techniques sur la lutte contre le phishing. Certaines techniques fonctionnent sur les Emails, certaines fonctionnent sur les features des sites Web et certaines sur les URL des sites Web.
En général, les techniques anti-phishing peuvent être classées dans les trois catégories suivantes :

❄ **Filtrage du contenu** : Dans cette méthodologie, le contenu / des Emails sont filtrés au fur et à mesure de leurs insertion dans la boîte aux lettres de la victime, à l'aide des algorithmes de Machine Learning, telles que Bayesien Additive Regression Trees (BART) ou Support Vector Machines (SVM) etc.

❄ **Les listes noires** : une Liste noire est une collection de sites Web ou adresses de phishing connus publiés par des entités de confiance telles que la liste noire de Google et de Microsoft. Il nécessite à la fois un composant client et un composant serveur. Le composant client est implémenté en tant que plug-in de messagerie ou de navigateur qui interagit avec un composant de serveur, qui dans ce cas un site Web public fournissant une liste des sites de phishing connues.

❄ **Prévention basée sur les symptômes** : La prévention basée sur les symptômes analyse le contenu de chaque page Web consultée par l'utilisateur et génère des alertes de phishing en fonction du type et du nombre de symptômes détectés.

4 Travaux Connexes

La liste noire est la technique anti-phishing la plus utilisée par les navigateurs Web modernes. Cependant, des études montrent que cette liste n'est pas suffisante pour protéger les utilisateurs contre les nouvelles pages Web qui apparaissent chaque jour par milliers et disparaissent rapidement [11], les listes noires posent deux problèmes majeurs :

- D'une part, l'adresse IP de l'hameçonner et l'URL malicieuse ont tendance à changer constamment pour éviter le suivi de l'expéditeur ou son identification.

- Deuxièmement, les listes noires ne parviennent pas à identifier l'URL de phishing dans les premières heures d'une attaque de phishing, car leur processus de mise à jour est insuffisamment rapide [12].

De nombreuses travaux ont étudié plusieurs features identifiant les URLs comme par exemple : le nombre de domaines dans l'URL [13] [14], l'adresse IP [13] [15] [16] [17], l'utilisation des formulaires avec le bouton "Soumettre" [5] [17] [18], le nombre de points [5] [18] [19] ,si le lien hypertexte est avec une image au lieu de texte visible, ou une image URL basé sur une adresse IP etc. [5] [14] [17].
Maher Arborous et al. [20] ont réalisé un sondage pour identifier les features requises, ce qui contribue à améliorer la précision et l'accurracy de la détection des URL malveillants. S. Shivaji et al [9] prouvent que le nombre des features utilisés dans le moteur de détection a un impact direct sur le temps de traitement. El-Khatib,K [10] montre qu'un ensemble exhaustif de features peut devenir le point de contrôle du système de messagerie.
Fette et ses collègues [5] ont utilisé une technique qui implique la machine learning avec 10 features, un outil anti-spam et le WHOIS service. Une telle approche peut augmenter considérablement le temps nécessaire pour évaluer chaque email,tout simplement parce que Le service WHOIS

n'est efficace que si le domaine est récent. Mais en général, le hameçonner se cache sous des domaines consolidés pour éviter la détection par ce type de requête.
Cook et al [21] proposent d'utiliser un classificateur avec 11 features, bien que les résultats indiquent un bon taux de détection, mais certaines features nécessitent une clarification quant à leurs inclusion dans le processus de classification.
Zhang et al. [22] présentent une approche (qu'ils ont nommée CANTINA) basée sur le contenu de l'email pour détecter les sites Web de phishing, elle emploie l'algorithme TF-IDF et l'algorithme Robust Hyperlink pour avoir 8 features (4 liées au contenu, 3 lexicales et 1 liée au WHOIS service), les résultats montrent que *CANTINA* peut détecter correctement environ 95 % des sitesy de phishing. Cependant, le téléchargement des pages Web réelles augmente le risque potentiel d'analyser le contenu malveillant sur le système de l'utilisateur.

On en déduit que les filtres proposés dans ces travaux permettent de détecter des URLs phishing avec des taux de précision satisfaisants. Cependant,la principale limite de ces approches est que la réduction des features n'est pas validée, ce qui peut entraîner des coûts de calcul inutiles pour la détection de phishing. De plus, il n'y en a pas d'évaluation des corrélations entre les features ; une telle évaluation permettra de mettre en valeur l'importance de chaque feature ,et par suite nous pouvons affecter à chacun un poids qui reflète sa pertinence, pour obtenir le minimum de features distinctes avec une fiabilité similaire à toutes les features ensemble .

5 Background

5.1 Auto Encoder

Les Auto Encoders(AE) font partie de la famille des réseaux de neurones. C'est un réseau non récurrent qui se propage vers l'avant, très semblable au perceptron multicouches - ayant une couche d'entrée, une couche de sortie ainsi qu'une ou plusieurs couches cachées les reliant -, mais avec toutefois une couche de sortie possédant le même nombre de nœuds que la couche d'entrée.
l'objectif de AE est de reconstruire les entrées X (plutôt que de prédire une valeur cible Y étant donné les entrées X). Par conséquent, un Auto Encodeur est un modèle d'apprentissage non supervisé.
Un Auto Encoder se compose toujours de deux parties, l'Encodeur et le Décodeur, qui peuvent être définies comme des transitions f et g, telles que :

$$f : X \longrightarrow H \qquad g : H \longrightarrow \hat{X} \tag{1}$$

$$f, g = argmin_{f,g} ||X - \hat{X}|| \tag{2}$$

où :

- X = $\{x_1, x_2 .... x_n\}$ est l'ensemble des entrées.
- H =$\{h_1, h_2, ... h_d\}$ est la nouvelle représentation de X (représentation compressée).
- $\hat{X}$ = $\{\hat{x_1}, \hat{x_2} ... \hat{x_n}\}$ est l'ensemble de sortie. Voir figure 1

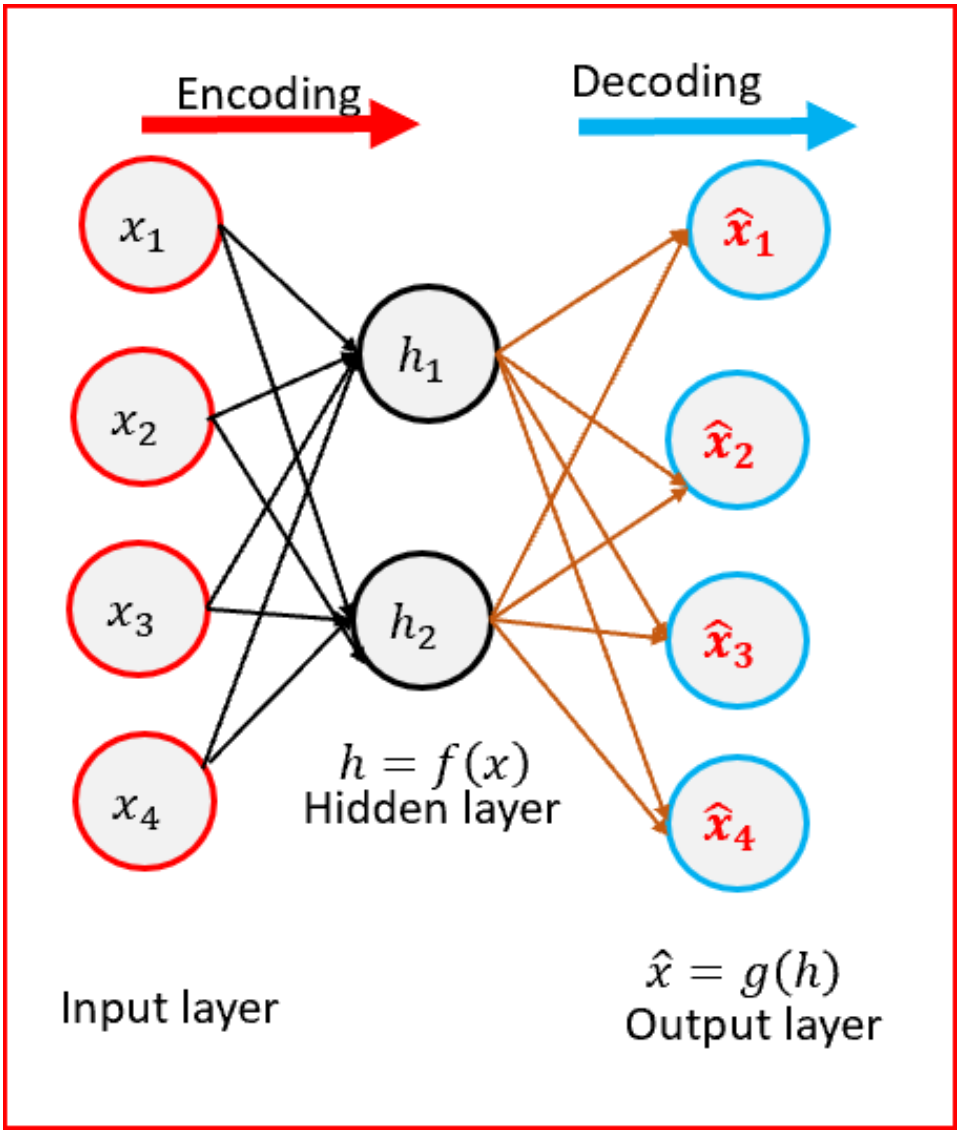

FIGURE 2: Architecture de Deep Auto Encoder

L'Auto Encoder sera entrainé de telle manière à ce que $\hat{X}$ sera proche de X . Dans le cas où il n'y a qu'une seule couche cachée, l'étape d'encodage prend l'entrée :
$X \in \mathbb{R}^n$ et l'associe à $H \in \mathbb{R}^d$ telle que :

$$H = f(W.X + b) \tag{3}$$

- f est une fonction d'activation, e.g., sigmoïde, ReLU.
- **W** une matrice de poids et **b** un vecteur de biais.

Le décodage associe **H** à la reconstruction de $\hat{\mathbf{X}}$ de forme identique à **X** :

$$\hat{X} = g(W'.H + b') \tag{4}$$

où **g** , $\mathbf{W}'$, et $\mathbf{b}'$ du décodeur peuvent différer ou non des f ,**W** ,et **b** de l'Encodeur, selon la conception de l'Auto Encoder. Si la couche H possède une dimension inférieure à celui de la couche d'entrée X , alors le vecteur caractéristique $f(X) = H$ peut être considéré comme une représentation compressée de X .
Si la couches cachée possède une taille plus grande que celle de la couche d'entrée, le réseau

risque d'apprendre la "fonction d'identité", également appelée *fonction nulle*, ce qui signifie que la sortie est égale à l'entrée, ce qui est inutile.

Les AE sont actuellement utilisés pour la compression d'image ou du son et la réduction de la dimensionnalité. Dans des cas spécifiques, elles peuvent fournir des projections de données plus intéressantes et efficaces que PCA (Principal Component Analysis) ou d'autres techniques de réduction de dimensionnalité [25].

5.2 Denoising Auto Encoder

Le Denoising Auto Encoder(DAE) est une version stochastique d'Auto Encoder. L'idée derrière le Denoising Auto Encoder est simple : Afin de forcer la couche cachée du AE à découvrir des features plus robustes et à l'empêcher de simplement apprendre l'identité, nous entrainons l'Auto Encoder à la reconstruction de l'entrée d'une version corrompue de celle-ci.
Le processus de corruption stochastique ramène au hasard certaines entrées (jusqu'à la moitié d'entre eux) à zéro. Par conséquent, le DAE essaie de prédire les valeurs corrompues (c'est-à-dire manquantes) à partir des valeurs non corrompues.
Pour que le DAE réduit le bruit , il effectue trois opérations :

- Essayer de coder l'entrée (conserver les informations relatives à l'entrée).
- capturer les dépendances statistiques entre les entrées.
- essayer d'annuler l'effet d'un processus de corruption appliqué de manière stochastique à l'entrée de DAE.

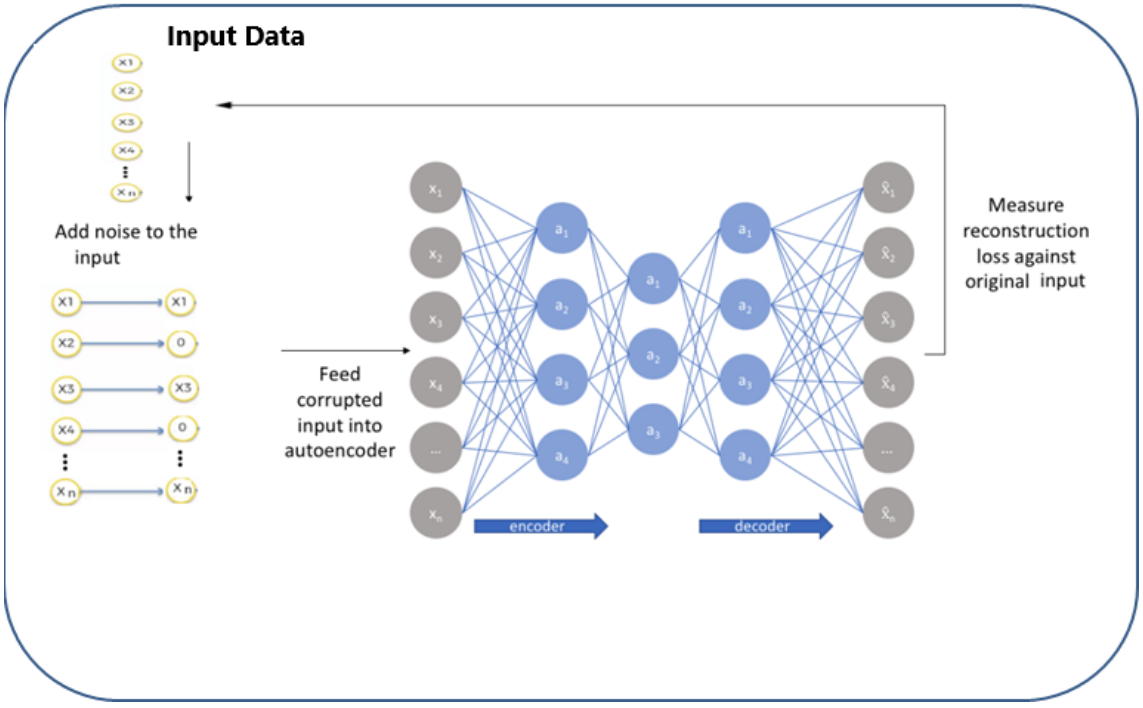

FIGURE 3: Processus de fonctionnement de Denoising Auto Encoder

6 Solution Proposé

Notre système de détection de phishing proposé fonctionne selon l'architecture décrite à la figure 3.

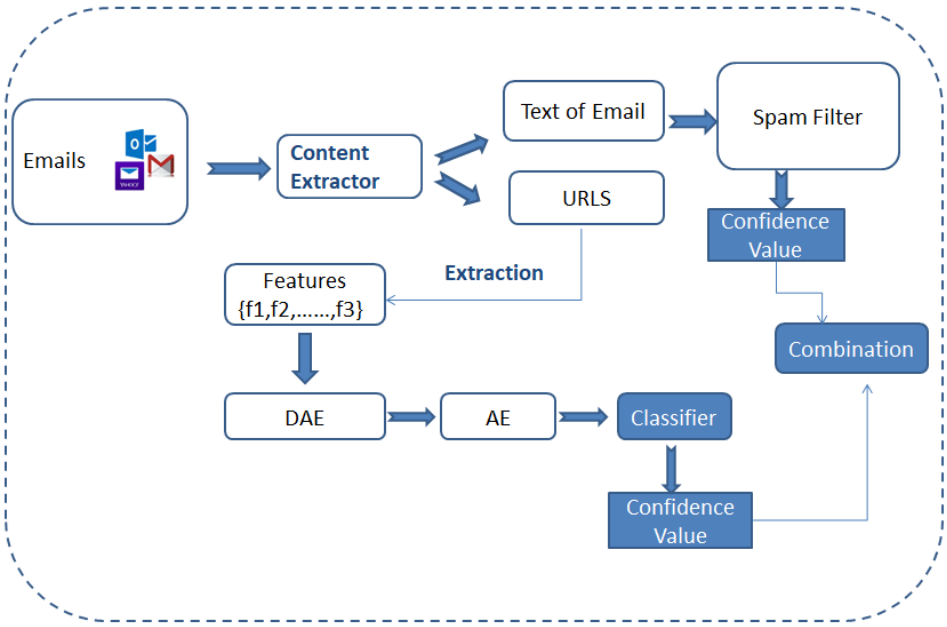

FIGURE 4: Système proposé pour la détection de phishing

L'architecture proposée comprend six modules qui fonctionnent comme un ensemble de tâches. Les fonctions de chaque module sont décrites comme suit :

- Emails d'entrée : ce module est responsable de la réception du contenu du courrier électronique en tant qu'entrée brute.Ce qui signifie que les Emails de ce module contiennent chaque partie de l'Email, telle que l'entête, le texte et l'URL, etc.

- Extracteur de contenu : ce module extrait le texte avec l'URL de l'email. La raison pour extraire uniquement ces deux parties est qu'elles décrivent souvent les caractéristiques du phishing.

- Le Spam Filtre : Les Emails phishing contiennent généralement un texte que les destinataires en réagissent immédiatement en cliquant sur le lien fournis dans l'Email ou envoient les informations cruciales en réponse. Cependant, les recherches dans ce domaine portent généralement sur la classification des phishing en utilisant uniquement les informations disponibles dans les URLs [11]. Par conséquent, pour analyser le contenu textuel de l'Email, nous proposons comme module dans notre architecture de détection, le filtre anti spam proposé dans le chapitre précédent [3], qui analyse le contexte de l'Email et ses pertinentes features,et renvoie une valeur de confiance.

- construction de features : une fois le jeu d'URLs extrait des Emails est disponible, le moteur de construction d'entités crée divers ensembles de features conçus en fonction de l'expérience et utilisés dans différents types d'Emails de phishing.

