People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Larbi Tebessi University - Tebessa
Faculty of Exact Sciences and Natural and Life Sciences
Department: Mathematics and Computer Science

End of study thesis

For obtaining the MASTER diploma

Field: Mathematics and Computer Science

Sector: IT

Option: Networks and information security

Novel method for computer worm detection

Presented by :

Ahmed Messai

In front of the jury:

Bendjenna Hakim	Professor	University Larbi Tebessi	President
Bouakez Fatima	MAA	University Larbi Tebessi	Examiner
Menassel Rafik	MCA	University Larbi Tebessi	Supervisor

Defence date: July 11, 2021


بسم الله الرحمن الرحيم

Abstraction

The spread of computer worms is a nightmare for cyber security officers, which was and still threatens information security, as well as networks infrastructures that contribute and play significant role in economic development and societies growth, and its protection depends on various systems and tools, including “Intrusion Detection Systems”, where its development has become an imperative due to the advancement of computer worm attack techniques and their methods of concealing themselves from been detected.

Among the many methods used by researchers in developing intrusion detection systems that are found in the literature, ‘artificial neural networks’ were the most used among the machine learning techniques, so they were chosen as a main focus in this study, which we relied on to develop a new model that can detect network anomalies.

The training stage of artificial neural networks is considered the basic stage in building the predictive model, there are different algorithms in the literature to get that done, including deterministic methods and stochastic ones, each has its pros and cons. to train our proposed model, we relied, within this study, on the improved 'tree-seed algorithm' which is a nature-inspired algorithm, it’s used for the first time to optimize a neural network model parameters for detecting network anomalies aiming to get a more accurate model.

The proposed model was trained with an improved dataset from its predecessor that was extracted from a simulation of the US Air Force network, and it was evaluated based on the test part, which contains types of attacks that the model has not previously trained on, the results obtained indicate good learning capabilities of the proposed model comparing to the results of two other models based on two stochastic algorithms, namely the 'genetic algorithm' and 'particle swarm optimization '.

Résumé

La propagation des vers informatiques est un cauchemar pour les responsables de la cyber sécurité, qui était et menace toujours la sécurité de l'information, ainsi que les infrastructures de réseaux qui contribuent et jouent un rôle important dans le développement économique et la croissance des sociétés, et sa protection dépend de divers systèmes et outils, y compris « Systèmes de détection d'intrusion », où son développement est devenu un impératif en raison de l'avancement des techniques d'attaque de vers informatiques et de leurs méthodes pour se dissimuler d'être détectés.

Parmi les nombreuses méthodes utilisées par les chercheurs pour développer des systèmes de détection d'intrusion que l'on trouve dans la littérature, les « réseaux de neurones artificiels » étaient les plus utilisés parmi les techniques d'apprentissage automatique, ils ont donc été choisis comme axe principal de cette étude, sur laquelle nous nous sommes appuyés pour développer un nouveau modèle capable de détecter les anomalies du réseau.

L'étape de formation des réseaux de neurones artificiels est considérée comme l'étape de base dans la construction du modèle prédictif, il existe différents algorithmes dans la littérature pour y parvenir, et y compris des méthodes déterministes et stochastiques, chacun a ses avantages et ses inconvénients. Pour entraîner notre modèle proposé, nous nous sommes appuyés, dans le cadre de cette étude, sur l'« algorithme des graines d'arbres » amélioré, qui est un algorithme inspiré de la nature, il est utilisé pour la première fois pour optimiser les paramètres d'un modèle de réseau neuronal pour détecter les anomalies du réseau dans le but d'obtenir un modèle plus précis.

Le modèle proposé a été entraîné avec un ensemble de données amélioré de son prédécesseur qui a été extrait d'une simulation du réseau de l'US Air Force, et il a été évalué sur la base de la partie test, qui contient des types d'attaques sur lesquels le modèle n'a pas encore été entraîné, les résultats obtenus indiquent de bonnes capacités d'apprentissage du modèle proposé par rapport aux résultats de deux autres modèles basés sur deux algorithmes stochastiques, à savoir « l'algorithme génétique » et « l'optimisation de la colonie de particules ».

ملخّـــص

يعد انتشار ديدان الكمبيوتر كابوس مسؤولي الأمن السيبراني الذي كان و لا يزال يهدد أمن المعلومات و كذا البنى التحتية للشبكات التي تساهم و بدور فعال في التنمية الاقتصادية و تطور المجتمعات ، و يُعتمد في حمايتها على نظم و وسائل شتى من بينها 'نظم كشف الاقتحامات' حيث أصبح تطوِيرها ضرورة حتمية لِمَا شوهد من تطور تقنيات هجمات ديدان الكمبيوتر و طرقها في إخفاء نفسها من الكشف.

و من بين الطرق العديدة المستعملة من قبل الباحثتين في تطوير 'نظم كشف الاقتحامات' و المتواجدة في مراجع الأدبيات المدروسة، كانت شبكات الأعصاب الاصطناعية التقنية الأكثر استعمالا من بين تقنيات التعلم الذاتي للكمبيوتر ، لذا اختيرت كمحور أساسي في هذه الدراسة و التي استندنا إليها لتطوير نموذج يمكنه التنبؤ بحالات الشذوذ في سيرورة الاتصال داخل الشبكة الذي يعبّر عن احتمالية وجود هجوم ضدها.

تعتبر مرحلة تدريب الشبكات العصبية الاصطناعية المرحلة الأساسية في بناء النموذج التوقعي، و تستخدم في ذلك خوارزميات مختلفة منها الرياضية الدقيقة و منها 'العشوائية' ولكل منها ايجابياتها و سلبياتها، في هذا السياق ، اعتمدنا ضمن هذه الدراسة على الخوارزمية المحسنة المسماة خوارزمية 'الشجرة و البذور' و هي إحدى الخوارزميات المستوحاة من الطبيعة التي تعتبر فئة فرعية من خوارزميات 'العشوائية '، يتم استخدامها لأول مرة لتحسين معلّمات نموذج شبكة عصبية لاكتشاف الانحرافات في الشبكة بهدف الحصول على نموذج أكثر دقة.

تمّ تدريب النموذج المقترح بمجموعة بيانات محسنة عن سابقتها المستخرجة من محاكاة لشبكة القوى الجوية الأمريكية، تم تقيم النموذج المقترح بناءا على الجزء المخصص للتقييم و الذي يحتوي على أنواع هجمات لم يسبق للنموذج أن تدرب عليها، و قد كانت النتائج المحصل عليها تشير إلى القدرات التعليمية الجيدة للنموذج المقترح بالمقارنة لنتائج نموذجين آخرين مبنيين على خوارزميتين عشوائيتين هما ' الخوارزمية الجينية' و 'خوارزمية تحسين صرب الجسيمات'.

Dedication

To my Mother

Who taught me about dreams and how to catch them

To my Father

For raising me to believe that anything is possible.

To my beloved brothers

For their enduring encouragement, physical and moral support

and making this work possible

Appreciation

I would like to express my deepest appreciation to my supervisor

Dr. Menassel Rafik *for his invaluable advices and undeniable help during the whole period of my study*

Thank you sir for been a good leader.

My sincere gratitude for all my teachers for all that they taught me
I'll have it stored safely in my mind and my heart.

These thanks would not be complete without a thought to all those that helped and encouraged me directly or indirectly during the elaboration of this modest work.

Content:

Figures List:

Tables List:

General introduction:

Nowadays economy and civilization lean on computer systems and networks which enable a variety of new possibilities such as online conferences, remote works , online social interactions and electronic commerce through its developments. On the other side those systems and networks are ideal targets for computer worms to attack.

One of the frequently asked questions when discussing computer worms is: what are differences between a worm and a virus? [Smith, et al., 2009] although both are considered to be malware, except that they differ on the way they propagate, worms generally have the functionality of self-propagate across networks by exploiting software vulnerabilities and not relying on users interactions to activate them, viruses rely on user actions to activates and transports them to a new host [Smith, et al., 2009]. However this difference can blur in case of mass-mailing worms that depend on user to activate them to self-propagate, this study will consider both types as worms and focus on those whom self-propagate across networks.

Maintaining systems and networks security against various types of computer worms can be a challenging task; to increase the security, diverse tools and systems has been developed. Intrusion Detection System is a popular one and a widely developed concept in literature.IDS used to monitor behaviours on a single machine or on a segment of a computer network to detect intrusion [Srinivasu, et al., 2012], IDS differ in their objectives and implementations techniques, one of the significant capabilities of an IDS, is that it's not only capable of detecting known attacks also new unknown attacks.

IDS can detect anomalies or unknown attacks by searching for unusual patterns implementing for that prediction techniques such as Artificial Neural Networks (ANN). Those are among the well known and strong techniques of machine learning.

ANN learns by getting inputs, process them and generate an output, learning can be supervised where the training data have an attached labels as a correct output, unlike unsupervised learning where learning data have no attached outputs on it. [Cinar, 2020]

Training ANN can be done with deterministic or stochastic methods, although deterministic methods are simple and fast, they have the problem of initial solution dependency, stochastic methods in the other hand start with random solutions and evolve them to produce better ones, even if they are time consuming they can get to better results.[Cinar, 2020].

Metaheuristic optimizers are subclass of stochastic methods [Cinar, 2020], its primary goal is to coordinate a learner to find a fitter solution to a given optimization problem. There are many existing metaheuristic algorithms such as genetic algorithms, particle swarm optimisation along others [Martins, et al., 2006]

Tree Seed Algorithms is a population-based iterative search algorithm developed By [Kiran, 2015] for solving continuous optimization problems, inspired by the relationship of the tree with its seeds, means by which trees spread to the space, those grow over time to become new trees, by considering the spreading space is the search space, the location of trees and seeds are possible solution for the optimisation problem, our aim in this study is to train an artificial neural network with improved tree-seed algorithm to develop a new model for anomaly detection.

This study is divided into three chapters, the first chapter start by presenting principal keys and concepts of computer worms such as the internal architecture and mechanisms by which it propagate, intrusion detection systems, their types and implementation techniques are discussed in the second part of it. Some of the related works to our thesis are presented in the beginning of the second chapter, follow that a dive into artificial networks and how can they be trained, exploring by that metaheuristic optimizers concepts along with a detailed study of the improved tress-seed algorithm, to wrap up with a presentation of NSL-KDD dataset, chapter three will be the practical side of our work, in which our proposed framework is explained to finish with results discussion.

