IMPLÉMENTATION ET ANALYSE DES RÉSULTATS

5.1 Introduction

L'implémentation de notre modèle de détection d'intrusions consiste à trouver tout d'abord une architecture optimale qui donne la possibilité de détecter les attaques avec un taux de réussite élevé.

Afin d'obtenir les meilleures performances possibles, nous avons effectué plusieurs expérimentations avec des paramètres du réseau différents. Dans ce travail, nous nous intéressons beaucoup plus aux poids qui relient les arcs du réseau, donc nous essayons de trouver ses valeurs optimales qui peuvent perfectionner notre modèle. Les résultats de ces expérimentations ainsi que les valeurs des paramètres utilisées seront détaillés dans ce chapitre.

5.2 Environnement de programmation

Nous avons choisi l'environnement de programmation Eclipse Jee 2019 pour Windows afin d'implémenter les deux modules d'apprentissage et de test de notre modèle de détection d'intrusions. Le choix du langage java a été guidé par les avantages offerts par la programmation orientée objet d'une façon générale.

Les caractéristiques techniques de la machine sur laquelle le modèle est implémenté et testé sont résumées dans le tableau suivant :

Composantes	*Valeurs*
Processeur	Intel®Core™ i5-4300M CPU
Vitesse	2.60 GHz
Mémoire	8.00 Go
Système d'exploitation	Windows 7 64 bits

Tableau 5.1 – Caractéristiques techniques de l'ordinateur utilisé pour l'implémentation

5.3 Test et résultats Expérimentaux

5.3.1 Paramètres de test

Afin d'obtenir la meilleure exactitude(taux de réussite) possible, nous avons effectué plusieurs tests en changeant les paramètres du réseau à chaque fois(taux d'apprentissage, nombre de couches cachées, nombre de neurones par couche et l'intervalle des poids et biais initiaux). Donc, on a fait le test pour 2 couches cachées, 3, 4, 5 et 6 et à chaque fois on a changé le nombre de neurones jusqu'à l'obtention de sa valeur optimale qui donne les meilleurs résultats.

Les valeurs des autres paramètres communs dans les différentes expérimentations soit pour le réseau de neurones ou bien pour le recuit simulé sont indiquées dans le tableau ci-dessous :

Paramètres d'apprentissage		*Paramètres d'optimisation*	
Paramètres	**Valeurs**	*Paramètres*	**Valeurs**
Taux d'apprentissage	0.3	Température initiale	1000
Fonction de transfert	$1/(1+e^{-x})$	Taux de refroidissement	0.997
Intervalle des poids et biais initiaux	[-5,5]	Critère d'arrêt	Lorsque la solution reste inchangée 10 fois successives

Tableau 5.2 – Paramètres d'apprentissage et d'optimisation

5.3.2 Résultats obtenus

Pour montrer l'impact de recuit simulé et recherche tabou sur les performances du réseau de neurone, nous avons effectué trois expérimentations pour les mêmes valeurs des paramètres d'apprentissage. Pour le premier cas, nous nous basons uniquement sur l'opération d'apprentissage pour calculer les valeurs optimales des poids et biais qui seront utilisées pour classifier

les connexions dans le système de détection d'intrusions. Dans le second cas, on utilise le recuit simulé pour trouver la solution optimale qui perfectionne les performances du réseau. Tandis que dans le dernier cas, on utilise la méthode de recherche tabou au lieu de recuit simulé.

Les résultats obtenus pour les trois expérimentations sont montrés dans les tableaux et les graphes suivants :

Premier cas : sans méthodes d'optimisation

- Mesures de performance :

Nbr couches cachées	*Nbr d'attributs*	*Erreur*	*Exactitude %*	*Rappel %*	*Précision %*	*TFA %*	*F_score %*
2	7	0.1	**80.88**	68.52	97.01	2.79	80.31
3		0.1	**78.05**	63.23	97.25	2.35	76.63
4		0.1	**80.22**	67.15	97.26	2.49	79.45
5		0.1	**80.69**	68.49	96.61	3.17	80.16
6		0.1	**80.96**	68.55	97.16	2.64	80.39

Tableau 5.3 – Évaluation des résultats obtenus sans méthodes d'optimisation

Dans ce cas, on constate que la valeur d'exactitude ne dépasse pas 81% et le rappel aussi ne dépasse pas 69%. D'où la nécessité de l'utilisation des méthodes d'optimisation pour améliorer ces performances.