- Denoising Auto Encoder : très souvent les hameçonneurs créent de nouvelles techniques en changeant des features, pour perturber le système de détection, créant ainsi un point d'incertitude pour le classificateur. L'un de nos objectifs est de créer un filtre suffisamment robuste pour faire face aux partielles modifications des features. Pour cela,nous proposons d'introduire un DAE (Denoising Auto Encoder) dans notre système. comme nous l'avons mentionné ci dessus, Le DAE est une variante plus récente de l'Auto Encoder conçu pour reconstruire les entrée «réparées» propre à partir d'une version corrompues [26]. Nous entrainons le Denoising autoencoder sur des copies corrompues des données . Pour la corruption, nous utilisons une simple fonction ϕ qui pour chaque entrée x, un nombre fixe C de

features sont choisis au hasard, et leurs valeur est forcée à 0, tandis que les autres sont laissés intacts [27], notons que nous pouvons considérer une fonction de corruption alternative. Pour restaurer les données initiales, Le Denoising autoencoder sera entrainé comme suit : Étant donné $X = \{x_1, x_2, ...x_n\}$ ensemble de données

- nous construisons les entrées corrompus :

$$\hat{X} = \phi(X) \tag{5}$$

- les entrée corrompues sont mappés sur la couche cachée :

$$H = \sigma(W.\hat{X} + b) \tag{6}$$

- À partir H on va reconstruire Y :

$$Y = \sigma'(W'.H + b') \tag{7}$$

Où W, W' , sont les matrices de pondération b, b' les vecteurs de biais de DAE,et σ ,σ' les fonctions d'activation.
Le DAE est entraîné à reconstruire une entrée réparée Y à partir de la version corrompue $\hat{X}$, en cherchant les paramètres W, W ', b et b' qui minimisent l'erreur de reconstruction. Cela correspond à minimiser la fonction objectif suivante :

$$\sum_{i}^{n} ||x_i - \tilde{x}_i|| \tag{8}$$

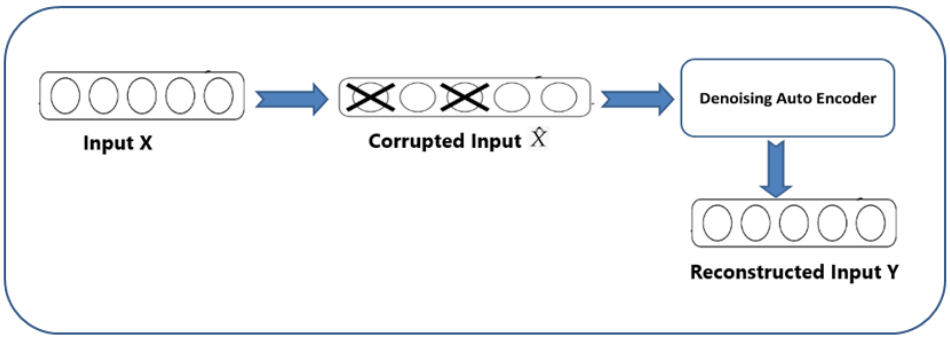

FIGURE 5: La reconstruction des données corrompus par un DAE

❄ Auto Encoder : Les données du phishing URL contiennent un grand nombre de features allant jusqu'au des dizaines de milliers de features [11] [9] ,ce qui augmentent considérablement l'erreur de classification. Par conséquent, pour accélérer les algorithmes de classification, un Auto Encoder est utilisé pour réduire la dimension des features, par un algorithme non supervisé permettant de sélectionner les features pertinentes, qui préservent la structure des données d'origine.

Étant donné un ensemble de features extraits des URLs $X = \{x_1, x_2, ...x_n\}$, l'entrainement de l'Auto Encoder consiste à trouver les paramètres W, b, W', b' afin de forcer la sortie $\hat{\mathbf{X}}$ à être aussi proche que possible de l'entrée $\mathbf{X}$,en minimisant l'erreur de reconstruction et utilisant la distance euclidienne comme fonction de perte standard d'Auto Encoder, cela correspond à minimiser la fonction objectif suivante :

$$\sum_{i}^{n} ||x_i - \hat{x}_i|| \tag{9}$$

Une des caractéristiques principales de l'Auto Encoder est qu'il peut être entraîné avec des données non labellisés. Ainsi, une fois la représentation cachée(réduite) H est apprise, elle peut être utilisée comme entrée d'un classificateur supervisé , entraîné avec un petit ensemble de données labellisés , ou il est également possible d'entrainer d'autres Auto Encoder avec cette représentation cachée.

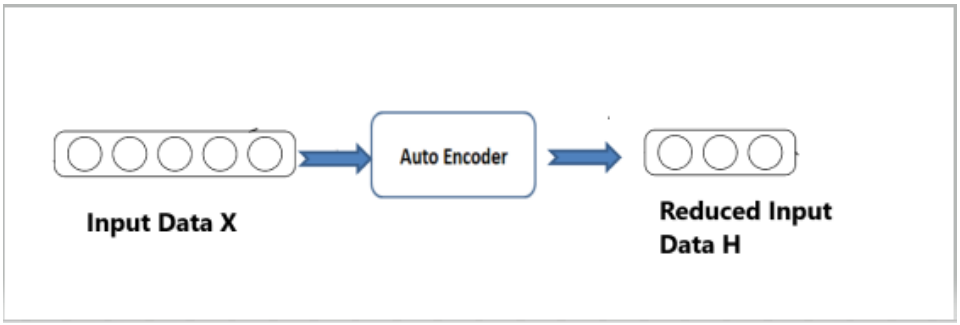

FIGURE 6: La Réduction des données par AE

- Entrainement du Classifier :Les features réduits appris par DAE constitueront l'entrée d'un classificateur tel que la Régression Linéaire(RL) ou Random Forest. Étant donnée un ensemble d'entrée $X = \{x_1, ..., x_n\}$ et l'ensemble des classes c = {0, 1} (0 pour le courrier électronique normal , 1 pour le courrier électronique phishing), les classificateurs Régression Logistique et Random Forest et SVM estimeront la probabilité $p(c = j/x_i)$ pour chaque i= 0,1,...n et j= 0,1 .

- Système de combinaison : le système de combinaison est basé sur les valeurs de confiance retournées par chaque filtre (voir Figure 4).
  Puisque les valeurs de confiance sont comprises entre zéro et un, nous calculons la moyenne des deux valeurs et le courrier électronique est étiqueté comme Normal si la moyenne est comprise entre 0 et 0.5 et phishing sinon.

7 Design Expérimental

7.1 Data Set

L'un des défis de notre recherche a été l'absence de bases de données fiables pour les messages phishing contenant le texte et le lien correspondant.On a trouvé que des bases des phishing contenant justes les liens URLs ou leurs features sans les messages textes correspondants.
Bien que de nombreux articles sur l'hameçonnage aient été publiés ces derniers temps, aucun ensemble de données d'entrainement fiable n'a été publié. Ceci est expliqué par le fait qu'il n'y a pas un accord dans la littérature sur les features définitives qui caractérisent les pages Web et les liens d'hameçonnage.
Enfin, un autre défi est qu'on n'a pas trouvé un ensemble des URLs d'entraînement assez suffisante pour que notre modèle puisse apprendre, découvrir , et extraire des features utiles,vu que le Deep learning a besoin de données massives pour s'ajuster et apprendre la distribution des données.
Étant donné ces contraintes, nous nous sommes contenté de la base de données de Phishing_Legitimate_full Data set [28]. Cette base de données contient 48 attributs extraites de 5 000 pages Web de phishing basées sur des URL extraites de PhishTank2 et OpenPhish3 et 5000 autres pages Web légitimes basé sur les URL de Alexa4 et l'archive Common Crawl5 [29].
Nous avons utilisé 80% de l'ensemble de données(i.e 8000 URls) pour l'entrainement du modèle, tandis que le modèle est testé en utilisant 20% du Data set. Le Tableau 1 montre quelques features du Data set utilisée dans les expériences .

caractéristique	Description
NumDots	nombre de point dans le lien
UrlLength	la longueur du lien
AtSymbol	nombre de @ dans le lien
TildeSymbol	nombre de ~dans le lien
NumUnderscore	nombre de _ dans le lien
NumPercent	nombre de % dans le lien
NumHash	nombre de Hachage
NumNumericChars	nombre de caractères numériques dans le lien
RandomString	nombre de caractères aléatoires dans le lien
IpAddress	Ip adress
PathLength	longueur du path
DoubleSlashInPath	nombre de // dans le lien
EmbeddedBrandName	existance d'un Nom de marque incorporé
PopUpWindow	existance de pop windows

TABLE 1: Exemple des features du Data set avec leurs description

7.2 La Normalisation des caractéristiques

Étant donné que la plage de valeurs des données brutes varie considérablement et que les fonctions objectives des algorithmes d'apprentissage automatique ne fonctionnent pas correctement sans normalisation. Par exemple, la majorité des classificateurs calculent la distance entre deux points en fonction de la distance euclidienne. Si l'un des features possède une large plage de valeurs, la distance sera régie par ce feature particulier. Par conséquent, la plage de toutes les features devrait être normalisée de sorte que chaque feature contribue approximativement proportionnellement à la distance finale.
Par suite , nous avons normalisé tous les features numériques de la base de données en appliquant la méthode *Standard Scaler* de *scikit-learn* [30] qui applique la formule suivante sur toutes les valeurs des features :

$$z_i = \frac{x_i - mean(X)}{stdev(X)}, i = 1....n \tag{10}$$

où X est l'ensemble de données d'apprentissage $X = \{x_1, x_2, ..., x_n\}$.$x_i$ est un vecteur de feature et z_i est le i ème feature normalisées. Après la normalisation , toutes les valeurs de features sont compris entre 0 et 1.

7.3 Nombres des couches cachées de l'Auto Encoder

Dans l'expérience, nous étudions l'impact du nombre de couches cachées sur les performances de L'Auto Encoder, qui est entrainé suivant l'algorithme suivant :

Algorithme d'entrainement de l'AE

- **Input** : Dataset $X = \{x_1, x_2, ...x_n\}$ et nombre de couches cachés L.
- pour $l \in [1, L]$: **initialiser** $W_l = 0, b_l = 0, W_l' = 0, b_l' = 0$.
- **définir** le vecteur de représentation de la couche cachée h_l :
  $h_l = \sigma(W_l.h_{l-1} + b_l)$
- **définir** la l ième sortie de couche cachée :
  $y_l = \sigma'(W_l'.h_l + b_l')$

- **tant que** le critère d'arrêt(seuil d'erreur accepté) n'est pas atteint faire :
- **calculer** h_l à partir de h_{l-1}
- **calculer** y_l
- **calculer** loss fonction
- **mettre à jour** les paramètres de la couche W_l, b_l, W_l', b_l'

le tableau 2 montre les différents architectures de l'Auto Encoder que nous avons expérimenté
.

Auto Encoder	nombre de couches cachés	nombre de neurones cachés
Auto Encoder 1	une seule couche cachée	input : 47 encoder :6 decoder :47
Auto Encoder 2	trois couches cachées	input :47 encoder : 24 x 12 x 6 decoder : 12 X 24 X 47

TABLE 2: les Auto Encoders expérimentés avec le nombre de couches et le nombre des neurones cachés

7.4 Resultats des Auto Encoders avec différents nombres de couches cachés

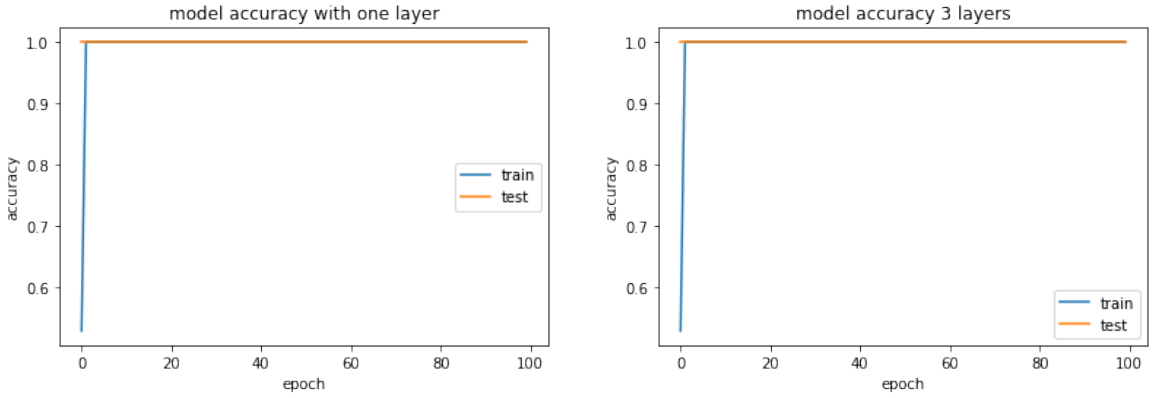

FIGURE 7: taux de la precision realisé parl' Auto Encoder avec nombre different de couches cachés

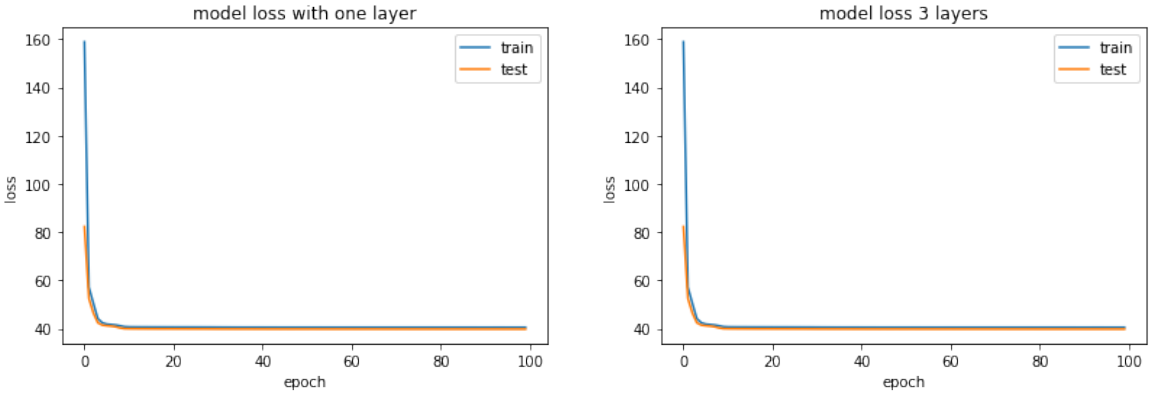

FIGURE 8: loss de l'Auto Encoder avec nombre différent de couches cachés

D'après la figure 7 et la figure 8, nous pouvons constater que le modèle peut être entrainé uniquement avec une seule couche cachée et que la reconstruction de l'erreur semble bien converger dans le cas d'une seule couche masquée mieux que avec plus d'une couche. Après l'entrainement de l'autoencoder, nous gardons que le codeur , ainsi nous pouvons compresser notre Data set de dimension 47 X 10000 jusqu'au 6X 10000 .

7.5 La Fonction De Corruption

Dans cette étude , nous effectuons une corruption des données partielle , par une fonction de bruit $\phi(x) = \tilde{x}$ dont la nature est déterminée au préalable . Il existe différent type de fonction de corruption :

- bruit gaussien : tiré d'une normale centrée en x : $\tilde{x} = x + N(x, \sigma^2)$
- masque : un sous-ensemble des éléments $x_{i,j}$ d'un vecteur x_i, choisi aléatoirement, est forcé à 0.
- bruit poivre et sel : un sous-ensemble des éléments $x_{i,j}$ d'un vecteur x_i , choisi aléatoirement, est forcé à 0 ou à 1.

Le bruit gaussien additif est le modèle de bruit le plus courant et constitue un choix naturel pour les entrées à valeur réelle, nous l'avons adopté pour notre expérience.

7.6 Entrainement de Denoising Autoencoder

Le Denoising Auto Encodeur ressemble beaucoup au modèle de l'Auto Encoder que nous avons vu précédemment. La différence est que,le DAE est entraîné à reconstruire son entrée à partir d'une version corrompue de celle-ci.
Dans les expériences, nous utilisons le DAE pour extraire les features pertinentes de façon à palier les erreurs qui y sont volontairement introduites.
Ainsi pour permettre au DAE de reconstruire l'entrée d'origine à partir de la version corrompue,l'astuce consiste à faire correspondre la version d'origine (et non la version corrompue) à la version reconstruite au moment du calcul d'erreur. On s'attend à ce que la représentation apprise ainsi soit plus robuste à d'éventuelles corruptions des données.
Les figures 9 et 10 montrent les performances du DAE durant la phase de L'entrainement et le test.

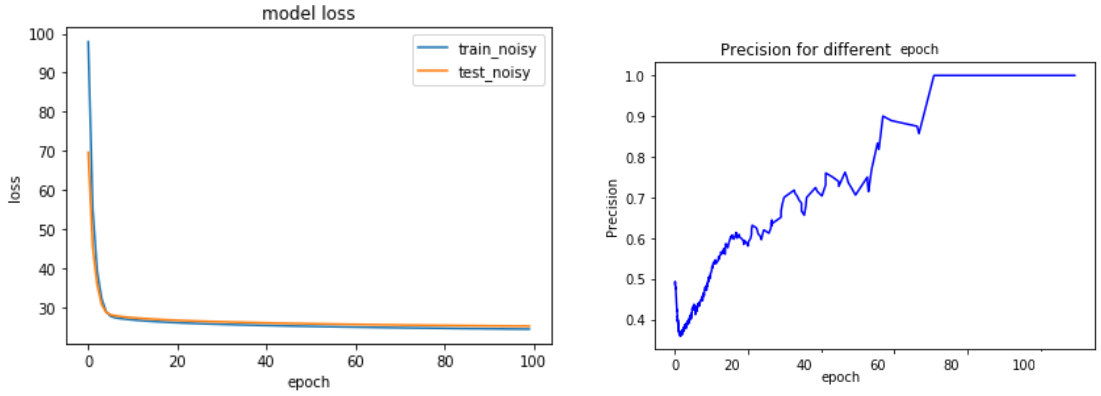

FIGURE 9: loss et precision de de DEA au cours de la construction des données

8 Résultats Expérimentaux

Une fois que le Denoising Auto Encoder et l'Auto Encoder sont entrainés à la reconstruction des données corrompus et à la réduction des features,nous pourrons utiliser les données réduites en tant qu'entrées des classificateur, par exemple Random Forest classifier ou Régression Logistique etc. les résultats de Classifications sont montrés dans le tableaux 3.

Classificateur	label	précision	recall	f1 score	support
Random Forest	Normal	0.62	0.57	0.59	975
	phishing	0.62	0.66	0.64	1025
SVM	Normal	0.59	0.72	0.64	975
	phishing	0.66	0.52	0.58	1025
Logistic Regression	Normal	0.59	0.68	0.63	975
	phishing	0.64	0.56	0.60	1025

TABLE 3: les performances de classification sur les données réduites par l' Auto Encoder

On constate que les performances des classificateurs sont très proches, mais on peut notifier que le classificateur Random Forest est le meilleur,vu son *f1* score à l'identification des intrusion atteint 64%.Mais en générale les résultats du classification sont non satisfaisantes,car la dimension du base de données est très petite et le processus de corruption doit être répété plusieurs fois sur les même entrées.
Si la base d'entraînement n'est corrompu qu'une fois au début de l'entraînement, le modèle reconnaîtra les distorsions en tant que modèles de données valides, ce qui conduit à un sur ajustement. En corrompant les données à chaque fois d'une manière différente, le modèle apprendra à extraire la structure sous-jacente, en évitant les sur ajustements.