Chapter I:

State of Art

Introduction

The traditional security techniques such as firewalls, user authentication and data encryption are not capable enough to fully maintain security due to sophisticated and diverse worms attacks, therefore, new defence line such as intrusion detection system are needed. An IDS is a software or hardware system that identifies intrusion on computer systems in order to maintain system security, to fully understand how IDS detect intrusions, it's necessary first to understand what worms are composed of and how they work.

The first part of this chapter presents the internal of computer worms, their propagation methods, and how they can infiltrate undetected, thereafter its detection techniques are presented in the second part.

1. Computer worms:

Malware in their general form designed to execute undesirable code in the infected machine, 'viruses' in the first hand can only spread through attaching itself to a file (like .EXE,.DOC, .XLS... etc)**,** and wait for the user to make a copy of that file to replicate somewhere else, worms in the other hand are 'self-replicating piece of software that is capable to execute without human interaction' [Kanani et al., August 2017], and those are the most dangerous threat on the integrity of information [Avinash, et al., 2012]:

Worms exploit a software vulnerabilities like the most known 'buffer overflow' to execute into memory [Smith, et al., 2009]making by that a possibility to execute a network code and propagate itself through networks.

The line between viruses and worms blurs with discussion of mass-mailing worms [Smith, et al., 2009], for that, this study considers any network-propagation malwares as a worm that can affect the behaviour of a given network. To dive into what a worm is, this section presents worms in depth.

1.1. Computer worm architecture:

Like any other software, computer worms have an internal architecture that contains some essential parts to run, and a few no-essential parts as sophistication functionalities, we may find in the worm the following components:

1.1.1. *Target Locator:*

The first and most important part of a worm is the component that search for new targets to spread on. Target locating is 'the process where worms find new hosts to infect and is a characteristic behaviour that worms exhibit' [Smith, et al., 2009], target locating mechanism depend on worm propagation environment and how the attacker implement it, there are a number of techniques by which a worm can discover new machines to infect [Weaver, et al. January 2003], a list of those techniques is given below:

i. Random scanning:

Using a pseudo-random number algorithm, the worm generates a random address and probe for it [Smith, et al., 2009], in this fashion propagation will be slowly due to multiple time scanning of the same address, CodeRed v2 and SQL slammer uses this type of scanning [Rajesh et al. , 2015]

ii. Permutation scanning:

To avoid scanning the same address more than once, worm coordinate between instances of infected host to permute the scanned address ranges [Smith & Matrawy, 2009], This allow the propagation to be much faster [Weaver,et al. January 2003]

iii. Localized scanning:

Localized scanning worm balances between local and global range scan to search for a vulnerable hosts [Rajesh et al., 2015]. The advantage of scanning in this manner is that once a host behind a firewall is infected, it can directly infect other hosts within the local network without passing through the firewall [Smith, et al., 2009].

iv. Hit-list scanning:

Attacker could target a predefined list of hosts statically or dynamically [TANG et al., May 2009] as described barfly below:

- *Static hit-list*: Before a worm is released, a static hit-list is created [TANG et al., May 2009] for example using information of BGP routing to attack a specific geographic zone.
- *Dynamical hit-list*: There are several methods to create dynamical hit-list; after infecting a machine worm search for communication topology information locally. [Weaver, et al. January 2003], uses web Search engines or network services that maintain lists of available servers [Szor, February 2005]

v. Passive scanning:

After infecting a machine, a worm wait passively for incoming and outgoing connections from a potential victims, and tries to infect them; CRClean implements this strategy [TANG et al., May 2009].

1.1.2. *Propagator:*

Propagator is an essential part for worm to work correctly, it's the mean by witch worm can infect other machines; it's responsible on sending a compatible worm version with the victim machine system [Szor, February, 2005], as the mechanics of how propagation occur is determined within this part. A worm can either actively spread itself from machine to machine, or it can be carried along as part of normal communication. [Weaver et al., January 2003], generally there are three distribution mechanisms:

i. Self-Carried:

During the initial communication with the victim, the worm fully transmitted itself [Smith et al., 2009]; the Code Red II worm was a self-carried worm [TANG et al. May 2009].

ii. Second channel:

Propagation occur on two phases, the worm send a small message in the first phase to execute in the infected machine, making a second connection to download and run the rest of the worm, the bluster worm implement this mechanism for its propagation [Smith, et al., 2009].

iii. Embedded: [Weaver et al., January 2003], [Smith, et al., 2009]

In order for a worm to propagate without appear as an anomalous behaviour, it attach itself a message or totally replace it during a normal communication with the target machine.

1.1.3. *Payload:*

Though it's not an essential part except that it's the most commune one, worm payload refer to the desired actions to take by a worm after infecting a machine. Endless alternatives of a propagator can be exists, limited only by the imagination of the author of the worm [Smith, et al., 2009].

1.1.4. *Self-Tracking:*

when the worm infect more than one host ,worm construct its own networks and send messages back and further between infected nodes[Ali Sulieman, et al., 2008] ,This component allows tracking of worm propagation, either for counting infected machine, or tracks the path of infection [Szor, February, 2005].

1.1.5. *Remote control and update:*

Even it's not an essential part for a worm to work correctly though a remote controlling module is one of the important components of a worm [Kanani, et al., August 2017], a remote trivial-to-use privileged backdoor was opened on machines infected with CodeRed II allows to execute arbitrary code. [Weaver et al., January 2003]

Another important feature is the updating component; advanced computer worms have the ability to update its components on an already-infected system. [Szor, February, 2005].

1.2. Activation mechanisms:

Activation mechanism is a major factor for detecting a worm, and significantly affects the speed of worm spread [Weaver et al., January 2003]; four methods by which worms are activated are mentioned in the above list:

1.2.1. *Human Activation:*

Viruses as mentioned above need a human interaction to execute and propagate, email worms and peer to peer worms such as Melissa virus and Nimda worm falls under this category [Smith, et al., 2009]. This kind of worm will be active if user executes the local copy of it. Usually, the worm involves some social engineering techniques to deceive the user [Pratama, et al., April 2012]

1.2.2. *Human activity-based activation:*

Unintended behaviour to activate a worm can lead to its execution, such as inserting a removable memory into usb or cd reader or executing a user login script [Smith, et al., 2009]

1.2.3. *Scheduled process activation:*

Scheduled process activated worms use unsecured legitimate automated process to activate; an automatically self-updated program can be a victim of an infected web server [Smith, et al., 2009].

1.2.4. *Self Activation:*

By exploiting vulnerability in an available services or used libraries the worm can activate itself, it attach itself to the exposed process, or use its privilege to execute another proportion of code. CodeRed worm uses this technique by exploiting a security flaw in IIS web servers [Weaver et al., January 2003].

1.3. Computer worm propagation environments:

Propagation environment is very important characteristic of computer worm to attack the vulnerable host efficiently. Based on what technology the worm can transfer itself; the literature defined 4 types of worms:

1.3.1. *Email Worms: [Szor, February 2005] [TANG et al., May 2009]..*

One of the biggest security threats are email worms, after activation and compromising the user's host, email worm search for a possible list of e-mails on the infected machine and

propagate through sending itself as an email attachment, there are several methods of target locating for this type:

- Address-Book Worms.
- File Parsing Attacks on the Disk.
- NNTP-Based E-Mail Collectors.
- E-Mail Address Harvesting on the Web.
- Mail Address Harvesting via ICQ.
- Monitoring User Access to SMTP and Newsgroups.

An explanation of those techniques can be found in [Szor, February 2005].

1.3.2. *Internet worms*

By scanning internet IP address ranges, internet worms can locate and exploit security flaws to propagate [TANG, et al., May 2009]. In general term internet worms propagate from a host to another over network, this principle could be applied over internet and/or LAN, to travel from a host to another worms use communication protocols, UDP transports data is a single way mode unlike TCP which transports it is a dual way mode and that make the propagation using UDP more likely to be faster than leaning On TCP mode for propagation, considering the best case connection situation [Kanani, et al., August 2017].

1.3.3. *Instant messaging worms:*

An instant Messaging worm (IM worm) is a malware that propagate over IM networks and systems, after compromising the victim host it scans for the online buddy list of the IM application and attempt to send either a file transfer request or an URL-embedded chat message, in both cases IM worm need an interaction of the receiver to activate it by executing the embedded file or by clicking on the embedded link. [Yan, et al., April 2008].

1.3.4. *P2P worms:*

P2P worms spread by copying itself in a shared folder of P2P network in order to propagate [Kanani et al., August 2017], due to the distributed architecture of P2P networks P2P worms are hard to control, therefore P2P clients are common targets to this type of worms [Szor, February, 2005].

1.4. Anti-Detection Techniques:

Evading detection systems and strategies is the best way for a worm to vastly spread in the infrastructure of its propagation environment, in attempt to do that, it must cover its track by one or more of the following anti-detection techniques:

1.4.1. *Slow Scanning:*

Gathering multiple pieces of similar network traffic before sounding an alarm is one of the employed strategies of detecting worm scanning, and for the reason of memory saving, uncommon packets are discarded; by slowing the rate of scanning, worm can evade detection and continue to spread [Smith, et al., 2009].

1.4.2. *Polymorphism:*

Most current intrusion detection systems and anti-virus are signature-based, for evading such systems polymorphic worms constantly changes its identifiable internal features with every instance with maintaining for its logic. Thus it will never match a fixed or predictable signature [Prahlad et al., 2005].

1.4.3. *Encrypting:*

To evade detection, encrypted worm encrypt itself with a random generated key, transmit that encrypted version along with the key and a short decryptor program. Polymorphism techniques can be applied on the decryptor to make the whole executable undetected by signature-based IDS (explained in the second part of this chapter) [Smith, et al., 2009].

1.4.4. *Blending/Mimicry:*

Blending attacks or mimicry worms aim to analyse and learn the network traffic before infecting hosts in order to simulate normal network profile, in attempts to pass through anomaly IDS (also explained in the second part of this chapter) without detection [Smith et al., 2009].

1.4.5. *Misleading signature generators: [Smith et al., 2009].*

Signature IDS try to generate a signature for the current behaviour and matching it with an existed one, To avoid detection, worms aim to deceive the IDS by causing it to generate useless signatures to avoid being detected, or by sending 'allergy attacks' which consist of sending a normal infection attempt along with a second (or third) 'fake' infection attempt to the target host, fake attempts are shaped such that they closely match the real infection attempt in many of the 'polymorphic' portion, but not in the invariant one, the IDS generate a signature for the common bytes in the polymorphic areas, and based on that signature it will fail to identify any other variation.