- Matrice de confusion :

	Nombre neurones/couche	*Réels*	*Calculés*	
			Attaque	*Normale*
MLP à 2 couches	2	*Attaque*	8794	4039
		Normale	271	9440
MLP à 3 couches	5	*Attaque*	8115	4718
		Normale	229	9482
MLP à 4 couches	5	*Attaque*	8618	4215
		Normale	242	9469
MLP à 5 couches	5	*Attaque*	8790	4043
		Normale	308	9403
MLP à 6 couches	5	*Attaque*	8798	4035
		Normale	257	9454

Tableau 5.4 – Matrice de confusion de modèle MLP sans méthodes d'optimisation

À partir de cette matrice on peut calculer les métriques précédents en appliquant les formules citées dans la section 4.3.3.

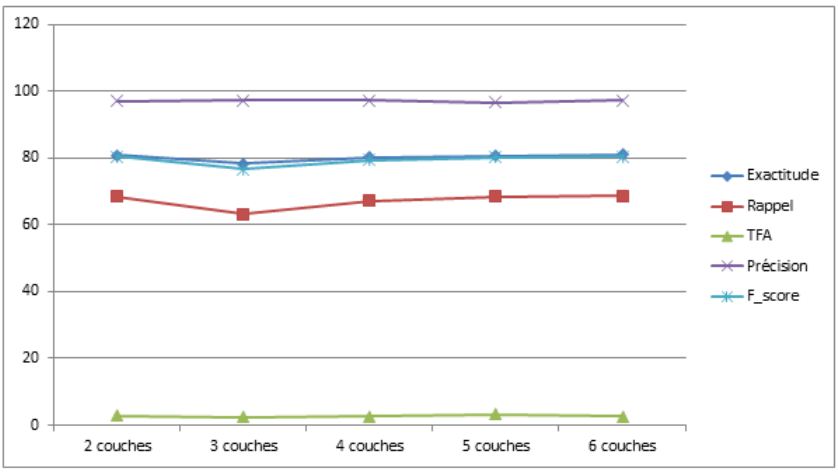

Figure 5.1 – Résultats obtenus sans méthodes d'optimisation

Cette figure montre clairement les résultats obtenus sans méthodes d'optimisation. Ces résultats nous ont incités à rechercher de nouvelles méthodes pour perfectionner notre modèle.

Deuxième cas : avec recherche tabou

- Mesures de performance :

Nbr couches cachées	*Nbr d'attributs*	*Erreur*	*Exactitude %*	*Rappel %*	*Précision %*	*TFA %*	*F_score %*
2	7	0.11	**85.04**	77.16	95.73	4.54	85.45
3		0.11	**84.94**	77.30	95.37	4.95	85.39
4		0.14	**85.96**	78.95	95.62	4.77	86.49
5		0.12	**85.94**	78.12	96.52	3.71	86.35
6		0.11	**86.10**	78.99	95.85	4.51	85.45

Tableau 5.5 – Évaluation des résultats obtenus avec recherche tabou

En utilisant la recherche tabou, les performances du modèle ont été mieux que celles obtenues sans méthodes d'optimisation. Donc, nous avons pu améliorer l'exactitude qui arrive jusqu'à 86,10% et le rappel qui atteint presque 79%.

- Matrice de confusion :

	Nombre neurones/couche	*Réels*	*Calculés*	
			Attaque	*Normale*
MLP à 2 couches	2	*Attaque*	9903	2930
		Normale	441	9270
MLP à 3 couches	5	*Attaque*	9920	2913
		Normale	481	9230
MLP à 4 couches	5	*Attaque*	10132	2701
		Normale	464	9247
MLP à 5 couches	5	*Attaque*	10026	2807
		Normale	361	9350
MLP à 6 couches	5	*Attaque*	10138	2695
		Normale	438	9273

Tableau 5.6 – Matrice de confusion de modèle MLP avec recherche tabou

D'après les résultats cités dans cette matrice, on peut montrer l'efficacité de recherche tabou qui nous a donné la possibilité de détecter plus d'attaque. Par exemple, pour un réseau de neurones de 5 couches cachées, le nombre des attaques bien détectées est 10026 alors que dans le premier cas, il a été 8790 seulement. Donc, nous avons pu détecter plus de 1200 nouvelles attaques.