9 Conclusion et Perspectives

Dans ce chapitre, nous avons proposé un system de détection de phishing augmentant la sécurité de la messagerie, par un double filtre : l'un pour l'analyse du contenu textuel d'email et le seconde pour l'analyse des URLs suspectes. Pour analyser le contenu textuel d'un' Email, nous avons proposé d'utiliser notre filtre anti spam implémenté en chapitre 1.[3]
Pour analyser les URLs nous avons proposé un nouveau filtre composé d'un Denoising Auto Encoder et un Auto Encoder.
Nous entrainons Le DAE pour la reconstructions des données originales à partir d'une version corrompue,dans l'objectif de créer un filtre robuste et résistant face à tout changement partielle dans les données
nous avons aussi forme un Autoencoder qui recherche une représentation compressée des features

des URLs. Cette méthodologie présente de nombreux avantages pour la classification des courriers de phishing :

- ❄ Le fait qu'elle exploite les informations disponibles sur les URL, ainsi que sur le contenu textuel des courriers électroniques.
- ❄ Elle apprend la structure profonde des données à l'aide d'un autoencoder non supervisé.
- ❄ Elle pourrait reconstruire les entités même si elles sont corrompues.

Les Résultats sont moyennes vue la petite dimension de la base de données qu'on a expérimenté, ce qui a une influence sur le processus d'apprentissage de L'autoencoder et le Denoising autoencoder qui nécessite une grande quantité de données pour l'entrainement.

Nous prévoyons dans les travaux futurs, d'explorer autres algorithmes de Deep learning telle que Reccurent Neural Network afin d'améliorer les performances de Notre Algorithme,et d'adopter la méthode de filtrage proposée comme un IDS pour cyber Security .

Bibliographie

[1] Ramzan, Zulfikar, Phishing Attacks and countermeasures. . Handbook of Information and Communication Security, Springer. ISBN 9783642041174,(2010)

[2] Bengio, Yoshua ,Learning Deep Architectures for AI, Foundations and Trends in Machine Learning : Vol. 2 : No. 1, pp 1-127. ,(2009) .

[3] Samira Douzi, Meryem Amar, Bouabid El ouahidi, Hicham Laanaya,Towards A new Spam Filter Based on PV-DM (Paragraph Vector-Distributed Memory Approach),Science Direct ,Procedia Computer Science Volume 110,Pages 486–491,(2017).

[4] E. El-Alfy, R. Abdel-Aal,. Using GMDH-based networks for improved Spam detection and e-mail feature analysis. Applied Soft Computing 11 (1) 477–488, (2011).

[5] Ian Fette, Norman Sadeh,Anthony Tomasic,Learning to Detect Phishing Emails. International World Wide Web Conference, pp. 649–656,2007.

[6] APWG , http ://www.apwg.org/resources/apwg-reports/. Phishing Activity Trends Report 4 th Quarter,2016.

[7] Ramzan Z., Wüest C,Phishing Attacks : Analyzing Trends in In Fourth conference on Email and Anti- Spam Mountain view : Citeseer, (2007).

[8] https ://www.vadesecure.com/fr/phisher-favorite-classement-des-marques-les-plus-ciblees/

[9] Aaron, G. The state of phishing .Computer Fraud & Security .(6) pp.5–8, (2010).

[10] S. Shivaji, E.J. Whitehead, R. Akella, K. Sunghun, Reducing features to improve bug prediction . IEEE/ACM International Conference on Automated Software Engineering 600–604. (2009).

[11] El-Khatib,K,Impact of feature reduction on the ef[U+FB01]ciency of wireless intrusion detection systems. IEEE Transactions on Parallel and Distributed Systems 21(8) pp1143–1149,(2010).

[12] Ram B. Basnet et al,Learning To Detect Phishing URLs, International Journal of Research in Engineering and Technology ,Volume : 03 Issue : 06,(Jun-2014) .

[13] Sheng.S et al, An empirical analysis of phishing blacklists,In Proceedings of the CEAS'09, (2009).

[14] Basnet, S. Mukkamala, A. Sung,Detection of phishing attacks : a machine learning approach , Soft Computing Applications in Industry 373–383, (2008)

[15] C.K. Olivo et al ,Obtaining the threat model for e-mail phishing. Appl. Soft Comput. J. (2011).

[16] Chen. J,Guo.C,Online detection and prevention of phishing attacks , Communications and Networking in China 19–21, (2006) .

[17] J. Yearwood, M.Mammadov,A.Banerjee, Pro[U+FB01]ling phishing e-mails based on hyperlink information, International Conference on Advances in Social Networks Analysis and Mining 120–127,(2010) .

[18] Biju Issac, Raymond Chiong and Seibu Mary Jacob,Analysis of Phishing Attacks and Countermeasures at www.arxiv.org, 2006.

[19] Detecting Malicious URLs in E-mail- An Implementation,2013 AASRI Conference on Intelligent systems and control, Procedia 4 (2013) 125-131. Dhanalakshmi Ranganayakulu, Chellappan C. .

[20] Santhana Lakshmi V, Vijaya MS,Efficient prediction of phishing websites using supervised learning algorithms . International Conference on Communication Technology and System Design 2011,Procedia Engineering 30 pp 798 – 805, (2012).

[21] Maher Aburrous, Hossain, M.A., KeshavDahal and FadiThabtah. Experimental Case Studies for Investigating EBanking Phishing Techniques and Attack Strategies. Cognitive Computing,Vol. 2, pp. 242-253 . (2010).

[22] D. Cook, V. Gurbani, M. Daniluk, Phishwish, a stateless phishing [U+FB01]lter using minimal rules , Lecture Notes in Computer Science 182– 186,(2008).

[23] Y. Zhang, J. Hong, L. Cranor, CANTINA , a content-based approach to detecting phishing web sites , 16th International Conference World Wide Web, WWW[U+201F]07 Banff, Alberta, Canada, pp. 639-648,(2007) .

[24] Fuzhen Zhuang,Fuzhen Zhuang , Representation Learning via Semi-supervised Autoencoder for Multi-task Learning,IEEE International Conference on Data Mining, (2015).

[25] [24] Ozan úIrsoy, Ethem Alpaydõn, Unsupervised Feature Extraction with Autoencoder Trees,Neurocomputing j.neucom.2017.02.075,(2017).

[26] Shao Haidong, Jiang Hongkai,Zhao Huiwei, Wang Fuan, A novel deep autoencoder feature learning method for rotating machinery fault diagnosis, Mechanical Systems and Signal Processing 95 pp187–204,(2017).

[27] Jun Deng, Zixing Zhang, Florian Eyben and Björn Schuller,Autoencoder-based Unsupervised Domain Adaptation for Speech Emotion Recognition. IEEE SIGNAL PROCESSING LETTERS, VOL. 21, NO. 9, (2014) .

[28] Pascal Vincent et al,Extracting and Composing Robust Features with Denoising Autoencoders, Proceedings of the 25 International Conference ence on Machine Learning, Helsinki, Finland, (2008).

[29] Phishing Dataset for Machine Learning : Feature Evaluation,https ://data.mendeley.com/datasets/h3cgnj8hft/1.

[30] Kang Leng Chiew et al,A New Hybrid Ensemble Feature Selection Framework for Machine Learning-based Phishing Detection System, Information Sciences journal (2019).

[31] https ://scikit-learn.org/stable.

Chapitre 4

Vers Un IDS Learning Via Le Model PV-DM Et Mutual Information

1 Motivation

Avec la croissance de la demande de connectivité universelle, les systèmes de détection d'intrusion (IDS) sont devenus l'une des contre-mesures les plus courantes en matière de sécurité des systèmes informatiques.
un système de détection efficace des intrusions sur les systèmes informatiques nécessite :

- la manipulation d'énormes quantités de données et que certaines ne sont peut-être pas pertinentes.D'où on propose le procédé de sélection des pertinents features.
- Un événement peut caractériser une attaque ou non suivant le contexte particulier dans lequel il est apparu.D'où le contexte d'un événement est important à définir.
- Une attaque peut concerner plusieurs événements,ces dépendances ne peuvent pas être prises en considérant uniquement le label d'un seul événement.

Ainsi,nous proposons une approche de detection des intrusions en Deep Learning , pour pallier à ces contraintes de dessus.
Pour entrainer et tester notre IDS , nous avons utilisé deux Data sets : NSL KDD et UNSW-NB15, nous avons pu réduire les features nécessaires à la detection des intrusions de 48 features dans UNSW-NB15 et 41 features dans NSL KDD à seulement trois features chacune.
Ces résultat sont pertinents en eux même, car la réduction des features était publique.

2 Le Système de Detection d'intrusion : Définition et Types

Le system de détection d'intrusion (IDS) est l'identification d'éléments, d'événements ou d'observations qui ne sont pas conformes à un modèle connu et défini [1] .
Les techniques de détection des intrusions les plus répandues sont généralement deux types : **la misuse detection** et **la détection d'anomalies** .

- La misuse detection est un système qui détecte les intrusions en recherchant les activités qui correspondent à des signatures d'attaques connues et stockées dans des bases de données. L'un des principaux avantages de ces techniques de détection est leurs grande précision à détecter des attaques connues et leurs variations.
  Les inconvénients principaux de cette approche est l'incapacité à identifier et à caractériser des nouvelles attaques.
- L'autre approche de détection d'anomalies construit de comportements normales et détecte tout écart par rapport à ces modèles.

Les algorithmes de détection d'anomalies présentent l'avantage de pouvoir détecter de nouveaux types d'intrusions. Cependant, elles présentent un taux élevé de fausses alarmes,pare ce que Le changement constant dans les schémas d'intrusion peut non seulement introduire de nouveaux types d'attaques, mais peut également modifier les aspects du comportement normal. ainsi des comportements non connus du système sont également classés comme des anomalies et donc signalés comme des intrusion malveillants.

Par conséquent, il existe un besoin réel d'améliorer les systèmes de détection et de les maintenir à jour afin de détecter les changements même les plus subtils dans un système fonctionnant normalement [2].
C'est ainsi que nous proposons une nouvelle approche basée sur le modèle Deep Learning PV-DM et l'algorithme de sélection de features Information Mutuelle.
Dans cette recherche , nous présumons que les paquets réseau sont écrits dans un langage naturel et essayons d'apprendre à l'aide de PV-DM :

- La différence entre le trafic normal et le trafic malveillant automatiquement .[2]
- Les dépendances entre les features et les événements d'un système informatique, et leurs comportements correspondants [3] .
- Capturer la liaison entre les événements,puisque un événement peut présenter ou non une attaque en fonction du contexte dans lequel il apparaît. Et de même, une attaque peut impliquer plusieurs événements. Ainsi, ces dépendances ne peuvent pas être capturées que par les labels des événements [4].

3 BACKGROUND

3.1 Paragraph Vector Distributed Memory (PV-DM)

Le PV-DM est un algorithme de NLP développé par Google [9] [10] [11]. C'est une technique basée sur les réseaux de neurones qui convertit les mots en des vecteurs réels, en tenant compte de l'ordre des mots, de sorte que Les mots similaires sémantiquement et syntaxiquement sont étroitement positionnés dans l'espace. Ces vecteurs peuvent ensuite être utilisés comme features pour des algorithmes de classification et d'apprentissage automatique.
Les modèles PV-DM ont été appliqués très récemment à plusieurs cas d'utilisation Par exemple, dans le domaine médical,en biologie,où elle a été utilisée pour la classification des protéines [5]. Dans le domaine du marketing, il a été utilisé prédire le comportement du consommateur.
Parmi les autres cas d'utilisation,le PV-DM ou le Paragraphe vecteur permet de calculer la similarité sémantique entre deux documents et d'inférer des documents similaires sémantiquement. Certaines implémentations prennent également en charge l'inférence de l'incorporation de documents dans des documents non vus.
Cette fonction est importante pour développer un système de detection des Intrusions , voire elle permet de détecter un trafic malveillant invisible,car le trafic malveillant invisible peut inclure un mot inconnu (par exemple, un nom de domaine complet (FQDN) récemment modifié, des chaînes aléatoires etc).
Dans notre approche ,les paquets réseau sont considérés et lus comme un langage naturel par le modèle PV-DM .En effet,chaque connexion ou événement du réseau comprend des features tels que l'adresse IP de source et de destination, le protocole, le service etc. Notre méthode utilise les de ces champs(features) comme des mots séparés. Figure 1 .
Ensuite, PV-DM construit un espace vectoriel à partir de l'ensemble des événements et convertit les événements en vecteurs avec les labels. Ces vecteurs labellisés sont des données d'apprentissage pour un classificateur.

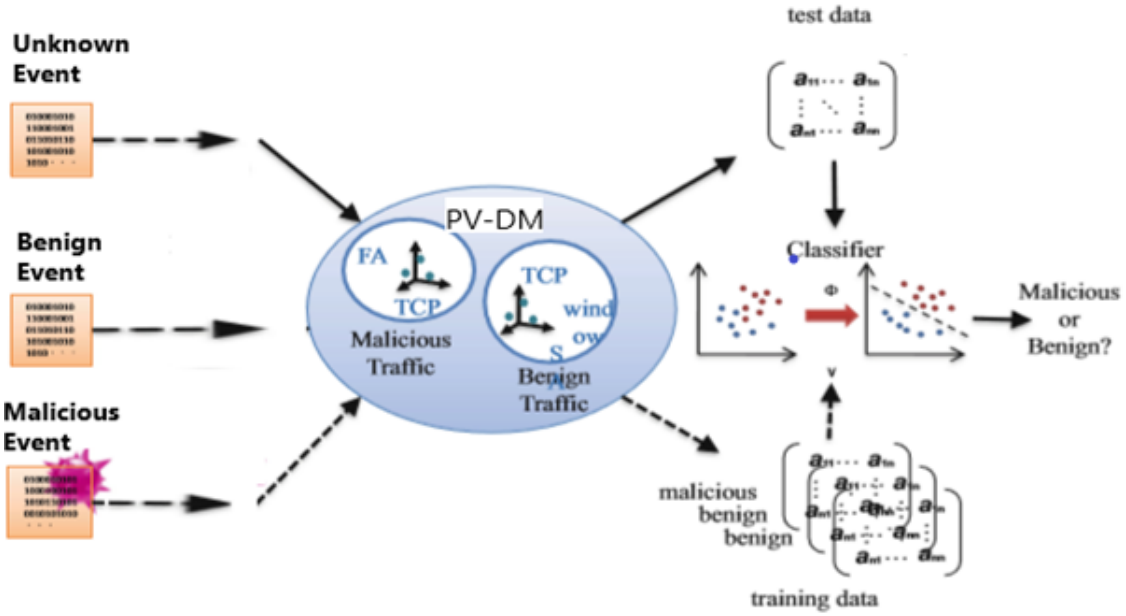

FIGURE 1: Application du modèle PV-DM sur les événements

3.2 Information Mutuelle

3.2.1 Les Algorithmes de Réduction de l'espace des Features

Les données de grande dimension constituent un problème significatif qui s'impose dans les apprentissages supervisés et non supervisé [12]. Ce problème devient de plus en plus important avec l'explosion récente de la taille des Dataset disponibles. La principale motivation pour réduire la dimensionnalité des données et garder le nombre de features assez faible que possible, est la réduction du temps d'entrainement ,les coûts de traitement et l'espace de stockage requis [13] .
Les méthodes de réduction de la dimensionnalité peuvent être divisées en deux groupes principaux : celles basées sur l'extraction de features et celles basées sur la sélection de features.

- ❄ Les méthodes d'extraction de caractéristiques transforment les entités existantes en un nouvel espace de dimension inférieure. Au cours de ce processus, de nouvelles caractéristiques sont créées en fonction de combinaisons linéaires ou non linéaires d'attributs du jeu de données originales. L'analyse en composantes principales (ACP ou PCA), l'analyse discriminante linéaire (LDA) et Autoencoder sont des exemples de tels algorithmes.

- ❄ Pour les techniques de sélection des caractéristiques, elles réduisent la dimensionnalité en sélectionnant un sous-ensemble de fonctionnalités minimisant une certaine fonction de coût. Contrairement aux méthodes d'extraction , les méthodes de la sélection ne modifient pas les données et sont utilisées au stade du pré-traitement des données avant la phase d'entrainement. Ce processus de sélection est également appelé la sélection variable ou la sélection de sous-ensemble variable [14] [15].

3.2.2 Information Mutuelle

L'Information Mutuelle d'un couple (X,Y) de variables représente leur degré de dépendance au sens probabiliste [16].
On dit que deux variables sont indépendantes si la réalisation de l'une n'apporte aucune information sur la réalisation de l'autre. Par suite,L'Information Mutuelle est nulle si et seulement si les variables sont indépendantes, et croit seulement lorsque la dépendance entre les deux variables augmente.

3.2.3 Définition

Étant donné deux variables aléatoires X et Y,Le MI de X et y est défini par :

$$\begin{aligned} I(X;Y) &= H(X) - H(X/Y) \\ &= H(X) + H(Y) - H(X;Y) \end{aligned} \tag{1}$$

Où :

- H (.) Est l'entropie.
- H (X / Y) et H (Y / X) sont des entropies conditionnelles .
- H (X,Y) est l'entropie conjointe de X et Y.

Ces entropies sent définies comme suit :

$$\begin{aligned} H(X) &= -\sum_{x\in X} p_X(x)\log p_X(x) \\ H(Y) &= -\sum_{y\in Y} p_Y(y)\log p_Y(y) \\ H(X;Y) &= -\sum_{y\in Y}\sum_{x\in X} p_{X,Y}(x,y)\log p_{X,Y}(x,y) \end{aligned} \tag{2}$$

Où $p_{X,Y}(x,y)$ est la fonction de densité de probabilité conjointe , $p_X(x)$ et $p_Y(y)$ sont des fonctions de densité marginales de X et Y, respectivement définies comme suit :

$$\begin{aligned} p_X(x) &= \sum_{x\in X} p_{X,Y}(x,y) \\ p_Y(y) &= \sum_{y\in Y} p_{X,Y}(x,y) \end{aligned} \tag{3}$$

En remplaçant les équations 2 et 3 dans l'équation 1, l'équation MI sera :

$$I(X,Y) = \sum_{y\in Y}\sum_{x\in X} p_{X,Y}(x,y)\log\frac{p_{X,Y}(x,y)}{p_x(x)p_y(y)} \tag{4}$$

Les techniques d'informations mutuelles ont été largement utilisées sur des données de grandes dimensions pour mesurer la pertinence entre les features et les classes cibles.
Le but de l'utilisation de cet algorithme dans notre étude est comme suit :

- Étant donné : un ensemble de données $X = \{x_1, x_2, ...x_n\}$.
- L'ensemble des features de X ,$F = \{f_1, f_2, ..f_m\}$ où $x_i = \{f_1^i, f_2^i, ...f_m^i\}$,pour $i = 1, ..., n$.
- L'ensemble des labels de X , $Y = \{y_1, y_2, ..., y_d\}$

nous visons à sélectionner les features $f_i, i = 1..l, l << d$ (d le nombre total de features) qui maximisent les informations avec les labels $y_i, i = 1, .., d$ dans la base de données X. Un score élevé de $MI(f_i, y_j)$, avec $i = 1, ..n$ et $j = 1, ..d$, signifie un meilleur pouvoir discriminant de f_i pour la prise de décision y_j. Par conséquent , les features sont triés dans l'ordre décroissant de leurs scores d'Information Mutuelle, pour éliminer ainsi les features de faible score et par suite de faible contribution dans la décision.