Split up large anomalous worm TCP packets into many small packets is another misleading technique; also overloading IDS with useless traffic such that it can no longer detect worms can lead to in undetected infection.

1.4.6. *Metamorphism:*

Worms can pass through security lines by replacing instructions with their alternatives; inserting no-ops ones, change the sequence of subroutines or even evolve itself [TANG et al., May 2009].

2. Intrusion detection systems:

According to [Khraisat, et al., 2019], any process that threatens information confidentiality, integrity or availability or lead to a temporary or lasting damage to an information system can be defined as intrusion. An IDS is a software or hardware system that identifies intrusion on computer systems in order maintain system security. This section discusses various types of IDSs along with few techniques of anomaly detection implementation.

2.1. Classification of IDS:

IDSs can be broadly categorized based on two factors: the source of data used to detect abnormal activities and the methods used to identify/matching intrusions, based on source of data it can be described as network-based or host-based IDSs. In term of methods used to identify intrusions it can be classified also into two groups Signature-based and anomaly-based IDSs [Khraisat et al., 2019].

2.1.1. *Network-based IDS:*

NIDS inspect and analysis sniffed network traffic from a network data sources such as packet capture and NetFlow... etc, so that it can monitor activities on the network [Khraisat, et al., 2019]. A single sniffer module can be placed to monitor network traffic, or multiple modules placed to monitor traffic only in each node of the network [Saeed et al., 2013].

Network-based IDS can detect worm propagation by analysing the content of a worm been transmitted over a network and matching packets containing the same sequence of bytes [Smith & Matrawy, 2009].

2.1.2. *Host-based IDS:*

HIDS are designed to inspect data and detect the execution of malicious code on a single system; its primary goal is to detect attacks that do not involve network traffic. [Gideon & Jiankun, APRIL 2014]. HIDS collect data from operating system logs, firewalls logs, application system audits, or database logs [Khraisat, et al., 2019], however this technique has some weakness such as the amount of useless data logged by the daemon programs, not to mention, the data collecting process is not a smooth process [Gideon & Jiankun, APRIL 2014].

Host-based IDS are able to detect the activation of worms since it occur in the host machine. Monitoring system calls, tracking or detect buffer overflows can detect self-activated worms [Smith & Matrawy, 2009].

2.1.3. *Signature-based IDS:*

Signature intrusion detection systems (SIDS) also called Knowledge-based Detection (Misuse Detection) detect recognized attacks by compare current activities against existing intrusion signatures database [Khraisat, et al., 2019]. Signatures can be generated automatically or man-mad, though an ideal signature should match only malicious behaviour and no legitimate ones [Smith & Matrawy, 2009].

The main advantage of SIDS is that they have low false positives as long as attacks are clearly defined in advance also are easy to use, however; they have number of weaknesses, first is that it requires prior knowledge of the intrusions behaviour, and those are very dependent on the operating system, version and application [Vinod & Om Prakash, 2012], beside signature detectors can detect known attacks but are unable to detect new attacks [Smith & Matrawy, 2009], A potential solution to those drawbacks would be to use an anomaly IDS techniques, which operate by outlining what is an acceptable behaviour rather than what is anomalous[Kharisat,et al,. 2019]

2.1.4. *Anomaly-based IDS:*

Anomaly detection generate a normal behaviour profile and aim to find samples in data, which do not conform to what is expected [Bhuyan et al., 2014], AIDS first create a baseline of normal activities for the protected system. theoretically, an intrusion alarm raised on any deviation from that baseline [Gideon & Jiankun, APRIL 2014].

Detecting segments of executable code, unusual byte frequencies, unexpected flags in packet headers, packet data being used to overwrite the return address within a program and infrequent sequences of bytes in network traffic are all examples of some current implemented methods of AIDS [Smith & Matrawy, 2009].

AIDS faces the problem of high false alarm rates due to the difficulties of creating a vigorous baseline, even though they are capable to identify new attacks [Gideon & Jiankun, APRIL 2014].

During worm scan, many unique IP addresses are reached in a short time period and many TCP RST packets indicating many failed connection attempts will be received, AIDS can detect those as anomalous behaviours and raises an intrusion alarm [Smith & Matrawy, 2009]. The next section present in more details implementing techniques for AIDS.

2.2. Techniques for implementing AIDS:

Classifying network anomaly detection methods and systems can be complicated due to the considerable overlap among techniques used in the different classes [Bhuyan,et al. 2014], [Khraisat, et al., 2019] classified those techniques into three main classes, statistical, knowledge and machine learning based techniques:

2.2.1. *Statistics-based techniques:*

Statistics-based IDS build a stochastic model to simulate normal behaviour profile, and then observe events with low probability to be generated from the assumed model and flag them as intrusions [Bhuyan, et al., 2014].

Statistical AIDS basically takes into consideration the statistical metrics such as the median, mean, mode and standard deviation of behaviour, statistical IDS normally use Univariate, multivariate or time series models [Khraisat et al., 2019].

- Univariate model: only one statistical measure of behaviours used to build the normal behaviour profile.
- Multivariate model: in order to understand the relationships between variables, two or more statistical measures are used.
- Time series model: through series of observations made over a certain time, time series model detect low probability occurring observations at that time [Khraisat et al., 2019].

Statistical approaches have a number of distinct advantages [Manasi et al. 2012]:

- They have the capability to detect zero-day attacks without a requiring previous knowledge of anomalous activities.
- They are also capable of detecting “low and slow” attacks so slow scanning of a worm will be useless.
- They don’t require entire sequence of a particular behaviour; rather they look for individual elements which may be part of an intrusion.

Along with those advantages, statistical methods have the following disadvantages [Manasi et al. 2012]:

- Training an accurate and efficient model can take days or weeks.
- Setting the threshold too high value will decrease the detection rate, while setting it too low value will produce high false positive rate and that presents a major problem with SA-IDS.
- Any changes in a user’s behaviours can results in unacceptable number of false alarms.

2.2.2. *Knowledge-based techniques:*

Knowledge based detection techniques can be used for both signature based IDS as well as anomaly based IDS, It accumulates the knowledge about specific attacks and system vulnerabilities [Manasi et al., 2012].

Knowledge based systems acquire knowledge by matching subsequent of current behaviour instances with pre-determined attack representations [Bhuyan, et al., 2014].

According to [Khraisat et al., 2019] Knowledge based detection technique can further be classified as: finite state machine approach (FSM), Description Language approach and expert systems approaches:

i. Finite state machine:

FSM are composed of a set of states, transitions, and activities. FSM represent normal behaviours, and transactions made based on noted variations in the input data, an attack can be expressed as any deviation from that FSM [Khraisat et al., 2019].

ii. The description language:

Description language defines the syntax of rules which can be used to describe exchanged messages within a network with a strict interaction between hosts [Studnia, et al., 2018] or to describe a defined attack, description languages such as N-grammars and UML could be used to build rules [Khraisat et al., 2019].

iii. Expert systems:

Attacks can be described by expert systems with a set of rules. Facts can be described as a translation of audit events adding to them a semantic signification to increases their abstraction level, based on rules and facts inference engine can comes to a conclusion [Manasi et al., 2012].

Reducing false-positive alarms is the main advantage of knowledge based techniques as the system has knowledge about all the normal behaviours [Khraisat et al., 2019]; however, it have some drawbacks and are given below: [Manasi, et al., 2012]

- A regular update is needed for those techniques to achieve high performances.
- Detailed analyse of vulnerabilities in order to maintenance the knowledge base is a time consuming task.
- Its generalization ability has some issue.

2.2.3. *Machine learning based techniques:*

Due to its significant role in anomaly detection, machine learning techniques have been applied to develop a good number of AIDS in recent decades [Bhuyan, et al., 2014]. Several Methods and techniques such as clustering, neural networks, association rules, decision trees, and nearest neighbour methods, have been applied to develop models that are capable to detect intrusions [Khraisat et al., 2019], this section presents some of well known ML techniques

i. Decision trees:

Along with its high performance advantage, decision trees are considered as a good classifier for large datasets what makes it a well known learning technique in the field of anomaly detection [Peddabachigari et al., 2004].

A decision tree is composed of three basic elements, a decision node specifying a test attributes, an edge or a branch corresponding to one of the test attribute outcomes, and a leaf also called answer node contains the class to which the object belongs [Ben Amor et al., 2004].

Including the mentioned above, decision trees offers a wide range of advantages, we list: [Peddabachigari et al., 2004]

- It constructs easily inspect and edit interpretable models.
- These models can also be used in the rule-based models with minimum processing.
- Its generalization ability has shown a great accuracy.

- The ability to detect new intrusions due to its generalization accuracy.

ii. Naïve Bayes:

Naive Bayes represented in the form of a directed acyclic graphs composed of unobserved root node (called parent), and a set of observed children nodes. In the context of the parent node, independence is strongly assumed among children nodes [Ben Amor et al., 2004] which make those techniques ease of use and efficient in calculation, to be the most widespread implemented models in IDS[Khraisat et al., 2019].

Naive Bayes are generally used for intrusion detection combined with statistical techniques. [Manasi, J.L, & R.N, 2012].

iii. Artificial Neural Network (ANN):

ANN is one of the most widely applied machine-learning methods to intrusion detection problem for their adaptability to environmental changes and their high accuracy of prediction [Saeed et al., 2013].

An artificial Neural Network transform a set of inputs to a set of searched outputs, through a set of simple processing units called neurons, and a collection of connection between them. Groups of neurons are divided into layers: input layer, output layer, and the layer between input and output called hidden layers. The connection between two units is weighted for determining how much one unit will affect others [Reddy, 2013]. The most frequent learning technique employed for training a supervised learning ANN is back propagation (BP) algorithm [Khraisat et al., 2019] expect that it often suffer from local minima, thus learning can become very time-consuming.

The ability to produce highly nonlinear models which capture complex relationships between inputs and desired outputs make the strength of ANNs [Khraisat et al., 2019].

iv. Fuzzy logic:

Based on fuzzy theory which use approximation rather than precise logic, fuzzy logic presents a simple artificial intelligence approach [Saeed et al., 2013], that arrive to a final conclusion upon incomplete, ambiguous data [Khraisat et al., 2019].