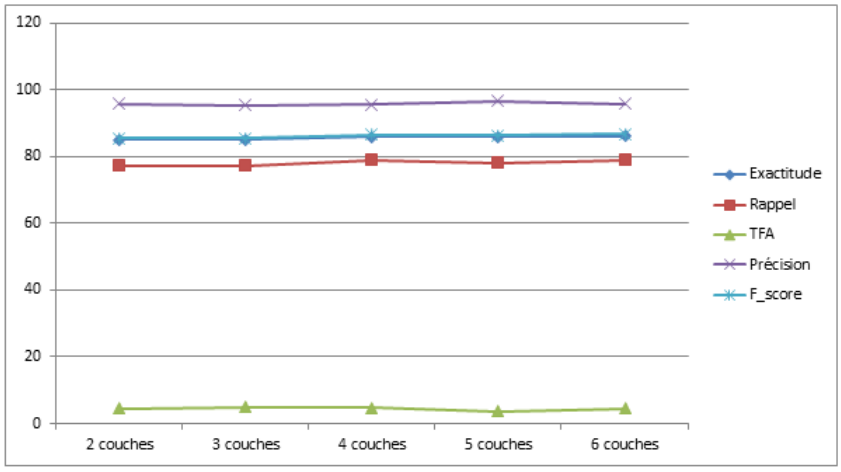

Figure 5.2 – Résultats obtenus avec recherche tabou

Cette figure montre l'amélioration apportée par l'algorithme de recherche tabou.

Troisième cas : avec recuit simulé

- Mesures de performance :

Nbr couches cachées	*Nbr d'attributs*	*Erreur*	*Exactitude %*	*Rappel %*	*Précision %*	*TEA %*	*F_score %*
2	7	0.11	**84.77**	76.27	96.18	3.99	85.08
3		0.11	**84.94**	77.02	95.68	4.59	85.34
4		0.14	**86.04**	79.24	95.45	4.98	86.60
5		0.11	**86.18**	78.57	96.49	3.76	86.62
6		0.12	**86.12**	78.60	96.33	3.95	86.57

Tableau 5.7 – Évaluation des résultats obtenus avec recuit simulé

Sachant que cette technique, recuit simulé, fournit des bonnes solutions pour la majorité des problèmes d'optimisation, dans ce travail, elle nous a donné aussi la meilleure exactitude qui égale à 86,18% pour un réseau de neurones de 5 couches cachées.

- Matrice de confusion :

	Nombre neurones/couche	*Réels*	*Calculés*	
			Attaque	*Normale*
MLP à 2 couches	2	*Attaque*	9789	3044
		Normale	388	9323
MLP à 3 couches	5	*Attaque*	9885	2948
		Normale	446	9265
MLP à 4 couches	5	*Attaque*	10170	2663
		Normale	484	9227
MLP à 5 couches	5	*Attaque*	10084	2749
		Normale	366	9345
MLP à 6 couches	5	*Attaque*	10088	2745
		Normale	384	9327

Tableau 5.8 – Matrice de confusion de modèle MLP avec recuit simulé

Si on prend le même exemple précédent, un réseau de neurones de 5 couches cachées, on trouve que le nombre des attaques bien détectées est 10084. Alors, on a plus de 50 nouvelles attaques ont été détectées par rapport à la méthode de recherche tabou et plus de 1250 par rapport à celle qui n'utilise aucune méthode d'optimisation.

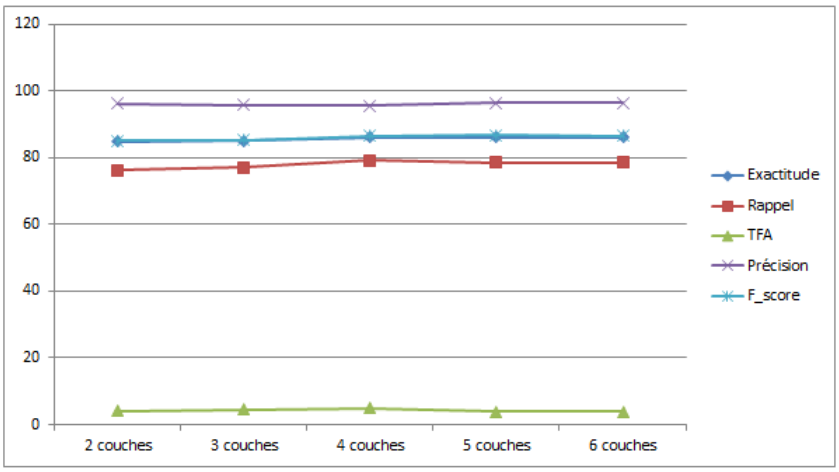

Figure 5.3 – Résultats obtenus avec recuit simulé

Cette figure donne une illustration des résultats obtenus en utilisant le recuit simulé qui nous a donné les meilleures résultats par rapport aux autres approches implémentées dans ce travail.