4 Le Design Expérimental

4.1 Description du Dataset

Nous avons utilisé les jeux de données UNSW-NB 15 et NSL-KDD. Ces Deux Data set ont été utilisés pour vérifier l'efficacité de l'approche proposée car ils sont largement utilisés dans l'évaluation des techniques de détection d'anomalies [17] [18].

4.2 NSL-KDD dataset

Le jeu de données NSL-KDD présente la version affinée du Data set KDDcup99. Elle est constitué d'une grande quantité de données qui dépasse 7 millions d'enregistrement. L'ensemble de données NSL-KDD qui était utilisé dans cet étude pour l'entrainement et le test est 10 % de l'ensemble de données originale, cela équivaut à 494 020 vecteurs de connexion qui sont labellisés comme étant attaque ou normale pour l'entrainement, et 311030 vecteurs d'enregistrements pour le test. Dans cette étude, 14 nouvelles attaques différentes de celles utilisées dans le jeu de données d'apprentissage ont été employés dans l'ensemble de test. Les performances de notre approche peuvent être évaluées à la fois par des attaques invisibles et connues, rendant l'analyse de détection d'attaque plus réaliste [19].

FIGURE 2: la distribution des classes dans le Data set NSL KDD

Chaque vecteur d'enregistrement dans la base de données est classé comme attaque ou normal et comprend 41 features continues et nominales. Ces features peuvent être classées en quatre catégories différentes :

- La première catégorie, contenant les features de 1 à 9, qui sont les features de base des connexions TCP individuelles.
- La seconde catégorie est intitulée du feature 10 à 22 et correspond aux features de contenu.
- La troisième catégorie nommée du feature 23 à 31 correspond au spécificités de trafic calculées en utilisant une fenêtre de deux secondes.
- la quatrième catégorie allant du feature 32 à 41 est une entité de trafic calculée à l'aide d'une fenêtre de deux secondes entre la destination et l'hôte.

Une liste des features NSL-KDD avec des descriptions détaillées est répertoriée dans le Tableau1.

TABLE 1: Les Différents Groupes de Features dans NSL KDD Dataset

Feature Name	Description
duration	Length(number of seconds)of the connection
protocol_type	Type of the protocol,e.g.tcp,udp,etc.
service	Network service on the destination ,e.g.,http,telnet,etc.
flag	Normal or error status of the connection
src_bytes	Number of data bytes from source to destination
dst_bytes	Number of data bytes from destination to source
land	1 if connection is from /to the same host/port ;0 otherwis
wrong_fragment	Number of wrong fragments
urgent	Number of urgent packets
hot	Number of hot indicators
num_failed_logins	Number of failed login attempts
logged_in	1 if successfully loggedin ;0otherwise
num_compromised	Number of compromised conditions
root_shell	1 if root shell is obtained ;0 otherwise
su_attempted	1 if suroot command attempted ;0 otherwise
num_root	Number of root accesses
num_file_creations	Number of file creation operations
num_shells	Number of shell prompts
num_access_files	Number of operations on access control files
num_outbound_cmds	Number of out bound commands in an ftp session
is_host_login	1 if the login belongs to the hot list ;0 otherwise
is_guest_login	1 if the login is a guest login ;0 otherwise
count	Number of connections to the same host as the current connection in the past two seconds
srv_count	Number of connections to the same service as the current connection in the past two second
serror_rate	% of connections that have SYN_ errors
srv_serror_rate	% of connections that have SYN errors
rerror_rate	% of connections that have REJ errors
srv_rerror_rate	% of connections to different hosts
same_srv_rate	% of connections to the same service
diff_srv_rate	% of connections to different services
srv_diff_host_rate	
dst_host_count	Count for destination host
dst_host_srv_count	Srvcount for destination host
dst_host_same_srv_rate	Samesrvrate for destination host
dst_host_diff_srv_rate	Diffsrvrate for destination host
dst_host_same_src	Same src port rate for destination host
dst_host_srv_diff_host	Diffhostrate for destination host
dst_host_serror_rate	Serrorrate for dest_host
dst_host_srv_serror_rate	Srvserrorrate for dest_host
dst_host_rerror_rate	Rerrorrate for dest_host
dst_host_srv_rerror	Srvserrorrate for destination host

4.3 UNSW-NB15 Dataset

Ces données ont été créées par the security product IXIA Perfect Storm dans le Cyber Range Lab du Centre australien de cyber sécurité (ACCS) [17] .L'objectif était de générer un ensemble hybride constitué d'activités réelles modernes normales et de comportements d'attaque synthétiques contemporains. Il a été signalé que cet ensemble résolvait plusieurs problèmes rencontrés dans les ensembles de données KDD99 et NSLKDD [20]. En particulier ceux concernant les types d'attaques obsolètes, les scénarios de trafic licites obsolètes et le déséquilibre entre les instances d'entrainement et du test [21].

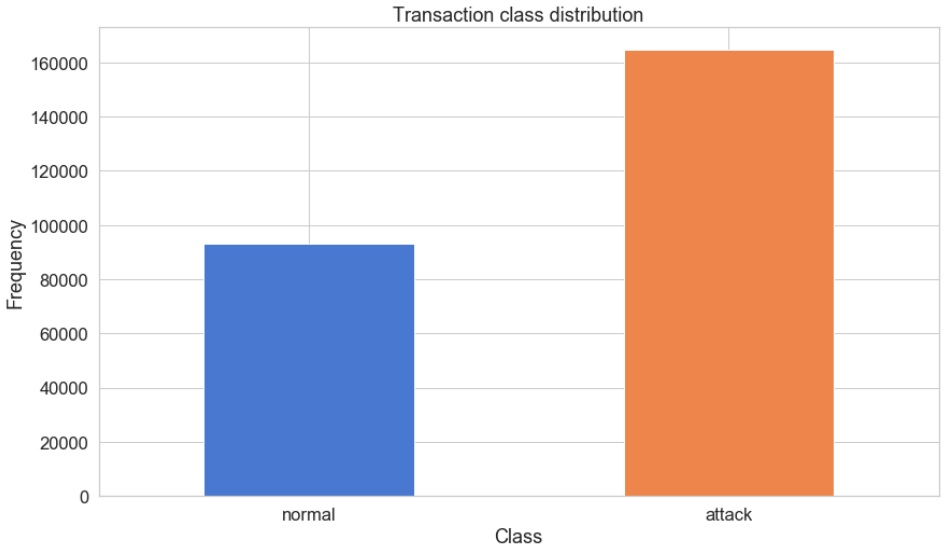

FIGURE 3: La distribution des classes dans UNSW-NB Data set

UNSW-NB15 contient neuf catégories d'attaques et 48 features ainsi que l'étiquette de classe qui indique si l'enregistrement est un trafic normal ou une attaque. L'ensemble de données contient 2540 044 enregistrements stockés dans 4 fichiers csv : UNSW-NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv et UNSW-NB15_4.csv. Pour éviter la charge de calcul due à la grande taille des ensembles de données, une partition de cet ensemble de données est configurée comme ensemble d'apprentissage et d'essai, à savoir, respectivement, UNSW-NB15_training set.csv et UNSW-NB15_testing-set.csv. Le nombre d'enregistrements dans l'ensemble d'apprentissage est de 175 341 enregistrements tandis que l'ensemble de test comprend 82 332 enregistrements des différents types, attaque et normal. Une liste de la plupart des features UNSW-NB15 avec des descriptions détaillées est répertoriée dans le tableau 2.

TABLE 2: Exemples des Features dans UNSW-NB15 Dataset

Feature Name	Description
srcip	Source IP address
sport Source	port number
dstip Destination	IP address
dport Destination	port number
proto	Transaction protocol (ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN....)
state	State Protocol dependant
dur	Total duration
sbytes	Source to destination transaction bytes
dbytes	Destination to source transaction bytes
sttl	Source to destination Time To Live (TTL)
dttl	Destination to source Time To Live (TTL)
sloss	Source packets retransmitted or dropped
dloss	Target packets retransmitted or dropped
service	http, ftp, smtp, ssh, dns, ftp-data ,irc, and (unusual service)
sload	Source bits per second
dload	Destination bits per second
spkts	Source to destination packet count
dpkts	Destination to source packet count
swin	Source TCP window advertisement value
dwin	Target TCP window advertisement value
stcpb	Source TCP base sequence number
dtcpb	Destination TCP base sequence number
smeansz	Mean packet size transmitted by the src
dmeansz	Mean packet size transmitted by the dst
trans_depth	Pipelined depth into the connection of HTTP request /response transaction
res_bdy_len	Size of uncompressed data transferred from the server's HTTP service
sjit	Source jitter (ms)
djit	Destination jitter (ms)
stime	Start time
ltime	Last time
sintpkt	Source interpacket arrival time (ms)
dintpkt	Destination interpacket arrival time (ms)
tcprtt	TCP connection setup round-trip time : sum of 'synack' and 'ackdat'
synack	TCP connection setup time : time between SYN and SYN_ACK packets
ackdat	TCP connection setup time : time between SYN_ACK and ACK packets
is_sm_ips_ports	1 if source and destination IP addresses and ports equal, 0 otherwise
ct_state_ttl	Number for each state according to specific range of values for source /target TTL
ct_flw_http_mthd	Number of flows with methods such as GET and PORT in HTTP service
is_ftp_login	1 if FTP session is authenticated, 0 otherwise
ct_ftp_cmd	Number of flows with a command in the FTP session

Il est important de mentionner que les jeux de données NSL KDD et UNSW-NB15 ne partagent que quelques fonctionnalités communes et que les autres features sont totalement diffé-

rentes.
Enfin le nombre de features à traiter est trop élevé , d'où le choix d'appliquer l'algorithme de sélection des features Mutuelle Information.

4.4 Critères D'Évaluation De la Performance

Les vrais positifs (TP) sont les cas classés comme une intrusion et qui sont en réalité une intrusion. Les faux positifs (FP) sont les cas classés comme des intrusions, mais en réalités sont des cas légitimes.les faux négatifs (FN) sont des intrusions classées comme du trafic légitime. Tant dis que les vrais négatifs(TN) sont des cas légitimes et classés comme légitimes. Dans cette expérience, nous comparons les performances de notre modèle en utilisant les sous-ensemble réduits de features sélectionné avec celles contenant toutes les features. Pour l'évaluation nous avons utilisé le *PR Curve* et les métriques : *Accuracy,Recall,Precision et F-score*. Le choix de ces métriques étant en grande partie lié au fait que ce sont les plus utilisés dans le domaine de detection des intrusions.

- ❄ Accuracy : est la proportion des événements correctement classés parmi l'ensemble des événements proposés, qu'ils soient correctement ou incorrectement classés, comme indiqué dans l'équation suivante :
  $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

- ❄ Recall : On l'appelle aussi taux de vrais positifs. Il est utilisé pour mesurer la proportion de résultats positifs correctement identifiés parmi l'ensemble des événements correctement classés :
  $Recall = \frac{TP}{TP+FN}$

- ❄ Precision : est la proportion de vrais positifs détectés parmi les cas classés comme intrusions.
  $Precision = \frac{TP}{TP+FP}$

- ❄ score-F : peut être interprétée comme une moyenne harmonique pondérée de la Precision et du Recall, le score-F atteint sa meilleure valeur à 1 tandis que le score le plus faible est 0.
  $F-measure = 2\frac{precision.recall}{precision+recall}$

- ❄ Precision-Recall curve :une métrique qui met l'accent sur la capacité du classificateur à identifier les événements malveillant et ignore les événements normaux correctement classés (TN),il est utilisé en particulier lorsque les classes sont très déséquilibrées.
  La courbe de PR montre le compromis entre la Precision et le Recall. Une grande surface sous la courbe représente à la fois une précision et un Rappel élevé, le meilleur scénario pour un classificateur, montrant un modèle renvoyant des résultats précis pour la majorité des classes sélectionnées.

5 Solution Proposée

Le système de detection d'intrusion est divisé en deux parties principales : La sélection des features , et La transformations des événements en vecteurs embedding .(Voir Figure.4)

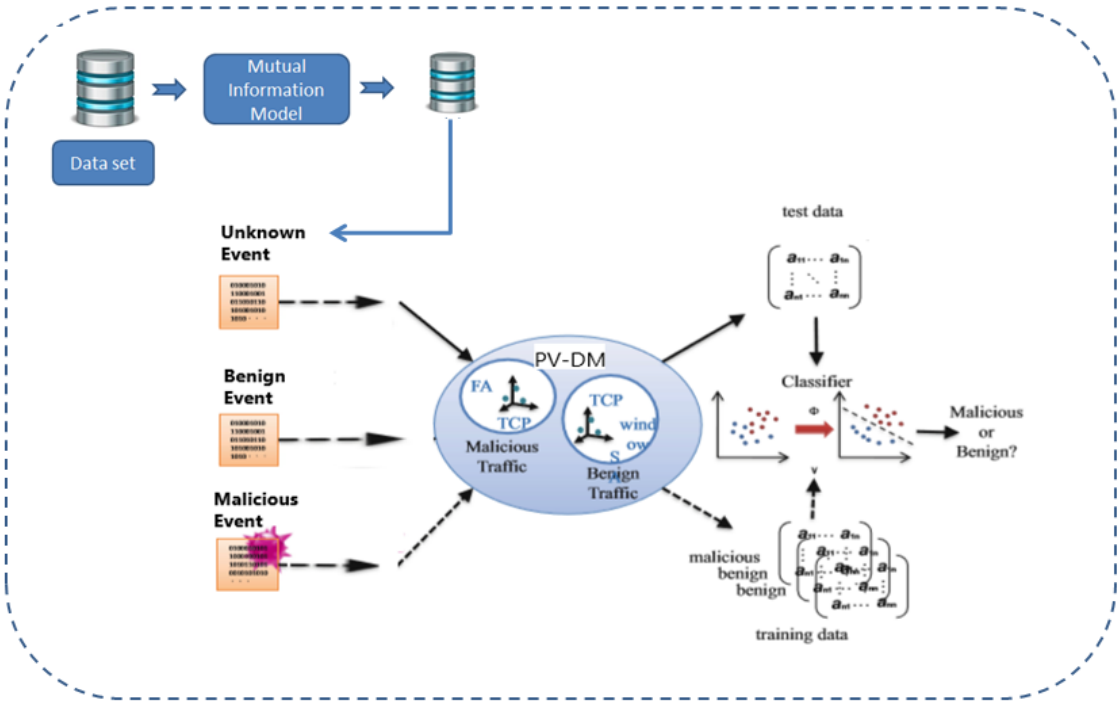

FIGURE 4: Architecture du système d'intrusion proposé

5.1 La Sélection Des Features

Nous utilisons l'algorithme Mutual Information (MI) pour sélectionner les meilleur sous-ensemble de features des Dataset NSL KDD et UNSW-NB15.
Nous présumons que La sélection des meilleures features avec des valeurs MI élevées, (donc grande capacité pour classer les événements (transactions) en différentes classes), permet à notre modèle de centrer sur ce qui est le plus pertinent pour la modélisation du comportement du système informatique.
Cette hypothèse a été confirmée expérimentalement ,en effet les ensembles des features réduits ont des performances identique ou même supérieure à celles contenant toutes les features .
L'algorithme Mutuelle d'Information n'utilise que des valeurs numériques pour les processus d'entrainement et de test.Par suite, pour convertir les features non numériques (comme **protocol_type**, **service** et **flag** dans la base de données NSL KDD, **proto**, **service** et **state** dans UNSW-NB15) en valeurs numériques, nous avons utilisons la méthode *Pandas factorize-python* [22].
Ensuite , Nous appliquons l'algorithme de Mutuelle Informations sur les Dataset en suivant les étapes suivantes :

- **Initialisation** : définir F ensemble initial de toutes les features, S un ensemble vide, Y ensemble des classes.
- **Conversion** des features de jeu de données non numériques en valeurs numériques.
- **Calcul** de l'information mutuelle (MI) des features avec les classes : pour chaque entité ($f_i \in F$), calculez MI (f_i;y) et ($y \in Y$).
- **Sélection** des meilleures features présentant les valeurs MI les plus élevées (plus grand d'un seuil prédéfinie).
- **Output** : les sous ensembles contenant les features sélectionnées.

5.2 Le Modèle Deep Learning PV-DM

L'idée clé de notre approche est la lecture des paquets de réseau par l'algorithme PV-DM , qui construit un espace vectoriel à partir du base de données et convertit chaque événement malveillant

connu ou normale,en un vecteur avec son label. Ces vecteurs labellisés sont les données d'apprentissage pour les classificateurs. Nous utilisons la régression logistique (LR) et Random Forest (RF) comme classificateurs qui vont construire un modèle qui prédit les label des nouveaux événements. les nouveaux événements non labellisé sont convertit en vecteurs. Ces vecteurs sont les données de test pour les classificateurs entrainés, et qui vont prédire, si ces événements sont malveillantes, ou bénignes(normaux). Le modele PV-DM a été implémentée avec*Python-3.6.2* , avec des bibliothèques open source de machine learning, *Genism 3.7.0* [23] et *Scikit-Learn v0.21.dev0* [24] . *Scikit-Learn* est une bibliothèque d'apprentissage automatique pour *Python* qui fournit des outils pour l'exploration de données.tandis que *Genism* est une bibliothèque *Python* permettant de réaliser des modèles sémantiques non supervisés à partir des textes, et inclut un modèle appelé *Doc2vec*. Le tableau 3.5 présente les paramètres du modèle Doc2vec utilisé dans les expériences.

Paramètre	Valeur
Dimensionality of the feature vectors	100
Window	8
Number of epochs	50
Training algorithm	PV-DM

TABLE 3: les paramètres de Doc2vec modele utilisée dans notre approche

Nous avons défini la dimensionnalité des vecteurs de chaque événement à 100 et choisi PV-DM qui est une variante de Doc2vec comme algorithme d'entrainement.
La fenêtre ou le Window désigne la distance maximale entre le mot prédit et les mots de contexte utilisés pour la prédiction dans un document.
Afin d'évaluer les performances de notre approche, nous avons mené plusieurs expériences sur les sous-ensembles de features réduites (tableaux 3 et 4), les résultats ont été comparés à celles de l'ensemble de données original, c'est-à-dire sans réduction . Les résultats expérimentaux montrent que les performances ont augmenté lorsque la réduction des features est effectuée.