Fuzzy logic constructs more abstract and flexible patterns for intrusion detection, and thus greatly increases the robustness and adaptation ability of detection systems [Singh et al., 2011]. As security includes vagueness and forming a well identifiable baseline between normal and abnormal behaviours can be difficult, fuzzy logic presents a good classification technique for IDS [Khraisat et al., 2019].

v. Support Vector Machines:

Support vector machine is one of the most widely used machine learning algorithms for classification problems [Xu, et al., 2014], their solving classification problems with high dimensional feature space make them suitable for intrusion classification [Singh et al., 2011].

As a discriminative classifier SVM linearly classified training data by plotting it into a higher-dimensioned space using kernel functions [Khraisat, et al., 2019].

SVM are widely known for their generalization capability [Khraisat, et al., 2019] though they are time consuming when it comes to a large dataset [Bhuyan et al., 2014].

vi. Hidden Markov Model (HMM):

HMM is a statistical Markov model assuming a Markov process with unseen data is been being modelled [Khraisat, et al., 2019]. HMMs can be seen as a machine learning technique, in the sense that the training process automatically extracts the relevant statistical information from the training data [Khreich, et al., 2012].as well

HMM is a stochastic process determined by the two interrelated mechanisms [Khreich, et al., 2012]:

- A latent Markov chain with a finite number of states.
- Every state is associated with an observation probability.

A great number of intrusion detection systems has been seen in the literature build based on HMM, HMMs have been applied either to anomaly detection, to model normal patterns of behaviour, or in misuse detection, to model a predefined set of attacks [Khreich, et al., 2012].

Conclusion:

This chapter attempt to give the bigger picture for the anatomy of computer worms, how worms can behave, and different spreading techniques also how they can evade detection, it discussed different detection techniques from general point of view, and goes in depth onto implementation techniques of an anomaly based intrusion detection system with their advantages and limitations .

Although the diverse techniques to detect a spreading worm and new attacks, networks and systems security remains threatened with spreading of new generation of sophisticated undetectable worms, therefore a more accurate anomaly detection is needed, the next chapter presents techniques used in this study aiming to attempt this objective.

Chapter II:
Proposed Method

Chapter II Proposed Method

Introduction:

ANNs and meta-heuristic algorithms are a hot research areas in the field of security systems in general and in anomaly detection systems specially, optimising a training process of a feed-forward neural network (FFNN) was and still a most discussed, growing and alive idea in the literature, Therefore, this chapter presents techniques and methods of developing a neural network model for detecting network anomalies, starting with mentioning some previous related works, after that the concept of artificial neural networks will be clarified along with back propagation training process, following that, basic notations of metaheuristics algorithms will be stated with a detailed presentation of the one that interested us in this study, afterwards, an examination of a chosen dataset that will be the benchmark of our developed model will be given out.

1. Related works:

This section focus on works related to optimize set of weights and biases of artificial neural network class with nature-inspired optimizers for anomaly detection.

[Sheikhan & Jadidi, 2014] had developed an approach where a modified gravitational search algorithm is used to determine the optimum values of weights of a MLP employed to detect anomalous activities in high-speed networks, thus results of testing the trained model in this study showed a higher accuracy and lower alarm rate.

In [JANTAN, et al., 2017] an ANNs are being relied on to create a learning based filter to detect spam e-mails, the authors proposed an optimising algorithms to set the weights and biases of FFNN model to optimum using a new modified bat algorithm. Results show that the concerned model achieved a high generalization performance compared to many other neural network based meta-heuristics models.

[Ghanem, et al., 2018] develop an artificial neural network with increased precision in classifying malicious from harmless traffic in a network based on Artificial bee colony and particle swarm optimisation hybrid algorithms (ABC-PSO) to optimize the interconnection weights of the FFNN. Results from the trained model compared to RBF Network, Voted Perceptron, Simple Logistic and Multilayer Perceptron trained models, the comparison reveal that the developed model has higher accuracy and lower error rate.

[Ali, et al., 2018] developed a learning model for fast learning network (FLN) a Double Parallel Forward Neural Network based on particle swarm optimization, the model has been applied to classify network activities and detect anomaly, the result of comparing the model against a wide range of meta-heuristic algorithms for training extreme learning machine and FLN classifier disclose that the PSO-FLN has outperformed other learning approaches in detection accuracy.

Additionally [Jantan, et al., 2020] carried out a new hybrid algorithm between Artificial Bee Colony algorithm (ABC) and Monarch Butterfly optimization (MBO) used to train an ANN such a way that it selects the suitable biases and weights, mainly to detect intrusions in a network, experiment results clearly demonstrated that the proposed approach provided significant improvement compared to nine other optimisation algorithms.

As shows **Tab II-1** a synthetic study for the cited related works above optimizing parameters set of neural network with metaheuristics algorithms can result an accurate model therefore in this study and as mentioned before, in the objective of getting a more accurate neural network model, a new metaheuristic algorithm called ‘improved tree seed algorithm’ is used to optimize the collection of weights and biases for a multilayer perceptron (MLP) , next section go in depth in used materials and methods used to attain our objective.

1- Average results
2- Training measurement (testing not included)
3- Divided into 2 sub simples (one for train and one for test)
4- Dataset was divided on multiple subsets each time a proportion used as training data and the others for testing.

authors	neural network class	metaheuristic algorithm	fitnessfunction	training dataset	testing datasets	Testing results								classification type
						ACC%	AR	MR	ER	MSE	MEA	RMSE	REA	RRSE	
[sheikhan & jadidi, 2014]	MLP	modified gravitational search algorithm	MSE	university of twente network capturing [2] (flow-based)	university of twente network capturing [2] (flow-based)	97.76[1]	0.21	2.48	2.24	N/A	N/A	N/A	N/A	N/A	Multiclass
in [jantan, et al., 2017]	FFNN	modified bat algorithm	MSE	uk-2011 webspam	N/A	N/A	N/A	N/A	N/A	4.76 e-01[2]	N/A	N/A	N/A	N/A	Binary
				spambase	N/A	N/A	N/A	N/A	N/A	3.27e-01[2]	N/A	N/A	N/A	N/A	Binary
[Ghanem, et al., 2018]	MLP	HYBRID ABC AND PSO	MSE	KDDcup99[3]	KDDcup99[3]	N/A	N/A	N/A	N/A	N/A	0.0017	0.0159	0.3558%	4.9758%	Multiclass
[Ali, et al., 2018]	FLN	PSO	ACC	KDDcup99[3]	KDDcup99[3]	99.68	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Multiclass
[Jantan, et al., 2020]	MLP	Hybridization of ABC and MBO	MSE	KDDcup99[4]	KDDcup99[4]	87.19[1]	0,167[1]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Binary
				SCX 2012-12[4]	ISCX 2012-12[4]	98.65[1]	0.003[1]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Binary
				UNSW-NB15[4]	UNSW-NB15[4]	96.86[1]	0.04[1]	N/A	N/A	N/A	N/A	N/A	NN/A	N/A	Binary

Tab II-1: synthetic study for the cited related works

2. Material and methods

2.1. Artificial Neural Networks:

An ANN consists of a set of interconnected processing units, simulating the physiological human brain structures in functioning and components; because of that, natural neuron is disscussed brieflly in the next section.

2.1.1. *Biological neuron :*

the human neural network consist of billions of cells called neurons **Fig II-1** [Gallo, 2015] composed of a cell body called "Soma" which contains the "nucleus" or more over "Neuritis" and a "Dendrites" which serve as input channels that receive signals from other neurons and pass them over to the cell body which process and transfer those signals through an "Axon" ; Axon represents the output channel that carries signals away through the "synapse" to the dendrites of neighbouring neurons [Basheer, et al., 2000].

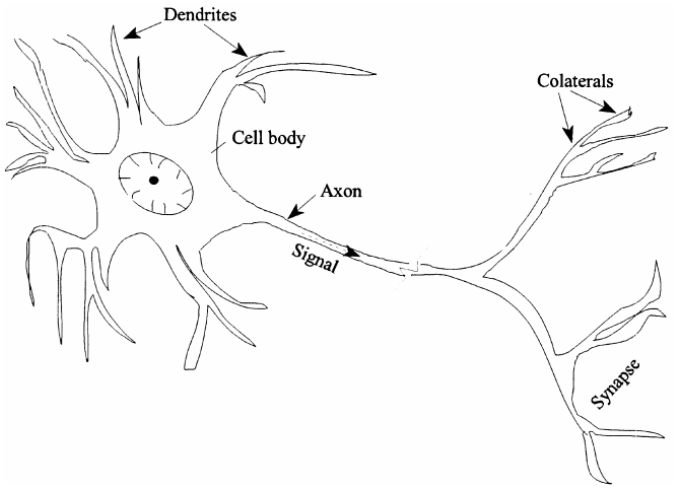

Fig II-1: Schematic of biological neuron [Basheer,et al., 2000]

The intellegent behavious can be described as a cell body recieves electrical signal eather from another neurons or from the external environment and based on the strength of the incoming signal a chimical component is released within axon through synapse weights toward the dendrites of the neighbouring neuron forcing them to generate a new electrical signal based on a threshold of the receiving neuron [Basheer, et al., 2000] [Gallo, 2015].

Imitating the componants and fonctionning of the natural neuron in a mathimatical model produces an artificial neuron.

2.1.2. ***Artificial neuron (Perceptron):*** [JANTAN, et al., 2017] [Basheer,et al., 2000]

Artificial neuron (also called perceptron) imitate the paradigm of the natural neuron by witch n inputs that either came from the environment or from other neurons represent axons and dendrites, each input provided with a weight (w) as the synapses in the biological unit, neuron calculate the summation ξ of its n inputs x with n weights.