5.3.3 Analyse et comparaison des résultats

D'après les résultats présentés dans la section précédente, on montre que les performances de notre modèle de détection d'intrusions sont acceptables. Il peut détecter les intrusions y compris les nouvelles attaques avec un taux de réussite arrivant jusqu'à 86,18% et un taux de fausses alertes ne dépassant pas 5%. De plus, on peut montrer l'importance de l'algorithme recuit simulé et la méthode de recherche tabou qui peuvent améliorer les performances du modèle par l'optimisation de l'ensemble des poids et biais du réseau. Cette amélioration est montrée dans les tableaux suivants :

Nbr de couches cachées	*Exactitude*			*Rappel*			*Précision*		
	Sans RS	*Avec RS*	*différence*	*Sans RS*	*Avec RS*	*différence*	*Sans RS*	*Avec RS*	*différence*
2	80.88	84.77	**+3.89**	68.52	76.27	+7.75	97.01	96.18	-0.83
3	78.05	84.94	**+6.89**	63.23	77.02	+13.79	97.25	95.68	-1.57
4	80.22	86.04	**+5.82**	67.15	79.24	+12.09	97.26	95.45	-1.81
5	80.69	86.18	**+5.49**	68.49	78.57	+10.08	96.61	96.49	-0.12
6	80.96	86.12	**+5.16**	68.55	78.60	+10.05	97.16	96.33	-0.83

Tableau 5.9 – Comparaison des performances avec et sans recuit simulé

Nbr de couches cachées	*Exactitude*			*Rappel*			*Précision*		
	Sans RT	*Avec RT*	*différence*	*Sans RT*	*Avec RT*	*différence*	*Sans RT*	*Avec RT*	*différence*
2	80.88	85.04	**+4.16**	68.52	77.16	+8.64	97.01	95.73	-1.28
3	78.05	84.94	**+6.89**	63.23	77.30	+14.07	97.25	95.37	-1.88
4	80.22	85.96	**+5.74**	67.15	78.95	+11.8	97.26	95.62	-1.64
5	80.69	85.94	**+5.25**	68.49	78.12	+9.63	96.61	96.52	-0.09
6	80.96	86.10	**+5.14**	68.55	78.99	+10.44	97.16	95.85	-1.31

Tableau 5.10 – Comparaison des performances avec et sans recherche tabou

Selon les différentes expérimentations effectuées on peut constater l'impact du nombre de couches cachées sur les résultats obtenus. Dans notre cas, les meilleures performances calculées sont celles du réseau de neurones optimisé par le recuit simulé et qui contient 5 couches cachées.

Pour conclure, l'histogramme suivant montre clairement les améliorations que le recuit simulé et la recherche tabou apportent au réseau de neurones :

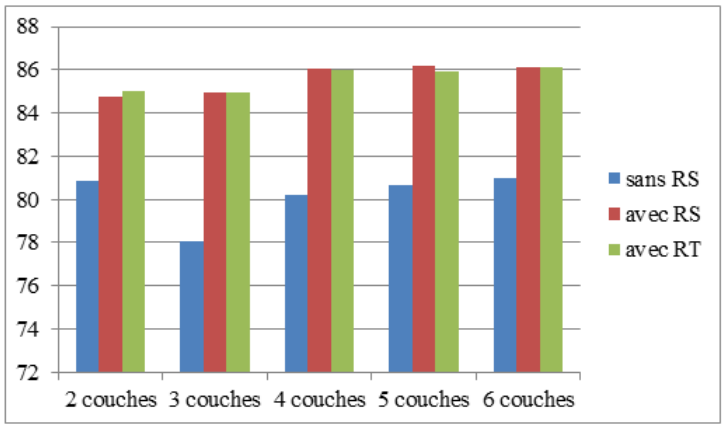

Figure 5.4 – Comparaison des taux de réussite calculés avec et sans méthodes d'optimisation

5.4 Conclusion

Dans ce dernier chapitre, nous avons présenté les résultats obtenus à partir de plusieurs expérimentations que nous avons effectué pour arriver au meilleur taux de réussite possible et les améliorations effectuées grâce aux algorithmes de recuit simulé et recherche tabou qui ont été utilisé comme des techniques d'optimisation sur les poids et les biais du MLP.

Les résultats obtenus ont montré l'efficacité des systèmes de détection d'intrusions basés sur les réseaux de neurones, et plus précisément le perceptron multi-couches, surtout lorsqu'il est combiné avec le recuit simulé ou la recherche tabou qui ont montré leur efficacité dans la résolution des problèmes NP-difficiles.