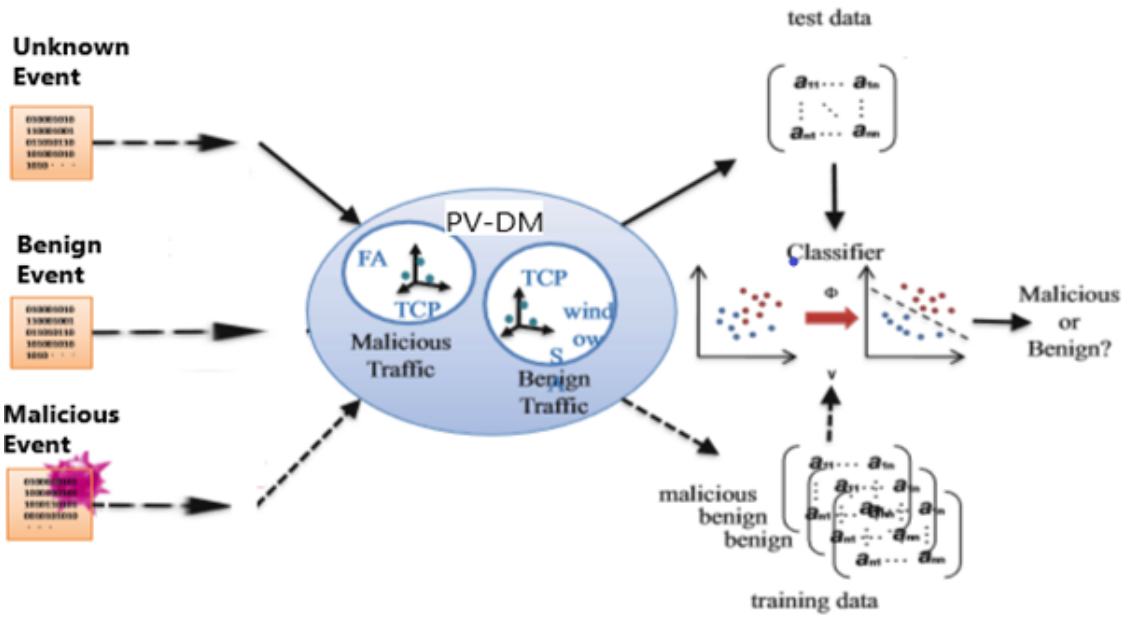

FIGURE 5: Application du modèle PV-DM sur les événements

6 Les Résultats Expérimentaux

Dans cette section, nous présentons les résultats de performance obtenus par l'algorithme Mutuelle Information , le modèle PV-DM , et les classificateurs Logistic Regression et Random Forest

sur les Data set NSL KDD et UNSW-NB15.
Les expériences ont montré que les features :**rate, dttl, ct_state_ttl,dbytes,sttl et sbytes** dans le Dataset UNSW-NB15 et les features :**service, src_bytes, dst_bytes, count, et dst_host_same_src-_port_rate** dans NSL KDD sont les features qui maximisent MI avec les classes (Fig 6, Fig 7).
Enfin, les sous-ensemble de features sélectionné à partir des jeux de données NSL KDD et UNSW-NB15 sont présentés dans les tableaux 3 et 4.

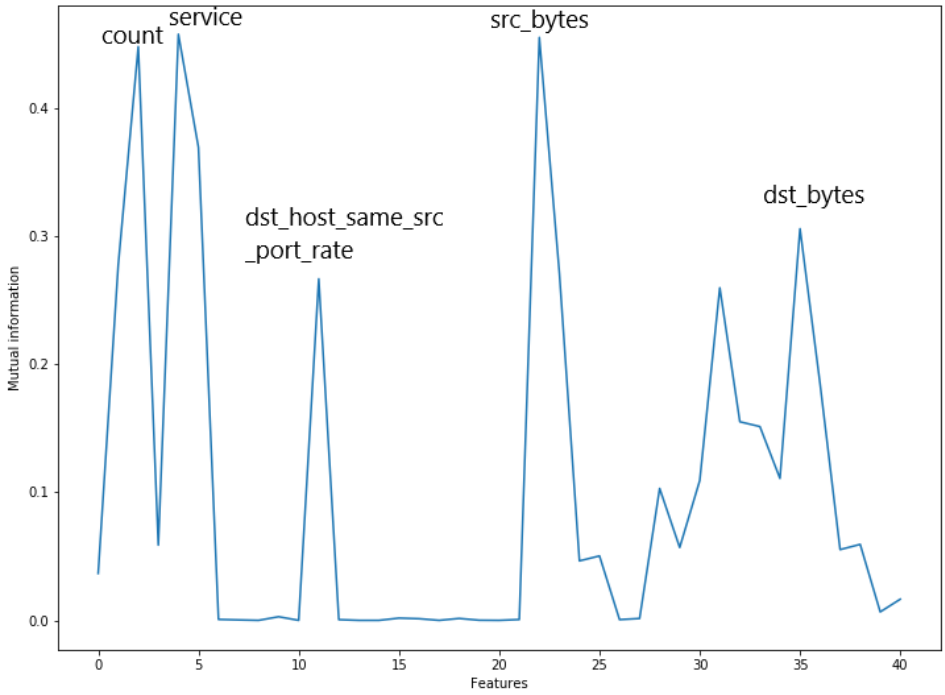

FIGURE 6: Les scores de mutuelle Information des caractéristiques de NSL KDD Dataset

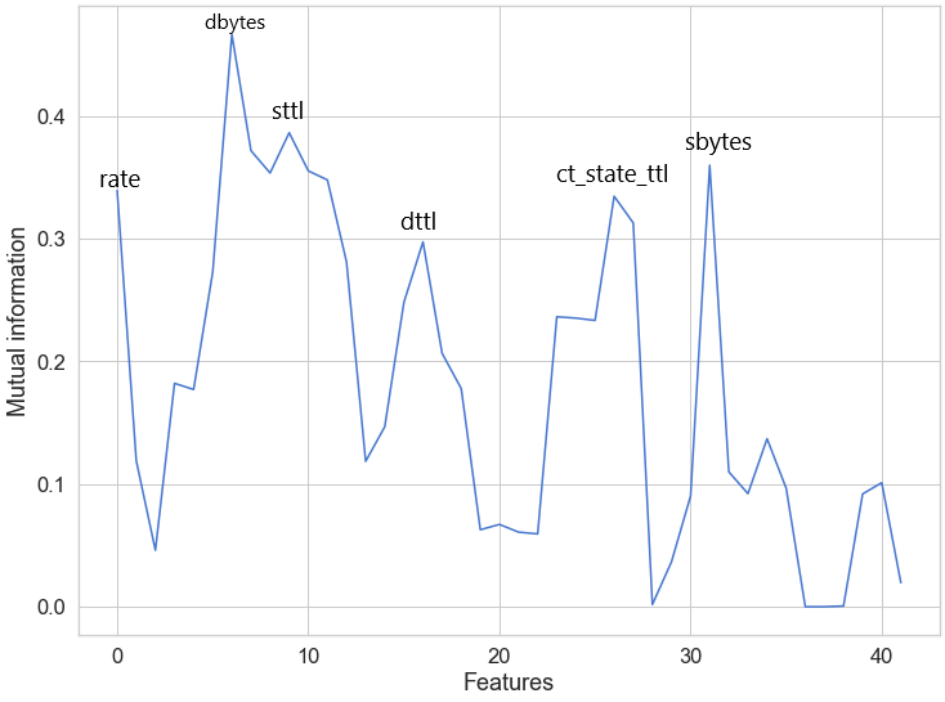

FIGURE 7: Les scores de mutuelle Information des caractéristiques de UNSW-NB15 Dataset

subset	Features
subset 1	service,src_bytes,dst_bytes,count, dst_host_same_src_port_rate
subset 2	service, count et src_bytes

TABLE 4: Les ensembles réduits de caractéristiques sélectionnés dans NSL KDD.

subset	Features
subset 1	rate,dttl,ct_state_ttl,dbytes,sttl,sbytes
subset 2	dbytes, sttl, sbytes

TABLE 5: Les ensembles réduits de caractéristiques sélectionnés dans UNSW-NB15

Les tableaux (6, 7) et les figures (8,9,10,11) montrent les résultats empiriques obtenus par la méthode proposée avec les sous ensembles réduits de features et les ensembles originales présentes dans NSL KDD et UNSW-NB15.
Nous pouvons voir que les performances du modèle proposé avec les sous ensembles réduits est meilleur qu'avec les ensembles sans réduction . Cela prouve que certaines features sont inutiles ou redondantes et que leur suppression améliore généralement les performances du modèle de classification.
En fait, le tableau 6 montre que parmi les 48 features disponibles dans le jeu de données UNSW-NB15, seules six sont nécessaires pour avoir les même performances .
De même, le tableau 7 montre que parmi les 41 features présentes dans le jeu de données NSL KDD ; cinq features seulement suffisaient pour notre modèle pour obtenir des performances élevées. En outre, on remarque que les performances de notre modèle sur l'ensemble de test du NSL KDD (tableau 7) sont supérieures à celles de notre modèle sur l'ensemble de test du UNSW-NB15. Cela peut être dû au fait que le test UNSW-NB15 inclut les même types d'attaques que l'ensemble d'entrainement UNSW-NB15, tandis que ce n'est pas le cas pour le jeu de données NSL KDD.En effet,l'ensemble de données de test NSL KDD comporte 14 types d'attaques de plus que l'ensemble de données d'entrainement NSL KDD.

Subset	Classifier	Precision	Accuracy	Recall	F1_score
All features(48 Feature)	Logistic Regression	66.46	69.44	89.83	76.39
	Random Forest	70.91	74.12	89.87	79.27
Subset1 :(6 features)	Logistic Regression	70.33	74.65	93.33	80.21
	Random Forest	79.59	84.17	95.81	86.96
Subset 2 :(3 features)	Logistic Regression	70.66	74.76	92.63	80.16
	Random Forest	76.82	81.74	95.72	85.23

TABLE 6: Les Performances de classification sur l'original test data UNSW-NB15, et les sous ensembles réduits

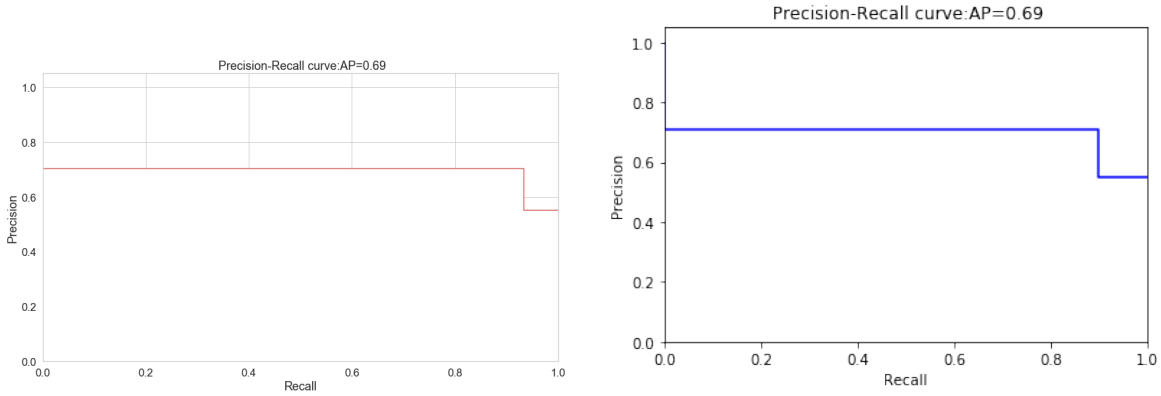

FIGURE 8: Area Under the Precision Recall curve capturant les performances des classificateurs LR et RF sans réduction de features dans UNSW_NB15.

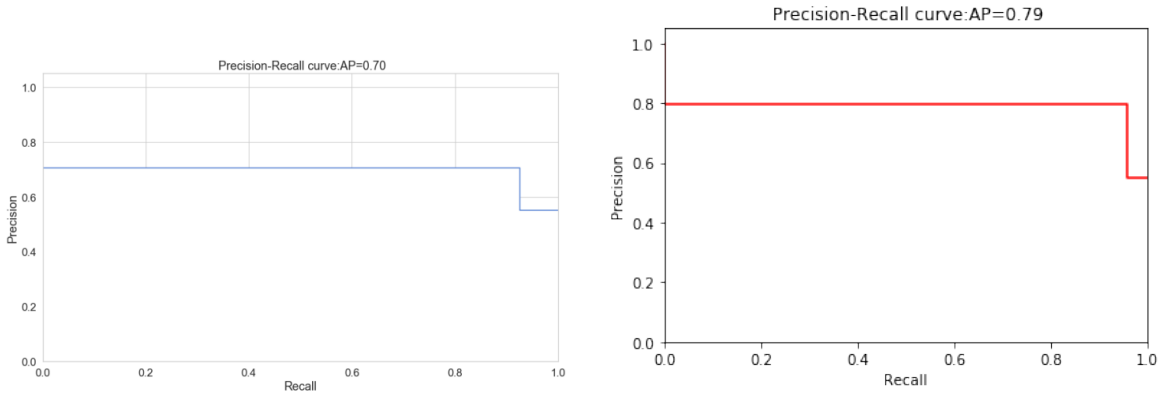

FIGURE 9: Area Under the Precision Recall curve capturant les performances des classificateurs LR et RF avec la réduction de features dans UNSW_NB15.

Nous remarquons également que les résultats de l'exécution du sous-ensemble 2 par rapport au sous-ensemble 1, soit dans UNSW-NB15, soit dans NSL KDD (tableaux 6 et 7), indiquent que le sous-ensemble 2 produit des résultats très proches de celle du sous-ensemble 1, malgré que le nombre de features utilisées sont moins ; comparé à ceux du sous-ensemble 1. Cela indique que dans NSL KDD, les features les plus significatives sont : **service, count et src_bytes** ,alors que dans UNSW-NB15 les importants features sont : **dbytes, sttl,et sbytes** .

Subset	Classifier	Precision	Accuracy	Recall	F1_score
All features (41)	Logistic Regression	89.74	84.87	91.68	90.70
	Random Forest	94.54	84.22	85.32	89.69
Subset 1 :(5 features)	Logistic Regression	95.46	89.37	91.13	93.24
	Random Forest	98.43	90.31	89.39	93.69
Subset 2 :(3 features)	Logistic Regression	95.88	89.03	90.25	92.98
	Random Forest	98.70	91.36	90.45	94.40

TABLE 7: Les Performances de classification sur l'original test data NSL KDD, et les subset réduits.

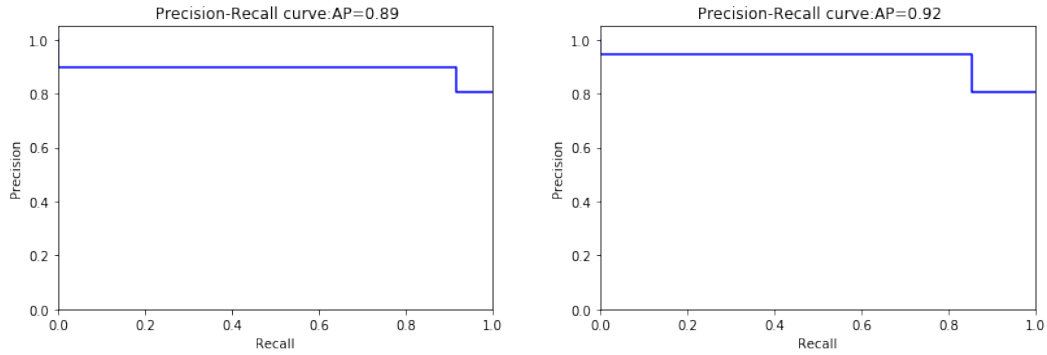

FIGURE 10: Area Under the Precision Recall curve capturant les performances des classificateurs sans réduction de features dans NSL KDD.

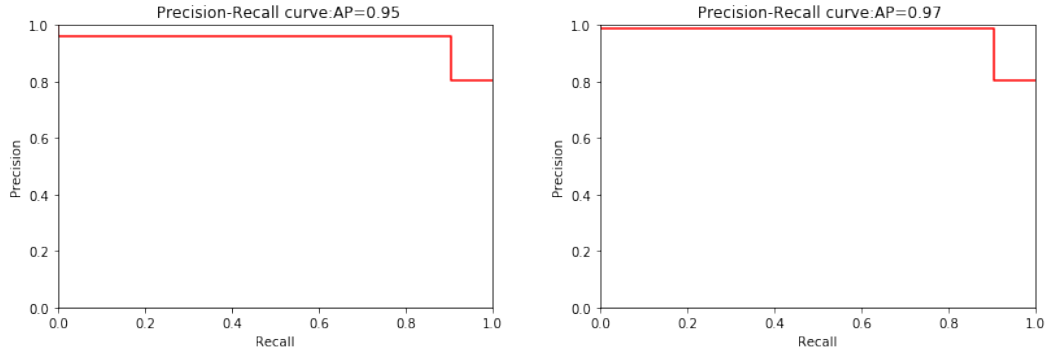

FIGURE 11: Area Under the Precision Recall curve capturant les performances des classificateurs avec réduction de features dans NSL KDD.

7 Conclusion

Le but de nos travaux dans ce chapitre est de proposer un nouveau système de détection d'intrusion basé sur l'algorithme d'Information Mutuelle pour la sélection des features et sur le modèle Deep Learning PV-DM pour la lecture des packets réseaux.
Les apports de notre système de detection d'intrusion proposé sont :

- ❄ Nous montrons que seules quelques features sont nécessaires pour détecter les événements malveillants et obtenir de hautes performances, cela entraîne à son tour moins de stockage nécessaire des données et exécuter l'algorithme.

- ❄ Ces performances ont été obtenues dans un environnement totalement non supervisé, c'est-à-dire sans aucune connaissance préalable de ce qui constitue une attaque. Par conséquent, si les attaquants modifient les techniques ou les protocoles d'attaques, nos méthodes apprennent automatiquement les features.
  Dans nos travaux futurs, nous prévoyons d'appliquer différents algorithmes de classification basés sur l'apprentissage automatique, tels que l'arbre de décision , le classificateur SVM ou CNN, afin d'améliorer les performances de notre approche. Nous envisageons aussi d'explorer autres techniques de détections des intrusions qui confèrent plus de flexibilité ce qui rend possible la prise en compte des imprécisions et des incertitudes. parmi ces techniques on peut citer Fuzzy logic (logique flou),ce qui permet la modélisation des imperfections des données et se rapproche dans une certaine mesure de la flexibilité du raisonnement humain.

Bibliographie

[1] A. Lazarevic et al. A comparative study of anomaly detection schemes in network intrusion detection. Proceedings of the 2003 SIAM International Conference on Data Mining, page Pages 25-36, 2003.

[2] K.Barotetal. Using natural language processing models for understanding network anomalies. Proceedings of the Twentieth Conference HPEC, page 7, September 2016.

[3] Q.Le and T.Mikolov. Distributed representations of sentences and documents. Proceedings of the 31 st International Conference on Machine Learning, 2014.