Once the result obtained the neuron passes signals to the next layer of neurons (or the environment) only if ξ exceeds the neuron's threshold Θ (also called bias 'b'), neurons activation is calculated using a transfer function f (will be discussed later in this section) to produce an outputy_i.**Fig II-2** illustrate an artificial neuron and **Eq II-1** can describe the output of the ith artificial neuron.

$$y_i = f_i((\sum_{i=1}^{n} w_i x_i) + \Theta_i \quad \textbf{Eq II-1}$$

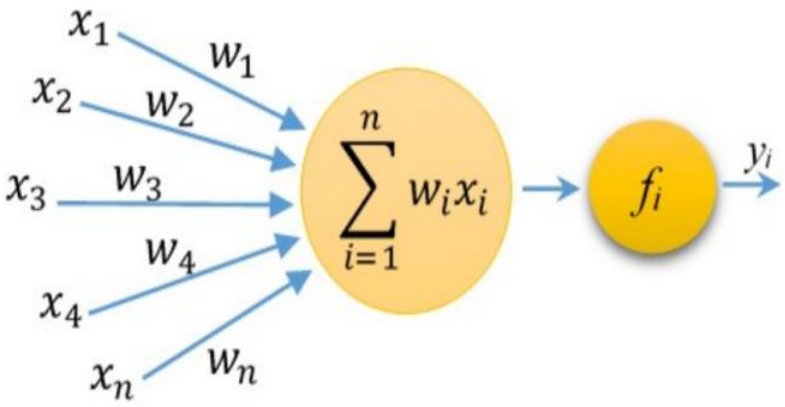

Fig II-2: The artificial processing Unit (Neuron) [JANTAN, et al., 2017]

Involving a specific search technique, a perceptron can be trained with a set of data samples to predict time series, approximate a function (regression) or classify data [Gallo, 2015], however, a simple perceptron can only solve linearly separable problems, to solve a no-linear problem more layers are needed between the input and the output existing layers and that yield a more complex neural network architecture called Multi Layer Perceptron or MLP [Basheer,et al., 2000].

2.1.3. *Multilayer Perceptron:*

As mentioned above, MLP compose from more than the input and the output layers, additional layers reside between those, for that they don't interact with the external environment, those are hidden layers, the primary goal of extending a perceptron with hidden layers is to enhance its capabilities to solve nonlinear classification problems by processing the input data and pass them to the output layer [Basheer,et al., 2000]. **Fig II-3** shows a three layer feed forward MLP.

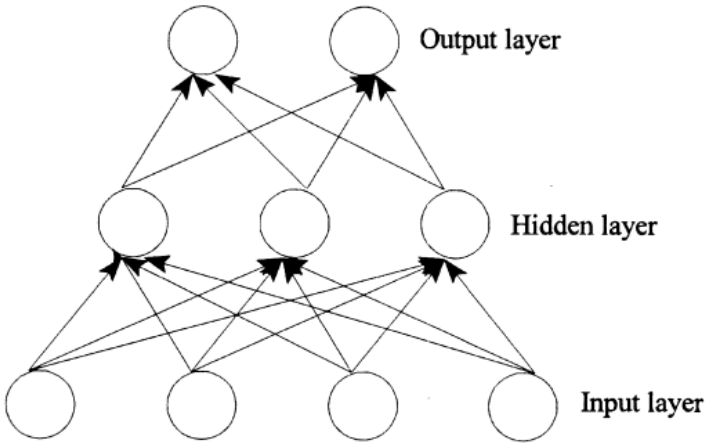

Fig II-3: three layer feed forward MLP

2.1.4. *Activation functions:*

Activation functions as its name indicated are used to distinct the state of a given neuron as it's ON or OFF, by calculating the activation output amount at that neuron level [Basheer,et al., 2000].

There are many activation function exist in the literature, and there is no theoretically rule to define which to use in a given layer. **Tab II-1** listed some of the well known functions [Gallo, 2015].

Function	Formula
Linear	$f(x) = x$
Logistic(sigmoid)	$f(x) = \frac{1}{1+e^{x}}$
Logistic symmetric	$\frac{1+e^{-x}}{2}$
Hyperbolic tangent	$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
Corrected tangent	$f(x) = \tanh(c \cdot x)$
Sinusoidal	$f(x) = \sin(x)$
Gaussian	$f(x) = e^{-x^2}$
Inverse Gaussian	$f(x) = 1 - e^{-x^2}$

Tab II-2: list of activation functions.

2.1.5. *Training an artificial neural network:*

In a supervised learning, neural networks are explicitly trained to get a given task done, and not directly programmed, this process require a training technique and a dataset with inputs labelled with its corresponding outputs according to those the learner adjust weights of interconnections, starting from random values to the global minimum[Gallo, 2015].

The characteristics of a training technique are as follow: [Alba, et al., 2006]

- *Efficiency*: an efficient technique gets to the global minimum with the lowest cost.
- *Robustness*: The algorithm should be resistant to unwanted noise.
- *Independence* of the initial conditions.
- *Generalization*: technique should be able to show adequate after the training phase.
- *Scalability* with the size and complexity of the data.

This section presents one of the classical training techniques for training a neural network.

Back Propagation algorithm: [Gallo, 2015] [Basheer,et al., 2000]

The Back Propagation algorithm is the most widely used algorithm in the learning task of neural networks. It comes from the stochastic gradient descent algorithm which aims to minimize a cost function E of the form:

$$E(w) = \frac{1}{2}\sum_{k}^{N}\sum_{i}^{m}(y_i(k) - y_{i,d}(k))^2 \quad \textbf{Eq II-2}$$

With:

$y_i(k)$: The i^{th} output of the neural network corresponding to input k.

$y_{i,d}(k)$: Desired output corresponding to $y_i(k)$.

N: represents the length of the training sequence.

m: the total number of neurons in the output layer.

The back propagation algorithm can be divided into two steps:

Forward Step: propagating input data to the network layer by layer to reach the output layer, where the network error E (w) is computed.

Backward Step: propagating the calculated error made by the network backwards, to updated weights according to the amount of change calculated according to the equation $\Delta(w)$.

$$\Delta(w) = -\eta\left(\frac{\partial E_{rr}}{\partial w(k)}\right) \quad \textbf{Eq II-3}$$

The weights are then adapted with the Delta rule equation as follow:

$$w(k+1) = w(k) + \Delta(w) \quad \textbf{Eq II-4}$$

BP has been broadly used to train a neural networks, and it shows an efficacy for solving many problems (e.g., classification), though so, it have the drawback of being extremely slow in convergence trends using of gradient descent method and getting fall into local minima, to address this problem metaheuristics optimization algorithms are used to optimize the process of training a neural network as discussed up the following section.

2.2. Metaheuristics Algorithms: [Mirjalili, et al., 2015], [Sevaux, et al., 2010]

Obtaining an optimal or near-optimal solution for a modern no-linear, complex optimisation problem based on exact mathematical method can be a difficult task, as there are no guaranties that a solution exists, even for simple linear problems sometimes, for that metaheuristic algorithms have been the focus of many studies in the field of optimization, and well known techniques for finding an improved and a fittest solutions to a given objective functions.

metaheuristics are 'stochastic' mechanisms developed specifically to find a solution that is "good enough" in a relatively "acceptable duration" to an optimization problem such as NP-hard problem, therefore it does not guarantee an optimal solution as the exact methods that come with a proof of existence of such solution, However, researchers has evidenced its capabilities and sometimes its superiority in balancing between the quality of solution and the time needed for that, also its independence of the optimisation problem, comparing to the traditional exact methods.

Most of metaheuristic algorithms inspired from natural phenomena, genetic algorithms (GA) [Holland, 1992] as an example were inspired from the survival of individuals in nature (Darwin's theory) by selecting and combining them in order to get a fitter individuals. Social and individual thinking of flying birds was the inspiration process of particle swarm optimization (PSO) [Kennedy, et al., 1995] where condidas shares its best food position reached so far.

2.2.1. ***Metaheuristic algorithms classification:*** [Almufti, 2019] [Sevaux, et al., 2010]

In order to get to best, fittest solution, metaheuristic algorithms operate on a represented (encoded) set of potential solutions, by evaluating and perform a succession of operations on them. Based on the way of representation and manupilation of solutions, it can be distinguished two class of metaheristics:

i. Single solution based algorithms:

Single solution based algorithms or (local search algorithms) generate a random solution and enhance it by performing simple modifications on its state to obtain a new solution, until an optimum solution is reached. Tabu search (TS) [Glover, 1990] is a famous technique fall under the local search category.

ii. Population based algorithms:

Population based algorithms employ selection from a set of randomly generated potential solution (usually called population), combining those to generate a new improved set repetitively until a stopping criteria reaches.

Evolutionary and swarm intelligence algorithms are two sub-set of population based algorithms. Evolutionary algorithms including genetic algorithms, genetic programming and evolutionary computing mimic the principles of natural evolution; swarm intelligence algorithms imitate the interaction and communication process of swarms like bird and bee swarms.

In recent years, researchers combined ideas from different categories to build new optimization methods, and those techniques were known in the literature as hybrid metaheuristic algorithms.

by concediring Neural networks training process as an optimization problem, metaheuristic algorithms will be a good alternative for the BP algorithm, in a sence that it would minimise time needed for trainig and evode falling in the local minima problem.

2.3. Tree seed algorithm (TSA):

2.3.1. *The Original TSA:* [Kiran, 2015] [Kiran, et al., 2018]

Tree Seed Algorithms is a population-based iterative search algorithm developed By [Kiran, 2015] for solving continuous optimization problems, inspired by the relationship of the tree with its seeds, means by which trees spread to the space, those grow over time to become new trees, by considering the spreading space is the search space, the location of trees and seeds are possible solution for the optimisation problem The main idea in TSA is generating seeds as children based on its parent tree, calculating fitness of seeds and updating the tree position in trees population if the seed fitness is better than its parent fitness; TSA employs exploration and exploitation mechanism and tries to equilibrate between them to locates new better solutions, **Fig II-4** presents the framework of TSA algorithms.

```
Step1:  Set algorithm parameters
        Set the value of the Search Tendency (ST) parameter to be between (0-1)
        Set the dimension of the problem (D) and number of individuals in the population (N)
        Set termination criteria
Step 2: Generate the initial population
        Generate the initial population randomly according to the dimension of the problem and the
        constraints
        Calculate the fitness value for each individual of the population
        Select the tree with the best fitness value (BestTree)
Step3: search with seeds
        FOR all trees
                Set number of seeds at random between 1 and N (k)
                FOR i=1 to k
                        FOR j = 1 to D
                                IF (rand < ST)
                                        Update the relevant dimension according to Eq II-6
                                ELSE
                                        Update the relevant dimension to Eq II
                                END IF
                        END FOR
                        Select the seed with the best fitness value among all seeds
                        Compare the best seed with the parent tree (parentTree)
                        IF the seed is in a better position than the parent tree, then update the position of
                        parent tree by position of the seed
                        END FOR
Step4: determine the best solution in the population
        Select the tree with the best fitness value (BestTree)
        Step 5 check the termination condition
        IF termination condition is not met
                Go to step 3
        ELSE
                Report the best solution
        END IF
        END.
```