CONCLUSION GÉNÉRALE

Aujourd'hui, les attaques informatiques représentent un risque réel qui menace les systèmes informatiques et les réseaux des entreprises, ce qui nous a conduit à essayer, dans ce travail, de développer un modèle de sécurité capable de confronter cette menace en détectant toute tentative malveillante soit connue préalablement ou bien récente. Pour accomplir ce but, on a utilisé les réseaux de neurones et plus précisément le perceptron multi-couches, vu que c'est le modèle le plus adapté aux données non linéairement séparables, et cela pour classifier les connexions TCP/IP en deux catégories : normale ou attaque en se basant sur le benchmark NSL-KDD.

En effet, le choix des paramètres du réseau de neurones a une grande influence sur les performances de ces derniers. C'est pour cette raison que nous avons utilisé deux techniques d'optimisation qui sont le recuit simulé et la recherche tabou pour choisir les meilleurs valeurs des poids et biais du réseau qui nous donnent le meilleur taux de réussite possible. Tandis que le choix du nombre de couches, nombre de neurones par couche et le taux d'apprentissage a été fait manuellement en changeant ces valeurs dans chaque expérimentation jusqu'à l'obtention du modèle le plus performant.

Dans ce travail nous avons effectué une étude comparative entre trois modèles : celui qui se base uniquement sur l'algorithme du gradient, celui qui utilise le recuit simulé et enfin celui qui utilise la recherche tabou et cela pour trouver l'ensemble des poids et biais optimaux. Les résultats obtenus montrent l'influence positif des algorithmes d'optimisation sur les performances de réseau de neurones.

Enfin, puisque la recherche scientifique n'a pas des limites et malgré que nous avons obtenu des bons résultats, il existe des améliorations possibles pour perfectionner ce modèle telles que l'utilisation des métaheuristiques pour choisir les autres paramètres du réseau et aussi la réalisation d'une classification multi-classes qui donne de plus le type d'attaque détectée.

ANNEXE A

LES ATTRIBUTS DE LA BASE NSL-KDD

Les détails des attributs sont répertoriés dans le tableau suivant :[69]

N°	Nom de l'attribut	Description
01	duration	Durée de la connexion(nb de secondes)
02	protocol_type	Type du protocol : TCP, UDP ou ICMP
03	service	Service du réseau sollicité sur l'hôte de destination(ftp, http,...etc)
04	flag	Statut de la connexion(REJ, RSTO,...etc)
05	src_ bytes	Nbr d'octets envoyés de la source vers la destination
06	dst_ bytes	Nbr d'octets envoyés de la destination vers la source
07	land	Vaut 1 si la connexion est de/vers le même hôte/port, 0 sinon
08	wrong_ fragment	Nbr de fragments erronés
09	urgent	Nbr de paquets urgents
10	hot	Nbr d'indicateurs hot
11	num_ failed_ logins	Nbr de logins échoués
12	logged_ in	Vaut 1 si login réussi, 0 sinon
13	num_ compromised	Nbr de cas de compromission (compromised condition)
14	root_ shell	Vaut 1 si un shell root est obtenu, 0 sinon
15	su_ attempted	Vaut 1 si une commande super utilisateur est tentée, 0 sinon
16	num_ root	Nbr d'accès en mode root
17	num_ file_ creations	Nbr d'opérations en création de fichiers
18	num_ shells	Nbr de shells lancés
19	num_ access_ files	Nbr d'opérations d'accès aux fichiers de contrôle
20	num_ outbound_ cmds	Nbr de commandes non autorisées dans les sessions ftp
21	is_ host_ login	Vaut 1 si le login fait partie de la list hot, 0 sinon

N°	Nom de l'attribut	Description
22	is_ guest_ login	Vaut 1 si le login fait partie de la list guest, 0 sinon
23	count	Nbr de connexions pour le même hôte
24	srv_ count	Nbr de connexions pour le même service
25	serror_ rate	% de connexions pour le même hôte ayant l'erreur SYN
26	srv_ serror_ rate	% de connexions pour le même service ayant l'erreur SYN
27	rerror_ rate	% de connexions pour le même hôte ayant l'erreur REJ
28	srv_ rerror_ rate	% de connexions pour le même service ayant l'erreur REJ
29	same_ srv _rate	% de connexions pour le même hôte utilisant le même service
30	diff_ srv_ rate	% de connexions pour le même hôte utilisant différents services
31	srv_ diff_ host_ rate	% de connexions pour le même service utilisant différents hôtes
32	dst_ host_ count	Nbr de connexions pour le même hôte
33	dst_ host_ srv_ count	Nbr de connexions pour le même hôte utilisant le même service
34	dst_ host_ same_ srv_ rate	% de connexions pour le même hôte utilisant le même service
35	dst_ host_ diff_ srv_ rate	% de connexions pour le même hôte utilisant différents services
36	dst_host_same_src_port_rate	% de connexions pour le même hôte ayant le port src
37	dst_ host_ srv_ diff_ host_ rate	% de connexions pour le même hôte en provenance de différents hotes
38	dst_ host_ serror_ rate	% de connexions pour le même hôte ayant l'erreur SYN
39	dst_host_srv_serror_rate	% de connexions pour le même hôte et service ayant l'erreur SYN
40	dst_ host_ rerror_ rate	% de connexions pour le même hôte ayant l'erreur REJ
41	dst_ host_ srv_ rerror_ rate	% de connexions pour le même hôte et service ayant l'erreur REJ