[4] Xiaoyan Zhuo et al. Network intrusion detection using word embeddings. Proceedings of IEEE conference on Big Data, Dec 2017.

[5] Rafael San Miguel Carrasco and Miguel-Angel Sicilia. Unsupervised intrusion detection through skip-gram models of network behavior. ,Computers & Security, pages 187–197, jan 2018.

[6] Samira Douzi et al. Towards a new spam [U+FB01]lter based on PV-DM (paragraph Vector Distributed Memory Approach. Procedia Computer Science, pages 486–491, 2017.

[7] Paccanaro Alberto and Hinton Geoffrey. Learning distributed representations of concepts using linear relational embedding. IEEE Transactions on Knowledge and Data Engineering, pages 232–244, April 2001.

[8] Christophe Bertero et al. Experience report : Log mining using natural language processing and application to anomaly detection. 28th International Symposium on Software Reliability Engineering, 2017.

[9] Tomas Mikolov et al. Distributed representations of words and phrases and their compositionality. arXiv :1310.4546v1 [cs.CL], page 9, Oct 2013.

[10] Tomas Mikolov et al. Ef[U+FB01]cient estimation of word representations in vector space. arXiv :1301.3781v3 [cs.CL], Sep 2013.

[11] Tomas Mikolov et al. Exploiting similarities among languages for machine translation. arXiv :1309.4168v1 [cs.CL], page 24, Sep 2013.

[12] Andreas G.K.Janecek. On the relationship between feature selection and classi[U+FB01]cation accuracy. Proceedings of the Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery at ECML/PKDD, page 16, 2008.

[13] K. El-Khatib. Impact of feature reduction on the ef[U+FB01]ciency of wireless intrusion detection systems. IEEE Transactions on Parallel and Distributed Systems, pages 1143–1149, Aug 2010.

[14] Jaesung Lee and Dae-Won Kim. Fast multi-label feature selection based on information-theoretic feature ranking .Pattern Recognition ,pages2761–2771,2015.

[15] MohamedBennasaretal. Feature selection using joint mutual information maximisation. Expert Systems with Applications, pages 8520–8532, 2015.

[16] R. Battiti. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks, page 537-550, juill 1994.

[17] N.MoustafaandJ.Slay. Unsw-nb15 : a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). Military Communications and Information Systems Conference (MilCIS), pages 1–6, 2015.

[18] Sara Mohammadi. Cyber intrusion detection by combined feature selection algorithm. Journal of Information Security and Applications, pages 80–88, jan 2019.

[19] MIT Lincoln Laboratory. 1998 darpa intrusion detection evaluation dataset.

[20] Ressources Ixia. www.ixiacom.com. https ://www.ixiacom.com/fr/resources.

[21] Moustafa Nour and Slay Jill. The evaluation of network anomaly detection systems : Statistical analysis of the unsw-nb15 data set and the comparison with the kdd 99 data set. Information Security Journal :A Global Perspective, pages 1–14, jan 2016.

[22] Pandas 0.24.1 documentation. pandas.factorize. https ://pandas.pydata.org/pandasdocs/stable/reference/api/pandas.factorize.html.

[23] gensim PyPI. gensim. https ://pypi.org/project/gensim/.

[24] scikit learn. machine learning in python-scikit-learn 0.20.2 documentation. https ://scikit-learn.org/stable/index.html/.

Chapitre 5

L'Extension de L'IDS Learning avec Fuzzy Logic et L'algorithme Weighted Fuzzy C Mean

1 Motivation

Les systèmes informatiques sont de plus en plus répandus et utilisés pour transférer de nombreuses informations sensibles entre de nombreux types de périphériques, des serveurs énormes aux terminaux mobiles en passant par les mini-ordinateurs.
Bien que de nombreux types de méthodes de sécurité, tels que le contrôle d'accès, le cryptage et les pare-feu, soient utilisés, les atteintes à la sécurité du systèmes informatiques augmentent jour après jour [1].Par suite, il existe un besoin urgent de systèmes de détection d'intrusion intelligents (IDS) pour détecter automatiquement les nouvelles intrusions.
Il existe deux méthodes principales de détection d'intrusion :**la misuse détection** et **la détection d'anomalie**.

- La misuse détection compare les activités du système avec des signatures ou des modèles prédéfinis tirés d'éléments caractéristiques qui représente une attaque spécifique [2] [3].Cette méthode de détection des intrusions peut détecter des attaques avec un taux de faux positifs (fp) faible, mais il ne peut pas découvrir de nouvelles attaques.

- La détection d'anomalie découvre les attaques en identifiant les écarts par rapport aux activités normales du réseau [4][5]. Cette Anomalie détection peut découvrir de nouvelles attaques mais avec un taux de faux positifs élevé [2].
  Pour éviter les inconvénients de misuse et de l'anomalie techniques, de nombreux IDS actuels sont des systèmes qui reposent sur des règles qui sont collectées et identifiées par des experts en sécurité.Ces IDS modélisent les connaissances collectées sur les événements des systèmes suspects, ce qui permet de parcourir les données du trafic réseau pour rechercher des preuves des vulnérabilités existantes.[5] [6].
  Il ne fait aucun doute que le codage manuel des règles est un processus très coûteux en temps et en argent. De plus,il dépend de l'efficacité des experts humains dans l'analyse d'une énorme quantité d'activités de système pour découvrir des modèles d'intrusion. Cependant,ces inconvénients sont surmontés en utilisant de nombreuses techniques de Data mining dans les IDS [7] [8].Le Data mining ou (L'exploration de données) consiste à analyser des ensembles d'observation (souvent volumineux) afin de rechercher des relations insoupçonnées et de résumer les données de manière innovante, à la fois compréhensible et utile pour le propriétaire des données [9].

2 Travaux Connexes

Snort [13] est l'un des IDS open source les plus répandus,et basé sur les règles d'associations . Ses règles reconnaissent les paquets réseau malveillants en faisant correspondre le paquet actuel à des règles prédéfinies. il produit un False positive rate(FPR) élevé en raison de sa méthodologie d'identification des signatures d'attaque. Mais il ne peut pas détecter *zero-day attacks* . Actuellement, Snort implique plus de 20 000 règles qui sont généralement mises à jour par les utilisateurs. Vaccaro et al. [14] ont proposé un système de détection des intrusions qui identifie les comportements malveillants en établissant un ensemble de règles, décrivant statistiquement les comportements des utilisateurs, en se basant sur les journaux de leurs activités en une certaine période. Il fait ensuite correspondre l'activité en cours aux règles stockées pour détecter les comportements suspects. Le schéma de détection des Outliers (les valeurs aberrantes), est l'une des techniques de Data mining, tente d'identifier un point de données très différent du reste des données. A.Lazarevic et al.l'ont appliqué à la détection d'anomalies. Ils ont comparé plusieurs variantes d' algorithme de détection des Outliers dans leur expérience, ils ont prouvé que les outliers locales (LO) réalisent la meilleure performance.
RIPPER [16], un outil d'apprentissage de règles, a été utilisé pour la construction de modèles de détection. RIPPER est appliqué aux ensembles de données d'entraînement et extrait automatiquement les schémas des intrusions dans le système suspect. Bien que ce soit un bon outil pour découvrir des modèles des intrusions , il ne peut pas être facilement appliqué à une anomalie détection afin de détecter de nouvelles intrusions.
Il y' a un inconvénient commun à ces techniques de detection basé sur le Data mining, c'est le temps considérable nécessaire à la formation et à l'apprentissage du modèle.

Le but de ce chapitre est de proposer une extension de L'IDS Learning proposé dans le travail précédent(voir chapitre 3), en ajoutant un second composant de detection basé sur le Fuzzy logic.
L'une des raisons de cette extension est qu'il n'est pas techniquement possible de construire un système sans aucune vulnérabilité. En fait,il est presque impossible d'intégrer tous les modèles d'intrusion et les futurs attaques qui pourraient utiliser des schémas complètement inconnus et difficiles à détecter.D'où l'insuffisance d'une seule méthode de detection, en dépit de ces performances.
La deuxième raison est l'idée d'introduire la logique floue pour la détection d'intrusion , vu que la sécurité elle-même inclut le flou. Étant donné une mesure quantitative,où un intervalle peut être utilisé pour désigner une valeur normale de cette mesure. Par suite, toutes les valeurs situées en dehors de l'intervalle seront considérées comme des anomalies au même degré, quelles que soient leurs distances différentes par rapport à l'intervalle(normal). La même chose s'applique aux valeurs à l'intérieur de l'intervalle, c'est-à-dire que tout les valeurs seront vu normales au même degré. ce qui provoque une séparation abrupte entre la normalité et l'anomalie. Par exemple, une valeur à l'intérieur de la bordure est supposée normale alors qu'une autre valeur à l'extérieur de la frontière est supposée anormale même s'il n'y a qu'une très petite différence entre ces deux valeurs. L'introduction de Fuzzy logique dans ces features quantitatives aidera à atténuer la séparation abrupte. L'hypothèse de ce travail est que la logique floue est capable de produire des règles plus générales qui augmenteront la flexibilité des systèmes de détection d'intrusion.

3 Concepts de Base de notre approche

3.1 Fuzzy Logic (la logique floue)

L'homme perçoit, raisonne, imagine et décide à partir de modèles ou de représentations. Sa pensée n'est pas binaire. L'idée de la logique floue est de capturer l'imprécision de la pensée humaine et l'exprimer avec des outils mathématiques appropriés.La logique classique

basée sur les deux valeurs de vérité, vrai ou faux est parfois inadéquate avec des informations incomplètes et peu fiables, et est donc incapable de prendre une décision.

En 1965 Lotfi Zadeh [16], de l'université de Berkeley aux USA, a publié l'article « Fuzzy sets » dans lequel il a développé la théorie des ensembles flous et introduit le terme fuzzy dans la littérature technique. L'idée de Zadeh consiste à utiliser le modèle de l'esprit humain qui dispose d'une très forte capacité pour appréhender la complexité et pour manier des notations vastes et imprécises. Cette compétence est due à l'habilité des humains à manipuler des informations floues.
Depuis, la logique floue s'est confirmée comme étant un outil adéquat pour le traitement des imprécisions et des incertitudes dans les systèmes intelligents. Au niveau industriel, les différentes applications de la logique floue ont bien montré son utilité dans beaucoup de domaines tels que la robotique , le contrôle des automatismes de processus et le diagnostic médical.

3.2 Les ensemble flou(Fuzzy set)

Une des notions fondamentales dans les mathématiques classiques est la notion d'ensemble, créé par le mathématicien Georg Cantor [17]. Il a définit les ensembles comme des collections d'objets, appelés éléments, bien spécifiés et tous différents. Dans la théorie des ensembles, un élément appartient ou n'appartient pas à un ensemble.
Ainsi, on peut définir un ensemble par une fonction caractéristique pour tous les éléments x de l'univers de discours U. L'univers de discours est l'ensemble référentiel qui contient tous les éléments qui sont en relation avec le contexte donné[18]. La fonction caractéristique de l'ensemble E , notée μE :

$\mu E : U \longrightarrow \{0,1\}$ est définie comme :

$$\mu E = \begin{cases} 1 \, si \, x \in E \\ 0 \, si \, x \notin E \end{cases}$$

Zadeh a étendu la notion d'un ensemble classique à l'ensemble flou qui le définit comme étant « une collection telle que l'appartenance d'un élément quelconque à cette collection peut prendre toutes les valeurs entre 0 et 1 ». La théorie des ensembles flous repose sur la notion d'appartenance partielle : chaque élément appartient partiellement ou graduellement aux ensembles flous qui ont été définis. Les contours de chaque ensemble flou ne sont pas « nets », mais « flous » ou «graduels » . Ainsi Dans la figure 1 par exemple l'élément z appartient totalement à l'ensemble B , tant que l'élément t appartient partiellement à B.

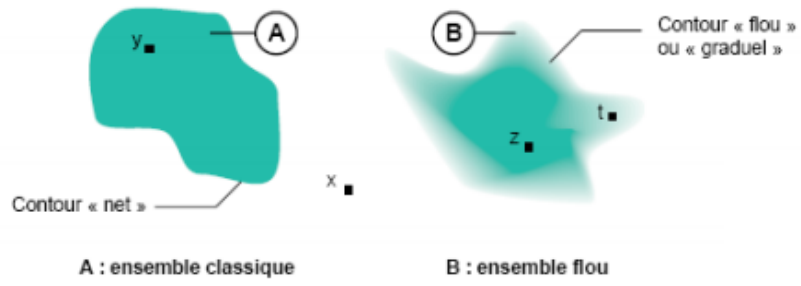

FIGURE 1: Comparaison d'un ensemble classique et d'un ensemble flou.

3.3 Fonctions d'appartenance(membership fonction)

Un ensemble flou(fuzzy set) est défini par sa « fonction d'appartenance », qui correspond à la notion de « fonction caractéristique » en logique classique. Supposons que nous vou-

lions définir un ensemble des personnes de « taille moyenne ». En logique classique, nous conviendrons par exemple que les personnes de taille moyenne sont celles dont la taille est comprise entre 1,60 m et 1,80 m. La fonction caractéristique de l'ensemble donne « 0 » pour les tailles hors de l'intervalle [1,60 m ; 1,80 m] et « 1 » dans cet intervalle.L'ensemble flou des personnes de « taille moyenne » sera défini par une « fonction d'appartenance » qui diffère d'une fonction caractéristique par le fait qu'elle peut prendre n'importe quelle valeur dans l'intervalle [0, 1]. A chaque taille possible correspondra un « degré d'appartenance » à l'ensemble flou des « tailles moyennes » , compris entre 0 et 1.

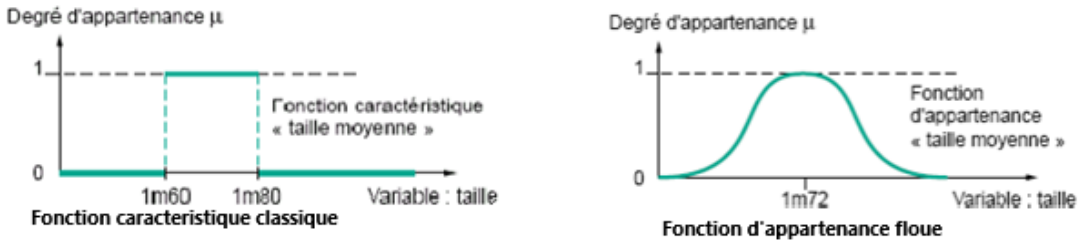

FIGURE 2: fonction caractéristique classique et floue.

3.4 Valeur d'appartenance(membership value)

La valeur d'appartenance est le degré de compatibilité d'un élément avec le concept qui est représenté par un ensemble flou. La fonction caractéristique de l'ensemble E .
$\mu E(x) : U \longrightarrow [0,1]$ est appelée « fonction d'appartenance ». La valeur $\mu E(x)$ mesure l'appartenance ou le degré avec lequel un élément x appartient à l'ensemble E :
$\mu E(x)$ = Degré$(x \in E)$

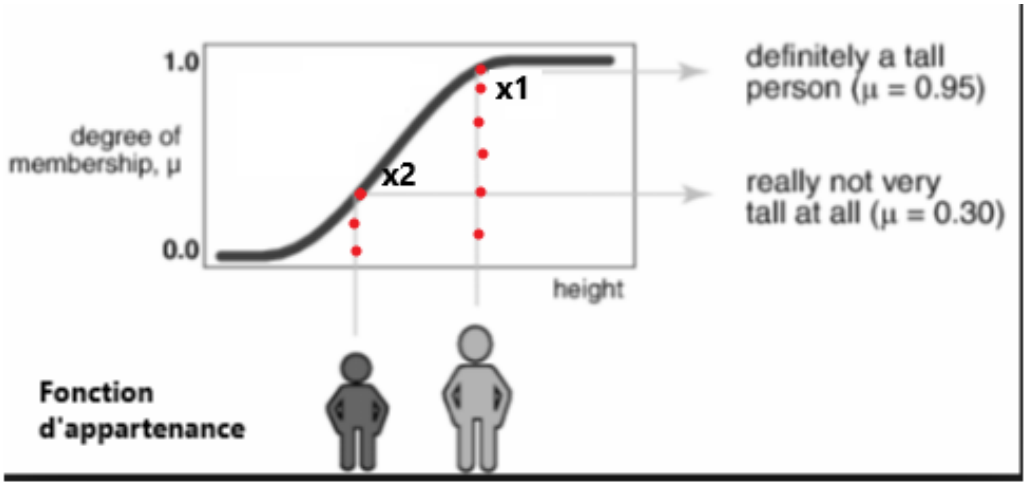

FIGURE 3: fonction caractéristique classique et floue.

Dans la figure.3 la valeur d'appartenance de$(x_1) = \mu(x) = 0.95$

3.5 Règles floue(Fuzzy association rules)

Une règle floue est une affirmation (Si… Alors) dont la prémisse et la conséquence sont des propositions floues ou des combinaisons de propositions floues par des connecteurs logiques (souvent 'ET' et 'OU'). Chaque règle a deux parties :

- partie antécédente (prémisse ou condition), exprimée par Si….

- partie conséquente (conclusion), exprimée par alors...

La partie antécédente est la description de l'état de système. La partie conséquente exprime l'action à exécuter. La forme générale est : Si (un ensemble de conditions est satisfait) alors (un ensemble de conséquences peut être exécuté). Une règle floue (Si... Alors) est représentée par une implication floue ayant la même fonctionnalité que celle utilisée dans la logique classique. Par exemple, la règle floue :
SI le feu est rouge et que je suis proche du feu et que ma vitesse est élevée, ALORS je freine fort.

3.6 la méthode de classification Weighted Fuzzy C-Means

3.6.1 La méthode k-means

La segmentation k-means est une méthode de classification automatique qui a pour objectif de partitionner l'espace en *k*classes, *k* connu.
A partir d'une partition initiale, on améliore itérativement la partition de l'espace en minimisant la variance et en maximisant l'écart entre les classes. On suppose que $X = \{x_1, ...x_n\}$, est une base de donnée de dimension n , X est divisée en k cluster. la fonction objectif de l'algorithme k-means peut être réécrite comme suit :

$$J(k,X) = \sum_{j=1}^{n} \min ||x_j - c_i||, \quad i = 1...k \tag{1}$$

Où c_i est le centre du i-ème cluster.
si la valeur k du nombre du clusters est bien choisi au départ L'algorithme renvoie une partition des données, dans laquelle les objets à l'intérieur de chaque cluster sont aussi proches que possible les uns des autres et aussi loin que possible des objets des autres clusters. la méthode k-means est très populaire car extrêmement rapide en pratique.cependant on doit choisir le nombre k de classes et il n'y a pas de critère unique pour déterminer le meilleur nombre de classes ; Lorsque les classes ne sont pas bien séparées, les k-means tendent à trouver uniquement des classes "sphériques", de même effectif et de même volume. Plusieurs initialisations peuvent conduire a plusieurs partitions très différentes dans leur composition.