Fig II-4: pseudo code for tree seed algorithm *[Kiran, 2015]*

In order to achieve the best quality solution in the local and global space, TSA generate random population in the exploration phase those are the parent trees using the **Eq-II-5** where $\boldsymbol{L}$ and $\boldsymbol{H}$ represent respectively the lower and the higher bounds of the search space and $\boldsymbol{r}$ is a random value in the range of [0,1] generated for every tree and every dimension; in the second phase, a control parameter named search tendency (ST) handles the search propensity to the best or parent tree to approach to it, the number of seeds generated from a tree is selected randomly between 1 and the number of generated trees in the first phase, every dimension of a generated seeds is created either by **Eq II-6 or Eq II-7** dependent on ST value and a random generated value between 0 and 1, the related dimension is updated according to **Eq II-6** if the $\boldsymbol{rand}$ value is smaller than the ST value, else it is updated with the **Eq II-7**.

$$\boldsymbol{Trees_{i,j} = L_{j,min} + r_{i,j} * (H_{j,max} - L_{j,min})} \quad \textbf{Eq II-5}$$

$$\boldsymbol{Seed_{i,j} = ParentTree_j + (BestTree_j - Trees_{r,j}) * (rand - 0.5) * 2} \quad \textbf{Eq II-6}$$

$$\boldsymbol{Seed_{i,j} = ParentTree_j + (ParentTree_j - Trees_{r,j}) * (rand - 0.5) * 2} \quad \textbf{Eq II-7}$$

Where $\boldsymbol{Seed}$ is the current generated childen population from the selected tree $\boldsymbol{ParentTree}$, $\boldsymbol{i}$ represent the i^{Th} seed been created and $\boldsymbol{j}$ represent the current dimension been created or been creating from. $\boldsymbol{BestTree}$ Is the current tree with best fitness and $\boldsymbol{Trees_r}$ is a random tree selected from tree's population with a random index $\boldsymbol{r}$.

2.3.2. ***Improved Tree Seed Algorithm (ITSA):*** [Kiran, et al., 2018]

The main benefit of meta-heuristics optimisation algorithms is the convergence speed to optimum or near-optimum solution for complex optimization problems, TSA as a new nature-inspired algorithms in the literature had difficulties to find the best solution when the dimensions of the problem increases, therefore the authors in [Kiran, et al., 2018] proposed some ameliorations to the original TSA to enhance its performance; this section highlight those ameliorations.

To achieve a best exploration and exploitation, acceleration Coefficient C parameter was introduced, which is directly related to the dimensions of the problem and can be calculated according to **Eq II-8**.

$$C = 2 - (D^2 * 0.0001) \quad \textbf{Eq II-8}$$

Therefore, equation **Eq II-6** and **Eq II-7** are affected by a variation factor Δ calculated according to equations **Eq II-9** and **Eq II-10** and are also controlled by searching tendency coefficient; as well some restrictions are made on Δ value by setting its lower and higher bounds $\mathcal{M}in$ (**Eq II-11**) and $\mathcal{M}ax$ (**Eq II-12**) respectively to boost the efficiency of the parent tree.

$$\Delta_{i,j} = \left(BestTree_j - Trees_{r,j}\right) * (rand - 0.5) * C \quad \textbf{Eq II-9}$$

$$\Delta_{i,j} = \left(ParentTree_j - Trees_{r,j}\right) * (rand - 0.5) * C \quad \textbf{Eq II-10}$$

$$\mathcal{M}in = -0.1 * (d_{max} - d_{min}) \quad \textbf{Eq II-11}$$

$$\mathcal{M}ax = 0.1 * (d_{max} - d_{min}) \quad \textbf{Eq II-12}$$

d_{max} And d_{min} are outer limits of the search space, as a result the seed dimension now calculated as **Eq II-13**:

$$Seed_{i,j} = ParentTree_j + \Delta_{i,j} \quad \textbf{Eq II-13}$$

2.4. NSL-KDD dataset:

Training and testing a machine learning model to detect an intrusion can be done only if there is an efficacious dataset, NSL-KDD data set is a container of a remarkable amount and quality network data, selected and corrected from its previous version the KDDcup99 dataset [Dhanabal, et al., 2015].

KDDcup99 dataset is a well known benchmark dataset for testing an IDS model, it was build based on DARPA'98 which is a result of 9 weeks tcpdump data from a simulated network of a typical US AIR FORCE real network, Training KDDcup99 set contained about 5 million connection records with 41 features for each, labelled either as normal or one of 21 different attacks, with about two million records for the testing set with additional 16 attack types all categorized into four main classes presented below [Tavallaee, et al., 2009], [Protić, 2018] :

- *Denial Service of Attack (DoS*): by flooding the target with traffic, DoS attack deprives legitimate users from reaching it.
- *User to Root Attack (U2R):* By exploiting vulnerability, an attacker initiates an attempt to gain root access to the system after gaining access to a normal user account.
- *Remote to Local Attack (R2L):* an attacker attempts to gain access to a remote machine, by exploiting a security flaw.
- *Probing Attack (Prob):* fingerprinting and evading systems and security protocols in a network in order to find vulnerabilities for later exploit.

The KDDcup99 dataset contains about 78% redundant records and about 74% duplicate records and those negatively impact the training process [Protić, 2018], to address this issue the authors in [Tavallaee, et al., 2009] conducted a statistical analysis on it and proposed the NSL-KDD,

which consists of no redundant records in the training set nor duplicate records in the testing set [Protić, 2018] , with 125,973 and 22,544 records for training and testing sets respectively, all have the same collection of original features in the KDDcup99 dataset, those are listed and described in **TAB II-2:**

Index	Feature name	Description
1	Duration	The length of the connection (seconds)
2	Protocol type	Type of protocol (TCP, UDP ...)
3	Service	Destination service(ftp, telnet ...)
4	Flag	Status of connection
5	Source bytes	No. of Bytes from source to destination
6	Destination bytes	No. of B from source to destination
7	land	If the source and destination address are the same land=1/if not, then 0
8	wrong fragments	No. of wrong fragments
9	urgent	No. of urgent packets
10	hot	No. of hot indicators
11	failed logins	No. of unsuccessful attempts at login
12	logged in	If logged in=1/if login failed 0
13	Num compromised	No. of compromised states
14	root shell	If a command interpreter with a root account is running root shell=1/if not, then 0
15	Su attempted	If an su command was attempted su attempted=1/if not, then 0 (temporary login to the system with other user credentials)
16	Num root	No. of root accesses
17	Num file creations	No. of operations that create new files
18	Num shells	No. of active command interpreters
19	Num access files	No. of file creation operations
20	Num outbound cmds	No. of outbound commands in an ftp session
21	Is hot login	is host login=1 if the login is on the host login list/if not, then 0
22	is guest login	If a guest is logged into the system, is guest login=1/if not, then 0
23	count	No. of connections to the same host as the current connection at a given interval
24	srv count	No. of connections to the same service as the current connection at a given interval
25	serror rate	% of connections with SYN errors
26	srv serror rate	% of connections with SYN errors

27	rerror rate	% of connections with REJ errors
28	srv rerror rate	% of connections with REJ errors
29	same srv rate	% of connections to the same service
30	diff srv rate	% of connections to different services
31	srv diff host rate	% of connections to different hosts
32	dst host count	No. of connections to the same destination
33	dst host srv count	No. of connections to the same destination that use the same service
34	dst host same src rate	% of connections to the same destination that use the same service
35	dst host srv rate	% of connections to different hosts on the same system
36	dst host same srv port rate	% of connections to a system with the same source port
37	dst host srv diff host rate	% of connections to the same service coming from different hosts
38	dst host serror rate	% of connections to a host with an S0 error
39	dst host srv serror rate	% of connections to a host and specified service with an S0 error
40	dst host serror rate	% of connections to a host with an RST error
41	dst host srv serror rate	% of connections to a host and specified service with an RST error

Tab II-3: KDD data set features [Protić, 2018]

Those futures are classified on three classes, Basic features, Traffic features and Content features [Tavallaee, et al., 2009]:

- *Basic features:* those can be extracted from a packet header. (Duration, protocol type, service, flag, source bytes and destination bytes).
- *Traffic features :* this category can be divided into two subcategories of features: those related to the same host and others related to the same services and they are computed in past 2 seconds interval window :
  - *"Same host" features:* examine connections to the same host as the considered connection.
  - *"Same service" features:* examine connection with the same services as the considered connection.
- *Content features:* those are gathered by analysing of the contents of a packet.

Conclusion:

This chapter dives into the concept of artificial neural network and how they inspired from the biological neural networks, how they can be trained along with their training disadvantages and

shows how metaheuristics algorithms can be a good solution for optimizing a neural network model, besides to that a more technical side of this work was presented, TSA algorithms as a new optimization algorithms was detailed along with a well known benchmark dataset for testing intrusion detection machine learning models.

The next chapter will be the practical side of this study; it well presents steps we had to develop a new model for detecting network anomalies, down to the discussing results we obtained.

Chapter III:

Experiments and results discussion

Introduction

Metaheuristic based optimisation ANNs reside on threes methodologies, at first, the optimisation algorithms can be used to find the collection of weights and biases that minimise the error and decrease the cost function for the ANN, secondly, it can be used to find the appropriate structure of the ANN that fit for a particular problem, lastly, it is applicable to adjust a gradient-based learning algorithm parameters.

In this work a combination of weights and biases of MLP is optimised using the improved tree seed algorithm in order to develop a more accurate network anomaly detector model, to evaluate the developed model GA and PSO algorithms are implemented for the aim of comparing performances. This chapter aim to presents main stages of experiment along with technologies used for it, to conclude with discussing the obtained results.

1. System paradigm:

1.1. Representation of parameters:

There are fully used types of ANN in the literature. In this work we are interested in unidirectional data feeding MLP or feed forward neural network (FFNN) with one hidden layer, we donate $\hbar$ the number of input neurons, ℓ the number hidden neurons and m number of output neurons, therefore, the number of weights is calculated as $(\hbar + m) * \ell$ and the number of biases is $\ell + m$ to adopt the collection of weights and biases for the ITSA algorithms an encoding is needed therefore a variable length vector holding floats values is used to represent parameters, **Fig III-1** illustrate the proposed encoding (representation).