Tableau A.1 – Liste des attributs de la base NSL-KDD

BIBLIOGRAPHIE

[1] B. Guttman and E. A. Roback, *An introduction to computer security : the NIST handbook*. Diane Publishing, 1995.

[2] W. Stallings, *Network security essentials : applications and standards*. Pearson, 2016.

[3] O. Lopez and F. Picard, *Cyber-assurance : nouveaux modèles pour quantifier l'impact économique des risques numériques*. No. 3, Association d'économie financière, 2019.

[4] J.-O. Gerphagnon, M. P. de Albuquerque, and M. P. de Albuquerque, "Attaques informatique," *CBPF-NT-007/00,Centre brésilien de recherche physique, Rio de Janeiro – RJ – Brazil*, 2004.

[5] S. Specht and R. Lee, "Taxonomies of distributed denial of service networks, attacks, tools and countermeasures," *CEL2003-03, Princeton University, Princeton, NJ, USA*, 2003.

[6] M. S. Hoque, M. Mukit, M. Bikas, A. Naser, *et al.*, "An implementation of intrusion detection system using genetic algorithm," *arXiv preprint arXiv :1204.1336*, 2012.

[7] S. Paliwal and R. Gupta, "Denial-of-service, probing & remote to user (r2l) attack detection using genetic algorithm," *International Journal of Computer Applications*, vol. 60, no. 19, pp. 57–62, 2012.

[8] R. Akimana, *Introduction à la sécurité informatique*. African Virtual University, 2018.

[9] R. Deal, *Cisco router firewall security*. Cisco Press, 2004.

[10] K. Salah, K. Sattar, Z. Baig, M. Sqalli, and P. Calyam, "Resiliency of open-source firewalls against remote discovery of last-matching rules," in *Proceedings of the 2nd International Conference on Security of Information and Networks*, SIN '09, (New York, NY, USA), p. 186–192, Association for Computing Machinery, 2009.

[11] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating computer intrusion detection systems : A survey of common practices," *ACM Comput. Surv.*, vol. 48, Sept. 2015.

[12] P. Biondi, "Architecture expérimentale pour la détection d'intrusions dans un système informatique," *Article de recherche,(Avril-Sptembre 2001)*, 2001.

[13] S. Mignault, *L'audit de sécurité et la protection des organisations*. Université de Montréal, 2009.

[14] G. Hiet, *Détection d'intrusions paramétrée par la politique de sécurité grâce au contrôle collaboratif des flux d'informations au sein du système d'exploitation et des applications : mise en œuvre sous Linux pour les programmes Java*. Theses, Université Rennes 1, Dec. 2008.

[15] N. Dagorn, *Détection et prévention d'intrusion : présentation et limites*. Laboratoire Lorrain de Recherche en Informatique et ses Applications(LORIA), 2006. https ://hal.inria.fr/inria-00084202.

[16] C. Frédéric, "Ids : Intrusion detection systems," *http ://www-igm.univ-mlv.fr/ dr/XPOSE2004/IDS/IDSSnort.html*, 2005.

[17] K. TABIA, "Développement de mécanismes de coopération entre algorithmes d'apprentissage automatique/classification dans un environnement incertain," *Mémoir de magistère, Université Mouloud Mammeri de Tizi ouzou*, 2005.

[18] H. Debar, M. Dacier, and A. Wespi, "A revised taxonomy for intrusion-detection systems," in *Annales des télécommunications*, vol. 55, pp. 361–378, Springer, 2000.

[19] A. Phillip, "Porras and alfonso valdes "live traffic analysis of tcp/ip gateways"," in *Proceeding ISOC Symposium on Network and Distributed System Security, San Diego, CA, March1998*.

[20] R. Graham, "Faq : Network intrusion detection systems," *http ://www. robertgraham. com/pubs/network-intrusion-detection. html*, 2000.

[21] T. Stéphane, *Data mining et statistique décisionnelle : l'intelligence des données*. Editions Technip, 2012.