3.7 la méthode de classification Fuzzy C-Means (FCM)

L'algorithme de Fuzzy C-Means ou des C-Moyennes Floues généralise l'algorithme des k-means en permettant la classification floue basée sur la notion d'ensemble flou.En fait La classification floue autorise le chevauchement des régions.tandis que Une segmentation non floue peut être obtenue par affectation de chaque item à la classe pour laquelle son degré d'appartenance est maximal.

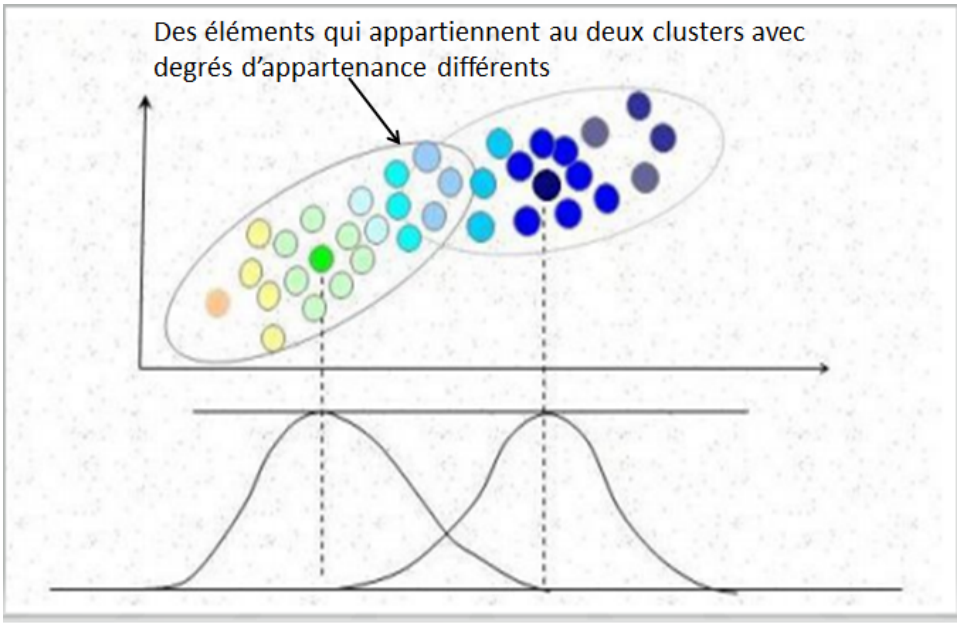

FIGURE 4: Exemple de classification floue.

L'algorithme des FCM effectue une optimisation itérative en évaluant de façon approximative les minimums d'une fonction d'erreur. Il existe toute une famille de fonctions d'erreur associées à cet algorithme qui se distinguent par des valeurs différentes prises par un paramètre réglable, m, appelé indice de flou "fuzzy index" et qui détermine le degré de flou de la partition obtenue. Le FCM est un cas particulier d'algorithmes basés sur la minimisation d'un critère ou d'une fonction objectif. on suppose que $X = \{x_1, ...x_n\}$, est un ensemble de données avec n point ,X est divisée en C classes c_i,Dans ce cas, les items x_j ne sont plus assignés à une unique classe, mais à plusieurs par l'intermédiaire de degrés d'appartenanceμ_{ij} du vecteur x_j à la classe c_i.Le but de algorithme est alors non seulement de calculer les centres des classes comme K-means mais aussi l'ensemble des degrés d'appartenance des vecteurs aux classes. Si μ_{ij} est le degré d'appartenance de x_j à la classe c_i, la matrice $U = [\mu_{ij}]$ est appelée matrice de C-partition floue si et seulement si elle satisfait :

$$\begin{cases} (\forall i \in \{1...C\})(\forall j \in \{1...n\})\, \mu_{ij} \in [0,1] \\ 0 < \sum_{j=1}^{n} \mu_{ij} < n \\ \forall j \in \{1...n\} \sum_{i=1}^{C} \mu_{ij} = 1 \end{cases}$$

le problème de partition de X en C classe floues pouvait être formulé comme la minimisation de la fonction J (C, U, X) définie par :

$$J(C,U,X) = \sum_{i=1}^{C} \sum_{j=1}^{n} \mu_{ij}^{m} ||x_j - c_i||^2 \qquad (2)$$

Avec m>1 est un paramètre contrôlant le degré de flou de la partition résultante.

$||x_j - c_i||^2$ est une distance du vecteur x_j au centre c_i . la méthode FCM, soufrent de trois défauts :

- ❄ Le nombre de clusters doit être fourni à l'avance.
- ❄ Chaque région est caractérisée par un centre, et les degrés d'appartenance sont calculés en faisant intervenir la distance euclidienne d'où une forme nécessairement hypersphérique.

❄ Moindre sensibilité au bruit : Comme l'a souligné Krishnapuram [19], si un point x est à distance égale de deux centres de clusters, le degré d'appartenance de x à chacun de ces deux clusters sera le même et sa valeur d'appartenance est égale à 0,5 Le problème de cette affectation est que les «points bruit» (qui peuvent très éloignés mais équidistants de deux centres de clusters) sont traités identiques aux points proches des centres du clusters mais en réalité, ces points de bruit sont supposés très faible et même leurs appartenance vaut zéro, à l'un ou l'autre cluster.

Pour améliorer la dernière faiblesse de la FCM , Li et al. [18] ont appliqué le concept de moyenne pondérée à la FCM pour créer un nouveau Algorithme de clustering de type FCM, appelé le poids pondéré flou Algorithme C-Means (WFCM).

3.8 Weighted Fuzzy C-Means

FCM utilise la distance euclidienne commune qui suppose que chaque entité a la même importance. Cette hypothèse affecte sérieusement les performances de FCM, de sorte que les clusters obtenus ne sont pas logiquement satisfaisants. Étant donné que dans la plupart des problèmes du monde réel, les caractéristiques ne sont pas également pertinentes. nous proposons l'algorithme C Moyen Flou Pondéré(WFCM) pour résoudre les problèmes susmentionnés. L'algorithme WFCM effectue le clustering des données en minimisant la fonction de coût suivante :

$$J(W,U,C,X) = \sum_{i=1}^{C} \sum_{j=1}^{n} w_j \mu_{ij}^{m} ||x_j - c_i||^2 \tag{3}$$

Où :
X est l'ensemble des données.
$U = [\mu_{ij}]$ est la matrice de C-partition floue.
C le nombre des partitions.
West la matrice des poids de X.
cette équation est une modification de L'équation 3 , permettant d'attribuer un poids w_j à chaque échantillon de données x_j.

4 Le Fuzzy Système de Détection d'Intrusion Proposé

Il existe trois étapes fonctionnelles dans la mise en œuvre de notre fuzzy système proposé, comme le montre Figure 5.

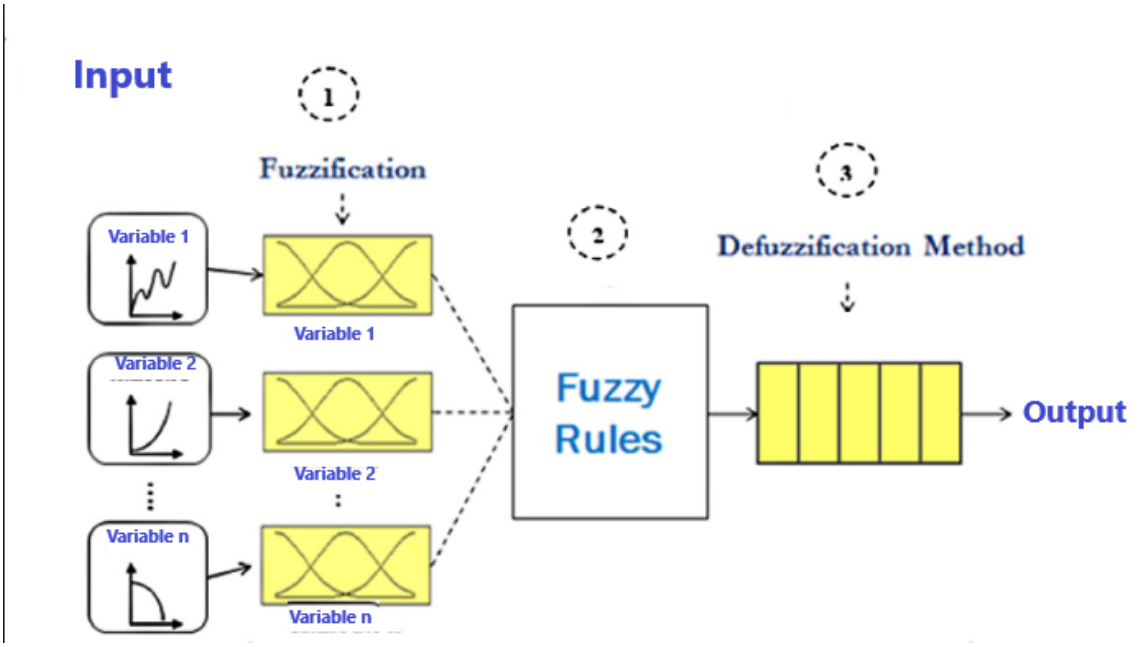

FIGURE 5: Les étapes Fonctionnelles de notre Fuzzy système

4.1 La Fuzzification des entrées

l'étape de Fuzzification consiste à définir les ensembles flous pour les variables(features) d'entrées et de sortie , pour chacune de ces variables ,on doit connaitre à priori son intervalle de définition , le nombre de d'ensembles flous et les fonction d'appartenance [14].
Dans ce travail on propose de fixer le nombre d'ensembles flous pour chaque variable d'entrée en trois (on prend trois) et toutes les fonctions d'appartenance sont considérées comme ayant une forme trapézoïdale. Pour induire les paramètres de trapèzes on propose d'utiliser l'algorithme de clustering pondéré Fuzzy C-Means (WFCM) [19], qui est une méthode de clustering rigide, permettant à chaque variable d'entrées d'appartenir à plus d'un cluster avec différents degrés d'appartenance, et un poids déterminé [20].
En fait , Le WFCM introduit des poids dans chacune des dimensions de données pour définir l'importance de chaque variable. Dans l'analyse pratique des clusters, le poids de chaque variable n'est pas connu à l'avance, la matrice des poids W doit être optimisée au cours du processus de classification.
Étant donnée un ensemble de transaction $T = \{t_1, t_2, t_3 ... t_m\}$ l'ensemble de toutes les transactions devant être utilisées pour entraînement et $X = x_1, x_2, ...x_n$ ensemble des variables (Features).
À la première étape, les valeurs $t_j[x_k]$ (pour chaque j tel que $1 <= j <= m$, $1 < k < n$) sont regroupées en C clusters ,à l'aide de l'algorithme WFCM. Pour chaque cluster généré,nous exploitons les degrés d'appartenance de variables pour s'adapter à un trapèze.Le trapèze représentera la fonction d'appartenance correspondant à ce cluster.

Algorithme de WFCM

- Fixer les paramètres : C le nombre de clusters(ici on fixé C=3), ε le seuil représentant l'erreur et m le degré de flous, généralement pris entre $[1.5, 3]$.
- Initialiser la matrice de degrés d'appartenances U et la matrice des poids W par des valeurs aléatoires dans l'intervalle $[0, 1]$.
  $t = 0$(epoch 0)

- Calculer les centres, pour $i = 1$ jusqu'à C :

$$c_i(t) = \frac{\sum_{j=1}^{n} w_j \mu_{ij}^m(t) x_j}{\sum_{j=1}^{n} w_j \mu_{ij}^m(t)} \tag{4}$$

- tant que $||U(t+1) - U(t)|| > \varepsilon$ faire :
- mettre à jour la matrice des appartenance U : pour tout $i = 1...C$, pour tout $j = 1...n$

$$U_{ij}(t+1) = \frac{1}{\sum_{k=1}^{C} (\frac{||x_k - c_i(t)||}{||x_k - c_j(t)||})^{2(m-1)}} \tag{5}$$

- mettre à jour la matrice des poids W : pour tout $i = 1...C$, pour tout $j = 1...n$

$$w_i = [\sum_{k=1}^{C} (\frac{\sum_{j=1}^{n} \mu_{jk}(x_j - c_i)}{\mu_{ik}(x_j - v_i)})^m]^{-1} \tag{6}$$

  $t = t + 1$
- fin de tant que

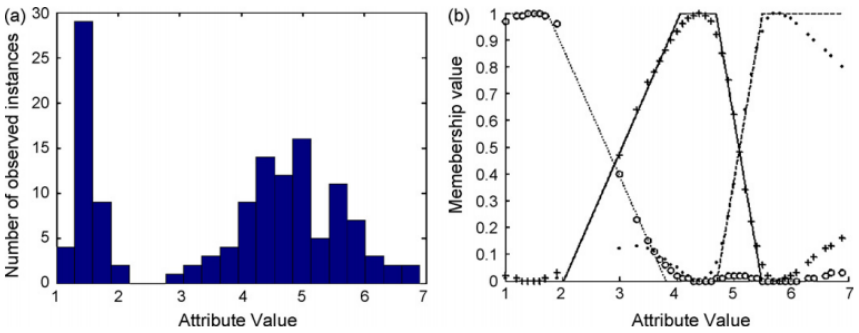

FIGURE 6: Des clusters basés sur l'algorithme WFCM .

La Figure. 6 partie**(a)** montre la distribution des valeurs d'un feature, et la partie **(b)** illustre la procédure de fonction d'appartenance trapézoïdale adaptée au même feature.
les degrés d'appartenance des features qu premier, au deuxième et au troisièmes clusters, sont respectivement tracés par les symboles *cercle, plus et point*.

4.2 Les Règles d'inférence Flou

c'est l'étape ou l'on établit les règles floues qui permettaient d'aboutir à une sortie en fonction des valeurs des features d'entrées.
Chaque règle est composée de prémisses liées par les opérateur **ET , OU** et donne lieu à une implication par l'opérateur **Alors**.
L'agrégation de ces règles est une opération qui doit aboutir à une seule valeur de la variable de sortie après défuzzification. Il existe plusieurs méthodes pour la génération des Règles floues , comme les méthodes génétiques , les réseaux de neurones, etc. Dans ce travail ,nous proposons l'utilisation de *Matlab* pour créer la base des règles floues. La création des règles floues se fait à l'aide de la commande *newfis* qui accepte jusqu'à 7 arguments.
La syntaxe générale de la méthode *newfis* avec sept argument est la suivante :

$Sys\_flou = newfis('nom\_syst','type','ET\_method, OU\_method, imp\_method, agg\_method, defuzz\_method)$
où :

- *nom_syst* : nom dy système floue.

- *type* : Mamdani ou Sugeno(le type du système diffère au niveau de la définition de la sortie soit une valeur constante indépendante des variables d'entrées (type Mamdani) soit une combinaison linéaire de celles-ciSugeno) .

- *ET_method* : méthode utilisé pour L'opérateur ET (min,prod, etc).

- *OU_method* : méthode utilisé pour L'opérateur OU (max, prod, etc).

- *imp_method* : méthode d'implication(min , prod ,etc).

- *agg_method* : méthode d'agrégation des règles (max, sum,etc).

- *defuzz_method* :méthode de Défuzzification (centroide, etc) .

Pour un système flou possédant n variables d'entrée et m variables de sortie , l'ensemble de règles floues est défini par une matrice possédant autant de lignes que d'ensembles floues pour chacun des variables d'entrées et $(m+n+2)$colonnes. le nombre des règles augmente de manière exponentielle avec l'augmentation du nombre des variables et les ensembles floues correspondants .
D'autre part , La sémantique d'une règle est satisfait s'il y'a une quantité suffisante de variables qui contribuent avec leurs votes,et la somme de ces votes est supérieure à un seuil spécifié par le concepteur du système.
Si une règle est intéressante, elle devrait avoir une signification suffisante. Étant donné $T = \{t_1, t_2, ...t_m\}$ $X = \{x_1, x_2, ...x_n\}$ un sous ensemble de variables , $A = \{a_1, a_2, ...a_p\}$ est un ensemble des ensembles floues de $a_j \in F(x_j)$,$F(x_j)$ est l'ensemble des ensembles floue dex_j ,et $\{\mu_1, \mu_2, ...\mu_p\}$ est l'ensemble des fonctions d'appartenance de A, tel que μ_j représente la fonction d'appartenance de a_j. La formule suivante est utilisée pour calculer le facteur de signification de (X, A) :

$$S_{(X,A)} = \frac{\sum_{t_k \in T} \prod_{x_j \in X} \{\alpha_{a_j}(t_k[x_j])\}}{||T||} \tag{7}$$

où :
||T|| représente le nombre de transaction dans T et

$$\alpha_{a_i}(t_k[x_j]) = \begin{cases} \mu_i(x_j) & si \quad \mu_i(x_j) >= \omega \\ 0 & otherwise \end{cases}$$

telle que ω est un seuil utilisé pour éviter les faibles degré d'appartenance. l'opérateur $\prod$ est utilisé pour calculer le vote de chaque variable.

Dans ce travail, nous proposons une nouvelle formule de signifiance d'une règle, en intégrant les poids des variables dans la formule initiale (équation 7) , de manière à ce que le vote d'une variable ne soit pas calculé uniquement en fonction de sa valeur d'appartenance , mais également il contribuera par son poids pour estimer l'importance d'une règle floue.
la nouvelle formule proposée est sous la forme :

$$S_{(X,A)} = \frac{\sum_{t_k \in T} \prod_{x_j \in X} \omega_j \{\alpha_{a_i}(t_k[x_j])\}}{||T||} \tag{8}$$

Où ω_j est le poids de la variable j calculé par l'algorithme WFCM à l'étape précédente et :

$$\alpha_{a_j}(t_k[x_j]) = \begin{cases} \mu_i(x_j) \quad si \quad \mu_i(x_j) >= \omega \quad et \quad \omega_j >= \theta \\ 0 \quad otherwise \end{cases}$$

comme la valeur d'appartenance ,Le poids d'un attribut ne doit pas être inférieur à un seuil spécifié par l'utilisateur, de sorte que l'attribut ayant une faible influence (poids) ne sera pas pris en compte. Le résultat final de cette étape est un ensemble de règles qui reflètent le degré de support fourni par les transactions, ce qui nous aide à estimer l'intérêt de chaque règle floue générée.

4.3 La Défuzzification

Dans la Défuzzification on réalise l'opération inverse du Fuzzification, à savoir obtenir une valeur numérique de la sortie à partir des variables et des règles issues du système d'inférence . Pour cela , il existe trois grande méthodes :la méthode de maximum , la méthode de la moyen pondérée, et la méthode de centroide.
on peut utiliser la boite à outils "Fuzzy logic toolbox " de *Matlab* pour choisir l'une de ces fonctions ou choisir d'autres.