1.2. Fitness function:

There are numerous implemented methods to evaluate the performance of a neural network model such as MSE, NMSE, RMSE, true positive rate, false positive rate, F-measure, and accuracy. We choose to use MSE (as shown in **Eq III-1**) as a fitness function due to its widespread use. For that the objective of the ITSA algorithms is to minimize the value of MSE for the neural network model

$$\mathbf{MSE} = \frac{\mathbf{1}}{\mathbf{N}}\sum_{\mathbf{1}}^{\mathbf{N}}(\mathbf{y_i} - \mathbf{y_{i,d}})^{\mathbf{2}} \quad \textbf{Eq III-1}$$

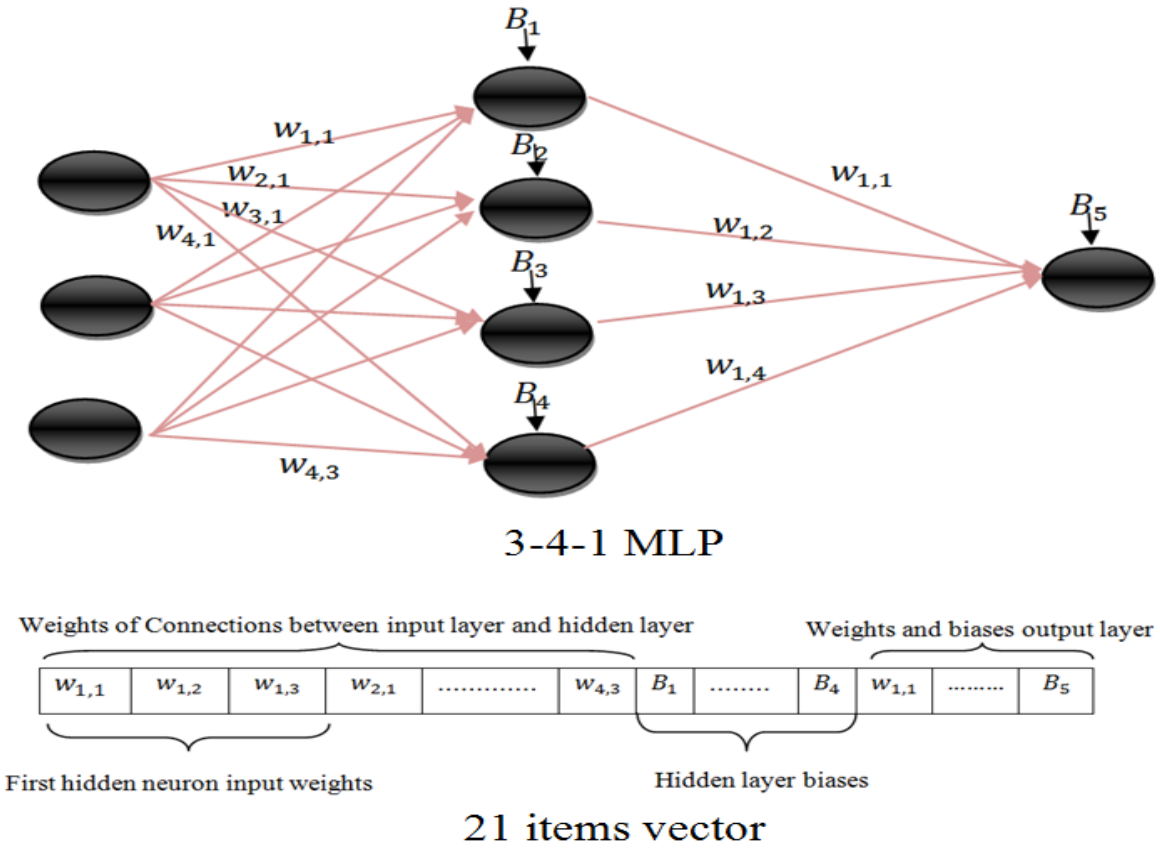

Fig III-1: illustration of parameters representation

1.3. Architecture and chosen parameters:

In this study we focus on optimizing weights and biases of an MLP with fixed structure, for that a fixed number of neurons is selected previously, input neurons are calculated according to the number of selected features from the NSL-KDD dataset, 60 neurons for the hidden layer, with 1 output neuron for the reason of classifying feed data as normal or attack as described later in the dataset pre-processing section.

Our model assumes a sigmoid function as a transfer function in both hidden and output layers.

We tried to set the optimal values of parameters for all three used optimizations algorithms to make their convergence speed to the optimized solution relatively close, therefore, the search tendency parameter of the ITSA algorithms was set to ST = 0.715 along with boundaries conditions on number of generated seeds for every tree with 10% of the population length as a minimum and 25% as maximum to evade overly long time execution while training. The PSO cognitive and social coefficients (also known as learning factors)

were specified as suggested in [Clerc, et al., 2002] with the value of 1.49445 to ensure the convergence with ω=0.65 for the weight coefficient, GA in the other hand with its 4 parameters where settled to cross_rate= 0.8, and mutation_rate = 0.08 as control parameters and elite_proportion=0.3 tornement_proportion = 0.2 as selection parameters. Along with those specific parameters the Number of population was shared among all algorithms with 50 elements in every population.

1.4. Dataset Pre-processing:

In the first step of dataset pre-processing we tried to investigate variation of values among features where we observe that 'num_outbound_cmds' and 'is_host_login'features were univariate and don't play a significant role in classification process, for that we eliminate them, the second step made is to change the 'label' column from multiclass to binary class by changing attack labels from its original to be labelled as 'attack' as the code shows for both the training set (**Fig III-2**) and the testing set (**Fig III-3**)

```
dataframe['label'] = dataframe['label'].replace(['back', 'buffer_overflow', 'ftp_write', 'guess_passwd', 'imap',
    'ipsweep', 'land', 'loadmodule', 'multihop', 'neptune', 'nmap', 'perl', 'phf', 'pod', 'portsweep',
    'rootkit', 'satan', 'smurf', 'spy', 'teardrop', 'warezclient', 'warezmaster'], 'attack')
```

Fig III-2: Python code to change 'label' colomn from multiclass to binary class(training set)

```
dataframe['label'] = dataframe['label'].replace(['neptune', 'guess_passwd' , 'mscan' , 'warezmaster','apache2' ,'satan',
    'processtable','smurf','back','snmpguess','saint','mailbomb','snmpgetattack','portsweep','ipsweep','httptunnel','nmap',
    'pod','buffer_overflow','multihop','named','ps','sendmail','rootkit','xterm','teardrop','xlock','land','xsnoop','ftp_write',
    'perl','phf','worm','udpstorm','sqlattack','loadmodule','imap'], 'attack')
```

Fig III-3: Python code to change 'label' colomn from multiclass to binary class(testing set)

Since ANN doesn't deals with other types of information but numerical, the next step was encoding categorical columns into numerical ones with a 'label encoding' method consist of replacing each value in a column to a number, starting from 0 and ending to the number of values minus one, the code below select and encode all categorical column at once.

```
cat_columns = dataframe.select_dtypes(['category']).columns
dataframe[cat_columns] = dataframe[cat_columns].apply(lambda x: x.cat.codes)
```

Fig III-4: Python code for label encoding categorical columns

NSL-KDD dataset values are measured on high different scales, feeding a model with such data might end up creating a bias. To address this issue a wise-feature standardisation technique is applied on the dataset, the main idea is to standardize columns individually before applying learning phase, The standard value of a sample x in column X is calculated as shows Eq III-2 where μ is the mean of X, calculated by the formula in Eq III-3and σ represents standard deviation calculated as Eq III-4, the Scikit-learn open source library has a built-in method to standardize a dataset with a single line as shown in **FIG III-5**.

$$z = \frac{(x - \mu)}{\sigma} \quad \textbf{Eq III-2}$$

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (x_i) \quad \textbf{Eq III-3}$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2} \quad \textbf{Eq III-4}$$

```
x_train = scaler.fit_transform(train_df)
x_test = scaler.transform(test_df)
```

Fig III-5: Python Code for standardisating Dataset

The mean and standard deviation are stored in the first line with the fit transform method to use on testing dataset using the transformation method to maintain the unseen data.

Based on the above, the **Fig III-6** illustrates our proposed framework to train an MLP with ITSA algorithm.

2. Evaluation metrics:

For measuring the performance of our proposed model, accuracy (ACC), detection rate (DR) and false alarm rate (FAR) metrics are used, those are defined in : [Ghanem, et al., 2020]

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad \textbf{Eq III-5}$$

$$DR = \frac{TP}{TP + FN} \quad \textbf{Eq III-6}$$

$$FAR = \frac{FP}{TN + FP} \quad \textbf{Eq III-7}$$

Where:

- True Negative (TN): represents the amount of normal data detected as normal.
- True Positive (TP): represents the amount of abnormal data detected as abnormal.
- False Negative (FN): represents the amount of abnormal data detected as normal.
- False Positive (FP): represents the amount of normal data detected as abnormal.