[22] N. Labroche, *Modelling of the chemical recognition system of ants for the unsupervised classification problem : application to web usage mining*. Theses, Université François Rabelais Tours, Dec. 2003.

[23] N. Monmarché, *Artificial ant based algorithms applied to clustering and optimization problems*. Theses, Université François Rabelais - Tours, Dec. 2000.

[24] G. Cleuziou, *A Clustering method for rules learning and information retrieval*. Theses, Université d'Orléans, Dec. 2004.

[25] Y. Fataicha, *Recherche d'information dans les images de documents*. PhD thesis, École de technologie supérieure - Montréal, 2005.

[26] N. B. Amor, S. Benferhat, and Z. Elouedi, "Réseaux bayésiens naïfs et arbres de décision dans les systèmes de détection d'intrusions," *Technique et Science Informatiques*, vol. 25, no. 2, pp. 167–196, 2006.

[27] R. Marée, *Classification automatique d'images par arbres de décision*. PhD thesis, University of Lìege-Electrical Engineering and Computer Science, 2005.

[28] A. DJEFFAL, *Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données*. PhD thesis, Université Mohamed Khider-Biskra, 2012.

[29] P. Wira, "Réseaux de neurones artificiels : architectures et applications," *Cours en ligne, Université de Haute-Alsace*, 2009.

[30] M. Y. Ammar, *Mise en œuvre de réseaux de neurones pour la modélisation de cinétiques réactionnelles en vue de la transposition batch/continu*. PhD thesis, Institut National Polytechnique de Toulouse, 2007.

[31] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks : A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[32] P. Borne, M. Benrejeb, and J. Haggège, *Les réseaux de neurones : présentation et applications*, vol. 15. Editions OPHRYS, 2007.

[33] A. Berkani and Y. Abdeessemed, "Métaheuristique hybride réseaux de neurones artificielspso du recuit simulé pour la commande d'un procédé industriel non-linéaire," *Mémoire de magistère en Électronique de l'université Batna*, 2013.

[34] G. Dreyfus *et al.*, "Les réseaux de neurones," *Mécanique industrielle et matériaux*, vol. 51, 1998.

[35] M. Parizeau, "Réseaux de neurones," *GIF-21140 et GIF-64326*, vol. 124, 2004.

[36] R. GHAYOULA, "Contribution à l'optimisation de la synthèse des antennes intelligentes par les réseaux de neurones," *Université de Tunis El Manar. Thèse de doctorat*, vol. 27, 2008.

[37] S. Gunadiz, *Algorithmes d'intellignece antificielle pour la classification d'attaquer réseau à partir de donnée TCP*. PhD thesis, Université de Boumerdès - M'hamed Bougara, 2011.

[38] S. Aissaoui, *Apprentissage automatique et sécurité des systèmes d'information. Application : un système de détection d'intrusion basé sur les séparateurs à vaste marg (svm)*. PhD thesis, Université d'Oran - Ahmed Ben Bella, 2008.

[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[40] A. Belgacem, *Classification des signaux EGC avec un système-multi-agent neuronale*. PhD thesis, Universite Abou Bekr Belkaid - Tlemcen, 2012.

[41] Y. HAMMOUCHE, *Comparaison de plusieurs méthodes pour la prédiction de la Charge Electrique Nationale*. PhD thesis, Université Badji Mokhtar - Annaba, 2009.

[42] C. Touzet, *LES RESEAUX DE NEURONES ARTIFICIELS, INTRODUCTION AU CONNEXIONNISME*. Collection de l'EERIE, EC2, 1992.

[43] M. R. Meireles, P. E. Almeida, and M. G. Simões, "A comprehensive review for industrial applicability of artificial neural networks," *IEEE transactions on industrial electronics*, vol. 50, no. 3, pp. 585–601, 2003.

[44] J.-K. Hao, P. Galinier, and M. Habib, "Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes," *Revue d'intelligence artificielle*, vol. 13, no. 2, pp. 283–324, 1999.

[45] E.-G. Talbi, *Metaheuristics : from design to implementation*, vol. 74. John Wiley & Sons, 2009.

[46] I. Boussaid, *Perfectionnement de métaheuristiques pour l'optimisation continue*. PhD thesis, Paris Est, 2013.

[47] J.-C. Boisson, *Modelling and Solving with cooperative metaheuristics : from atom to protein sequence*. Theses, Université Lille 1, Dec. 2008.

[48] L. Jourdan, *Métaheuristiques pour l'extraction de connaissances : Application à la génomique*. PhD thesis, Université des Sciences et Technologie de Lille-Lille I, 2003.