5 Phase de Detection

Pour la phase de test, une nouvelle transaction sera transmise au système de logique floue conçu dans la section 4 pour trouver le score flou. Tout d'abord, on applique le fuzzifier aux variables d'entrée de la nouvelle transaction ,pour convertir les variable numérique en variable linguistique en utilisant l'algorithme WFCM. La sortie du fuzzifier est transmise au moteur d'inférence qui compare cette entrée particulière avec la base de règles. La sortie du moteur d'inférence est l'une des valeurs linguistiques du jeu suivant faible and Fort et ensuite, il est converti par le défuzzifier en tant que valeur scalaire. La valeur nette obtenue à partir du flou le moteur d'inférence varie entre 0 et 1, où «0» indique que les données sont complètement normales , «1»spécifie les données complètement attaquées, pendant que une valeur compris entre 0 et 1 indique le degré de menace ou degré de déviation du comportement normale.

6 Extension de L'IDS Learning Proposé

L'extension de L'IDS Learning(Chapitre 3) se fait par l'ajout le système de detection basé sur la logique floue , la nouvelle architecture est montré dans la Figure .7

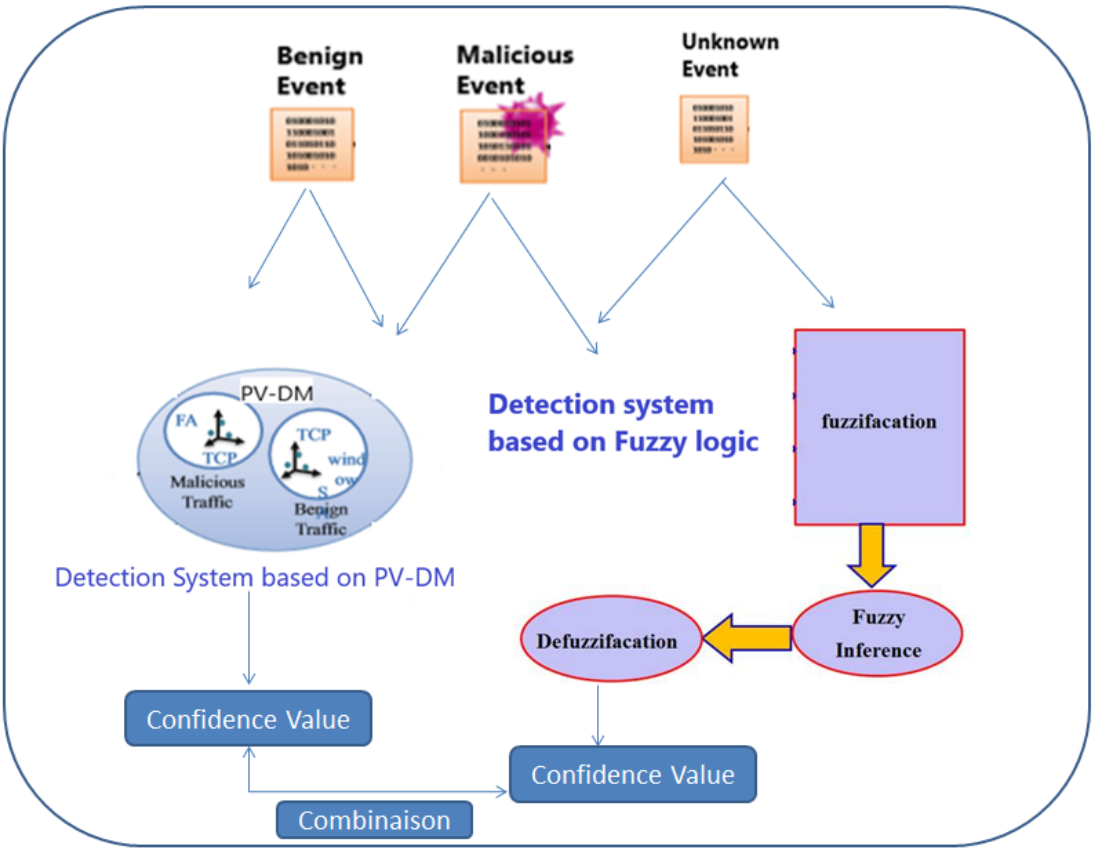

FIGURE 7: L'architecture de l'extension de L'IDS Learning

Dans ce nouveau système de detection , les transactions sont analysé par les deux systèmes de détection, celui basé sur le PV-DM et celui basé sur le fuzzy logic , les valeurs de confidence retournés par les deux systèmes sont combinés en calculant leur moyen , ce moyen est le degré du "menace" de cette transaction ou le degré de déviation du comportement normal .

7 Conclusion

Dans le présent travail, nous avons proposé une extension de notre IDS basé sur l'algorithme PV-DM . la nouvelle composante ajoutée est un système de detection basée sur la logique floue pour faire face aux comportements malveillants au sein d'un système. nous proposons d'utiliser l'algorithme WFCM qui introduisant des pondérations dans chacune des dimensions de données.nous avons proposé une nouvelle formule de signification d'une règle floue qui permet à chaque feature de voter non seulement par son valeur d'appartenance mais aussi par son poids.
Cette approche vise à réduire les taux de fausses alertes, qui constituent un problème très grave pour un IDS, et à garantir la flexibilité des systèmes de détection d'intrusion dans un environnement imprécis et incertain.
le futur travail consiste à implémenter cette approche en utilisant le langage de programmation *python* et *Matalb* ce qui nous permettra de valider notre travail et de produire des résultats expérimentaux pertinents.

Bibliographie

[1] CSI/FBI. Computer crime and security survey. computer security inst, 2004. <http ://www.issa-sac.org/.

[2] Cole E, Krutz R, and Conley JW. Network security bible. Wiley Publishing, 2005.

[3] Bivens A, EmbrechtsM, Palagiri C, Smith R, and Szymanski B. Network-based intrusion detection using neural networks. Proc Artif Neural Netw Eng, page 527–35, 2002.

[4] G. Kim et al. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Syst. Appl. 41, page 1690–1700, 2014.

[5] Barbara D, Couto J, Jajodia S, Popyack L, and Wu N. ADAM. detecting intrusions by data mining. Proc 2nd annual IEEE workshop inf assur security, page 11–6, 2001.

[6] www.snort.org. Snort network intrusion detection system,jun2006. www.snort.org.

[7] Ujwala Ravaleetal. Feature selection based hybrid anomaly intrusion detection system using k means and rbf kernel function. Proceeding of International Conference on Advanced Computing Technologies and Applications, page 428–435, jan 2015.

[8] Arman Tajbakhsh et al. Intrusion detection using fuzzy association rules. Applied Soft Computing 9, page 462–469, 2009.

[9] Chan Man Kuok, Ada Fu, and Man Hon Wong. Mining fuzzy association rules in databases. ACM SIGMOD Record, pages 41 – 46, March 1998.

[10] Cohen W.W. Fast effective rule induction. Proceedings of the 12th International Conference on Machine Learning, page 115–123, July 1995.

[11] Lee W, S. Stolfo, and K. Mok. A data mining framework for building intrusion detection models. Proceedings of the IEEE Symposium on Security and Privacy, page 120–132, jan 1999.

[12] A.Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. Proceedings of the Third SIAM Conference on Data Mining, May 2003.

[13] The snort tool.URL https ://www.snort.org/

[14] H. S. Vaccaro, G. E. Liepins, Detection of anomalous computer session activity, in : Security and Privacy, 1989. Proceedings., 1989 IEEE Symposium on, IEEE, 1989, pp. 280–289.

[15] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, J. Srivastava, A comparative study of anomaly detection schemes in network intrusion detection, in Proceedings of the Third SIAM Conference on Data Mining, May 2003.

[16] K. Chadha, S. Jain, Hybrid genetic fuzzy rule based inference engine to detect intrusion in networks, in : Intelligent Distributed Computing,Springer, 2015, pp. 185–198.

[17] wikipedia. fr.wikipedia.org, Avril 2018. https ://fr.wikipedia.org/wiki/Th % C3%A9orie_ des_ ensembles.

[18] wiki. wikipedia.org,Avril2018.https ://fr.wikipedia.org/wiki/Univers_du_discours.

[19] R. Krishnapuram and J. M. Keller. A possibilistic approach to clustering. Le beau journal, page 98–110, May 1993.

[20] C.H Li et al. A novel fuzzy weighted c-means method for image classi[U+FB01]cation. Int.J.Fuzzy Syst, page 168–173, Sep 2008.

[21] Chih-Cheng Hung et al. A new weighted fuzzy c-means clustering algorithm for remotely sensed image classi[U+FB01]cation. IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, JUNE 2011.

[22] Tombini.E et al. A serial combination of anomaly and misuse idses applied to http traf[U+FB01]c. 20th Annual Computer Security Applications Conference, pages 428–437, jan 2004.

Chapitre 6

Conclusions et Perspectives

La croissance rapide de diverses technologies émergentes, telles que les capteurs, les appareils connectés, les appareils ménagers intelligents, les villes intelligentes, les supports de communication 5G, les smart phones, le cloud mobile, les applications de santé, le multimédia et la réalité virtuelle, et les automobiles autonomes contribuent à l'énorme accumulation de données en temps réel qui circulent dans le réseau. Cette croissance envoie des signaux alarmants concernant la sécurité du réseau. Des différents types de cyber-attaques ont été identifiés dans l'infrastructure des systèmes informatiques. Par exemple, spamming, réseaux de zombies, phishing,les logiciels malveillants ,les attaques pour les sites Web etc.
Malgré le développement significatif de la sécurité des réseaux,les solutions existantes sont incapables de défendre complètement les réseaux informatiques contre les menaces malveillantes. Les techniques de sécurité traditionnelles telles que les pare-feu, l'authentification des utilisateurs et le cryptage des données ne sont pas suffisamment capables de protéger totalement la sécurité du réseau en raison du développement rapide de techniques d'intrusion.Par conséquent, des nouveaux mécanismes de défense comme les outils de Big Data analytique , Deep learning et les IDS qui sont suggérés pour faciliter la sécurité du système .

Contributions

Nos travaux comportent plusieurs contributions .Nous avons proposé une nouvelle approche pour le filtrage du spam. Ce filtre met l'accent sur la nature complémentaire des informations fournies par le contexte global et locale des termes les plus pertinents d'un courrier électronique . Notre filtre utilise le modèle de Deep Learning PV-DM et la méthode de Big Data Analytique TF-IDF pour avoir une représentation vectorielle pertinente pour chaque message . Les résultats expérimentaux ont confirmé clairement que les classificateurs entrainés avec les vecteurs générés par notre modele obtiennent les meilleurs résultats et surpassent les modèles basés sur PV-DM et BoW. De plus, ils prouvent que la méthode proposée est plus résistante aux différences de système linguistique et de cohésion des messages.

En dépit des performances de notre filtre anti spam, ce filtre ne peut pas détecter un courrier électronique avec un contenu légitime et un URL malveillant, ce qui nous a motivé d'étendre notre filtre pour la detection des phishing emails qui eux aussi sont des classés comme parmi les top sources de failles de sécurité dans le cyber espace.

Ainsi , nous avons proposé une architecture de détection de phishing augmentant la sécurité de la messagerie, par un double filtrage : l'un pour l'analyse du contenu textuel d'email et le seconde pour l'analyse des URLs suspectes. Pour analyser le texte Email, nous proposons d'utiliser notre filtre anti-spam précédent .
Afin de créer un filtre robuste et résistant face tout changement partielle dans les données ,l'ensemble des données est converti en une version corrompue ,ensuite nous avons introduit un Denoi-

sing autoencoder(DAE) qui est entrainé pour reconstruire les données initiales à partir de la version corrompue de celle ci . Pour réduire l'espace des features , qui constitue un problème majeur dans le contexte de phishing , vu l'explosions des URLs Malveillant , on introduit un autoencoder (AE) qui va chercher une représentation compressée et pertinente pour les données. Ce modele présente de nombreux avantages pour la classification des courriers phishing :

- Le fait qu'elle exploite les informations disponibles sur les URL, ainsi que sur le contenu textuel des courriers électroniques.
- Elle peut apprendre la structure profonde des données et construit, àl'aide d'un Autoencoder non supervisé,les features pertinentes de l'URL suspecte.
- Il pourrait reconstruire les données même si elles sont corrompues.

les Résultats sont moyennes vu la petite dimension de la base de données qu'on a expérimenté, ce qui a une influence sur le processus d'apprentissage de L'Auto Encoder et le Denoising Auto Encoder qui Requiert une masse assez grande de données pour la phase d'entrainement.

Après l'élaboration des filtres précédents , nous avons proposé un nouveau système de détection d'intrusion qui utilise l'algorithme d'Information Mutuelle pour la sélection des features pertinents et le PV-DM pour la lecture des packets réseaux. le système proposé a apporté les contributions suivantes :

- Il a montré que seulement quelques features sont nécessaires pour détecter les trafics malveillants et obtenir de hautes performances. Cela entraîne à son tour moins de stockage nécessaire pour stocker les données et exécuter l'algorithme. De même, les résultats peuvent facilement être interprétés.
- Les performances de notre méthode, ont été obtenues dans un environnement totalement non supervisé, c'est-à-dire sans aucune connaissance préalable de ce qui constitue une attaque. Par conséquent, si les attaquants modifient les techniques ou les protocoles d'attaque, nos méthodes apprennent automatiquement les caractéristiques.

Par ailleurs, les attaques constituent un environnement imprécis, incertain et incomplet , et que le Fuzzy Logic ou la logique floue a la capacité de prendre des décisions rationnelles dans tel contexte , nous avons envisagé d'explorer les techniques de Fuzzy logic pour la détection des intrusions, ces techniques qui confèrent plus de flexibilité ce qui rend possible la prise en compte des imprécisions , des incertitudes et des imperfections des données et qui se rapprochent dans une certaine mesure de la flexibilité du raisonnement humain.

Par suite , Nous avons proposé un système hybride de détection d'intrusion, en utilisant le Fuzzy logique pour faire face aux comportements malveillants au sein d'un réseau. nous avons proposé d'utiliser l'algorithme weighted Fuzzy C-Mean (WFCM) qui introduit des pondérations dans chacune des dimensions de données.nous avons aussi proposé une nouvelle formule de signification d'une règle qui permet à chaque attribut de voter non seulement par sa valeur d'appartenance uniquement, mais aussi par son poids.Cette approche a visé à réduire les taux de fausses alertes, qui constituent un problème pour un IDS, et à garantir la flexibilité des systèmes de détection d'intrusion dans un environnement imprécis et incertain.

Perspectives

Le présent travail étudie le cyber Sécurité et propose des méthodes pour Filtrer , analyser et extraire les pertinents informations , ce qui aide à une meilleure compréhension des données, à la réduction des données, à la limitation de l'espace de stockage requis et à la réduction des coûts de traitement.Ce qui nous donnent un avantage précieux dans l'amélioration de la sécurité et de la qualité des performances de cyber Security .

Cependant, notre travail de recherche n'est pas terminé et se poursuivra. nous prévoyons faire le suivi de ce travail de plusieurs manières :

la première de nos perspectives implique l'implémentation du système de detection des intrusions basé sur le Fuzzy logic , et par suite tester toute l'architecture de detection basée sur PV-DM et Fuzzy logic.

Deuxièmement nous prévoyons aussi , l'exploration d'autres algorithmes Deep Learning tel que le Reccurent Neural Network, en Particulier LSTM , BI-LSTM qui eux aussi prennent en compte le contexte des événements. ce qui pourrait constituer un autre mécanisme pour la compréhension des Intrusions Malveillantes , et par suite le pouvoir d'identifier de nouvelles attaques.

Enfin , nous prévoyons l'élaboration d'un filtre global pour tout type de spam (texte , phishing , image ,etc), et l'exploration des algorithmes génétiques , qui sont des paradigmes informatique d'intelligence artificielle et qui vont grandement aidé à détecter les fraudes , les tentatives d'attaques,et à surveiller le système, par leurs capacité à évoluer et à apprendre automatiquement.

RESUME

Malgré le développement important de la sécurité des systèmes informatiques, les solutions existantes ne peuvent pas défendre complètement les systèmes informatiques contre les menaces malveillantes. La plupart de ces attaques sont de petites variantes des Cyber-attaques connues et répertoriés, mais même des mécanismes avancés tels que les machines Learning rencontrent des difficultés pour détecter ces petites attaques mutantes au fil du temps.

Le succès de Deep Learning (DL) dans divers domaines a suscité l'intérêt de l'utiliser pour la detection des attaques, ce qui pourrait constituer un mécanisme résilient face à ces petites mutations ou à des nouvelles attaques.

Dans cette thèse, nos travaux se focalisent sur la sécurité des systèmes informatiques, en particulier, Nous nous intéressons au filtrage du courrier électronique et à la detection des intrusions malveillantes.

Nous avons proposé des nouvelles méthodes de filtrages des emails spam et phishing en utilisant des outils Deep Learning comme le model Neural Paragraph Vector-Distributed Memory (PV-DM), L'AutoEncoder (AE) et le Denoising AutoEncoder(DAE).

Ces filtres anti spam et anti phishing nous ont inspiré par la suite l'élaboration d'un système de détection des intrusions Malveillantes en se basant sur les Modèles PV-DM et Fuzzy Logic.

ABSTRACT

We are at the dawn of the era of "Big Data". The growing volume of information generated by businesses, the rise of social media and the Internet are fueling exponential data growth. Companies based on technologies such as Microsoft, Yahoo, Amazon and Google have kept data in Exabyte or even more. Most of the information cannot be managed by traditional tools. On the other hand, the increasing dependence on computer systems, offers a large attack surface to attackers, having all kinds of motivations: financial theft, data theft, disruption, damage to the reputation or simply to have "epic lulz". The result is a landscape of threats ranging from highly sophisticated attacks to opportunistic cyber-criminality.

Despite the significant development of network security, existing solutions cannot fully defend computer networks against malicious threats. Moreover, most of these attacks are small variants of the cyber-attacks known until now. This indicates that even advanced mechanisms such as traditional machine learning systems have difficulty detecting these small mutant attacks over time. In addition, the success of Deep Learning (DL) in various areas of Big Data has spurred many interests in the field of cyber security.

The use of DL for attack detection in cyber space could be a resilient mechanism against small changes or new attacks due to its ability to extract high-level features.

In this thesis, we propose Methods of filtering unwanted emails, commonly known as SPAM using Neural Networks and Deep Learning tools: Auto Encoder and Denoising Auto Encoder. Moreover, we have adopted this filter as an intrusion detection system that identifies suspicious actions based on the local and global context of the attack. Thus, we have proposed an Intrusion Detection system that differs from the above by integrating fuzzy logic concepts such as membership functions and fuzzy sets as well as fuzzy association rules.

Mots Clés :
Big Data, Deep Learning, IDS, Cyber Security, Fuzzy Logic.

Année Universitaire : 2018/2019