[49] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[50] W. Tfaili, "Conception d'un algorithme de colonie de fourmis pour l'optimisation continue dynamique," *Paris val of Marne university*, 2007.

[51] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[52] S. Kirkpatrick, "Optimization by simulated annealing : Quantitative studies," *Journal of statistical physics*, vol. 34, no. 5-6, pp. 975–986, 1984.

[53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[54] M. O'Keeffe and M. O. Cinnéide, "A stochastic approach to automated design improvement," in *Proceedings of the 2nd International Conference on Principles and Practice of Programming in Java*, PPPJ '03, (USA), p. 59–62, Computer Science Press, Inc., 2003.

[55] T. Sibalija, *Application of simulated annealing in process optimization : A review*, pp. 1–48. 01 2018.

[56] F. Glover, "Future paths for integer programming and links to ar tifi cial intelli g en ce," *Computers operations research*, vol. 13, no. 5, pp. 533–549, 1986.

[57] F. Glover, M. Laguna, and R. Marti, "Principles of tabu search," *Approximation algorithms and metaheuristics*, vol. 23, pp. 1–12, 2007.

[58] Y. Xu, *Metaheuristic approaches for QoS multicast routing problems*. PhD thesis, University of Nottingham Nottingham, 2011.

[59] R. K. Cunningham, R. P. Lippmann, D. J. Fried, S. L. Garfinkel, I. Graf, K. R. Kendall, S. E. Webster, D. Wyschogrod, and M. A. Zissman, "Evaluating intrusion detection systems without attacking your friends : The 1998 darpa intrusion detection evaluation," tech. rep., MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, 1999.

[60] S. Stolfo, W. Fan, W. Lee, *et al.*, "Kdd-cup-99 task description," *http ://KDD. ics. uci. edu/databases/kddcup99/task.html*, 1999.

[61] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, pp. 1–6, IEEE, 2009.

[62] N. Paulauskas and J. Auskalnis, "Analysis of data pre-processing influence on intrusion detection using nsl-kdd dataset," in *2017 open conference of electrical, electronic and information sciences (eStream)*, pp. 1–5, IEEE, 2017.

[63] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[64] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018.

[65] Y. Ding and Y. Zhai, "Intrusion detection system for nsl-kdd dataset using convolutional neural networks," in *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, CSAI '18, (New York, NY, USA), p. 81–85, Association for Computing Machinery, 2018.

[66] P. Aggarwal and S. K. Sharma, "Analysis of kdd dataset attributes-class wise for intrusion detection," *Procedia Computer Science*, vol. 57, pp. 842–851, 2015.

[67] A. DJEFFAL, *Cours Fouille de données avancée*. Université Mohamed Khider - Biskra, 2014-2015. www.abdelhamid-djeffal.net.

[68] M. LABONNE, A. OLIVEREAU, and D. ZEGHLACHE, "Automatisation du processus d'entraînement d'un ensemble d'algorithmes de machine learning optimisés pour la détection d'intrusion," 2018. cesar-conference.org.

[69] L. Dhanabal and S. Shantharajah, "A study on nsl-kdd dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446–452, 2015.

Résumé

Aujourd'hui, et à cause de l'évolution technologique et l'utilisation d'Internet à grande échelle, le fait de tout sécuriser devient une nécessité incontournable et un défi pour la plupart des entreprises. Et vu que les moyens traditionnels de sécurisation sont devenus insuffisants à cause de l'augmentation du nombre et types d'attaques informatiques qui apparaissent presque chaque jour, les chercheurs du domaine de la sécurité informatique s'occupent d'élaborer des outils de sécurité basés sur des notions de l'intelligence artificielle pour détecter les nouvelles attaques. Dans ce travail, on a réalisé un modèle de détection d'intrusions basé sur les réseaux de neurones multi-couches optimisés par le recuit simulé et la recherche tabou en utilisant le benchmark NSL-KDD pour générer et évaluer ce modèle.

Mots clés :intrusions,réseaux de neurones, intelligence artificiel, recuit simulé, NSL-KDD.

Abstract

Today, and due to technological developments and the use of the Internet on a large scale, making everything secure became an unavoidable necessity and a challenge for most companies. And since traditional means of security have become insufficient due to the increase in the number and types of computer attacks that appear almost every day, researchers in the field of computer security are busy developing security tools based on notions of artificial intelligence to detect new attacks. In this work, an intrusion detection model based on multi-layer neural networks optimized by simulated annealing and tabu search was produced using the NSL-KDD benchmark to generate and evaluate this model.

Keywords :intrusion, neural networks, artificial intelligence, simulated annealing, NSL-KDD.