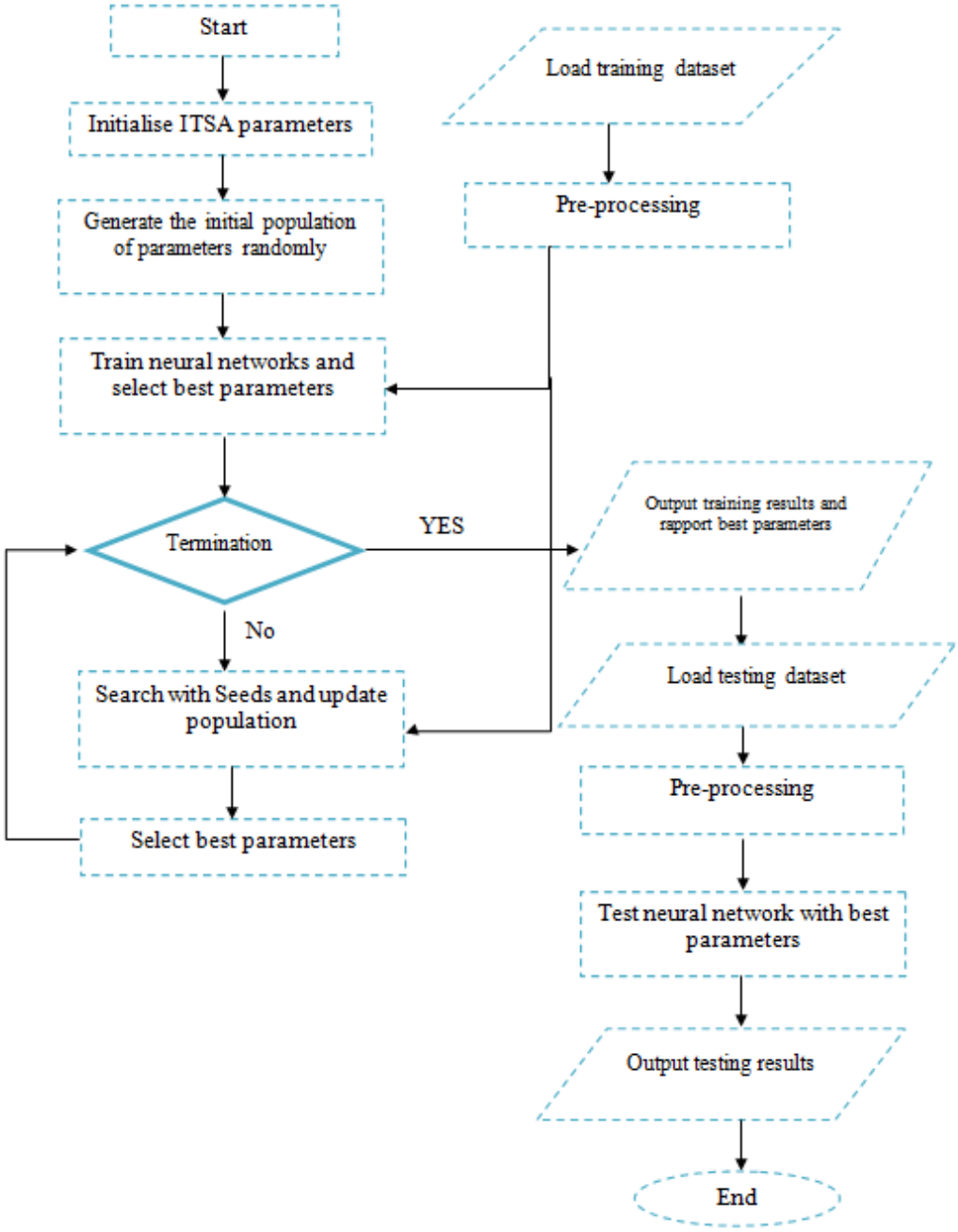

Fig III-6: proposed framework

3. Implementation tools:

We relied on the following technologies in our work to develop the proposed model:

3.1. Python programming language:

Python is the open source programming language used by many computer scientists in the field of data science and machine learning, the installed version at the development tie was 3.8.8 due to its compatibility with 32bits windows system.

Fig III-7: Python programming language's logo

3.2. Pandas library:

Pandas is a Python library of data structures and statistical tools facilitate working with datasets in memory [McKinney, 2010]. Version 1.2.4 was installed for the developing purpose.

Fig III-8: Pandas library's logo

3.3. Scikit-learn library:

Scikit-learn library offers an implementation of machine learning algorithms and statistical data analysis methods to non-specialists using a general-purpose high-level language [Pedregosa, et al., 2011]. It's used in our work as an easy way for standardisation of dataset.

Fig III-9: Scikit-learn library's logo

3.4. NumPy library:

Is an open source library realised to offer a flexible use of multi-dimensions arrays, it enable numerical computing with Python and contain a built in linear algebra, and random number functionalities, our proposed model is leaned on version 1.19.2.

Fig III-10: NumPy library's Logo

3.5. Matplotlib library:

Matplotlib is an open source library for Python programming language intended for plotting and visualizing data in the form of graphs. Using general-purpose GUI toolkits Matplotlib provides an API for embedding plots into applications. Version 3.3.4 was used in our work to output graphical results of the proposed model.

Fig III-11: Matplotlib library's Logo

4. Results and discussion:

This study focus on optimizing the set of weights and biases of a multi-layer perceptron using the new improved tree-seed optimization algorithm, the proposed framework performance was compared against PSO and genetic algorithms performances, for the sake of reducing run-time complexity a proportion of 20% from the NSL-KDD train and test dataset which has been mentioned in section 2.4 were feed to the model for 200 epochs for each algorithm. This section is dedicated to discuss obtained results from the training and testing phases.

Fig III-12 shows a representative sample of the convergence plots of all three algorithms toward the best MSE value while training, where ITSA shows a significant convergence speed along with PSO, with approximate average of MSE of 0,032 and 0.047 for the PSO and ITSA respectively, the rate of convergence in the case of GA was slower.

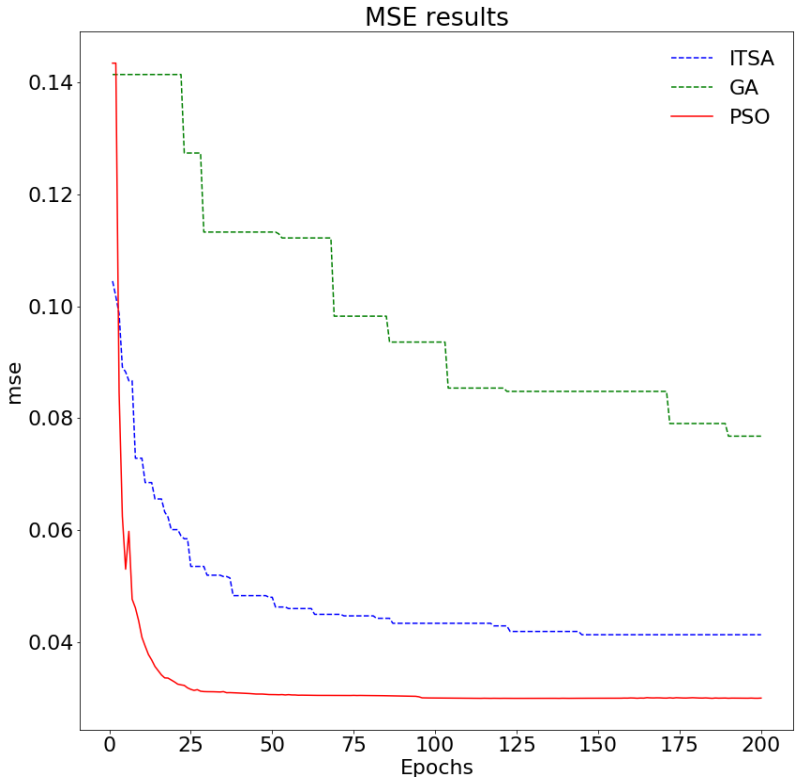

Fig III-12: Convergence Curves of approaches based on MSE value (MSE Vs epochs)

Furthermore, the GA algorithm obtained lower false alarm rate (FAR) compared with the other algorithms (average of 200 epochs), though it has the lower accuracy(ACC) and detection rate (DR) , as shown in **Fig III-13**, ITSA and PSO show notable superior results in those metrics as well

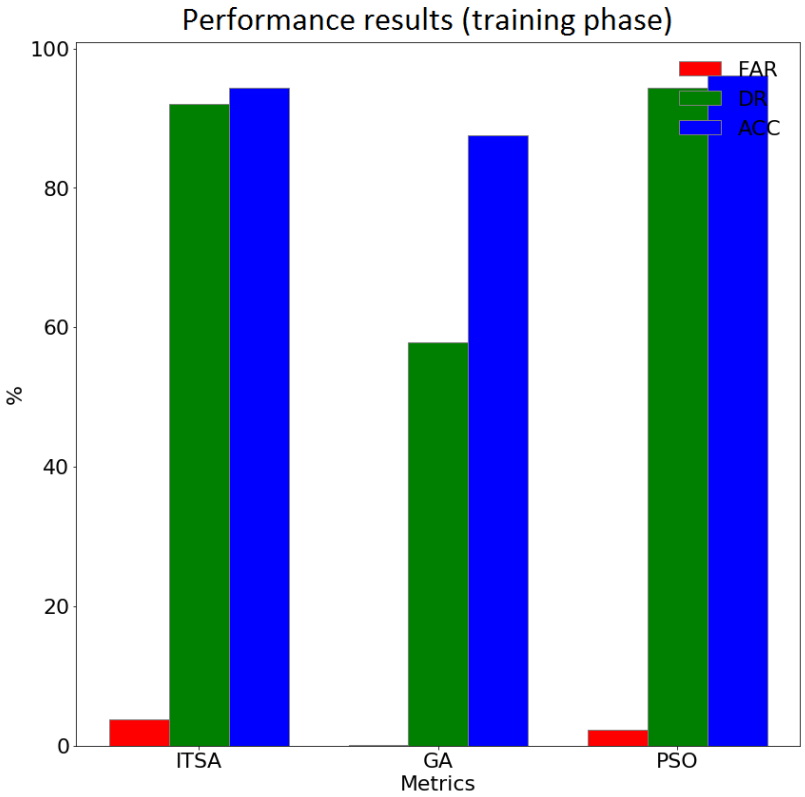

Fig III-13: Training performance results (average)

as MSE results. **Tab III-1** numerically detailed those results.

. **Fig III-14** shows graphical comparison of testing performance results achieved by the three approaches, where we can observe the proximate results for the three algorithms considering ACC and DR measurements, but ITSA shows a high false alarm rate comparing to the other algorithms, also those measurements are detailed numerically in **Tab III-1**

	Training performances(average of 200 epochs)				Testing performances			
	MSE	**ACC(%)**	**FAR**(%)	**DR(%)**	**MSE**	**ACC(%)**	**FAR(%)**	**DR(%)**
ITSA	**0.0474**	**94.2976**	**3.7279**	**91.9963**	**0.1853**	**76.0700**	**5.4723**	**62.1695**
PSO	**0.0326**	**96.1562**	**2.3322**	**94.3944**	**0.2039**	**77.0680**	**2.1166**	**61.3919**
GA	**0.0989**	**87.5239**	**0.0004**	**57.9664**	**0.1748**	**74.4289**	**2.3748**	**56.9595**

Tab III-1 Numerical testing preformance results

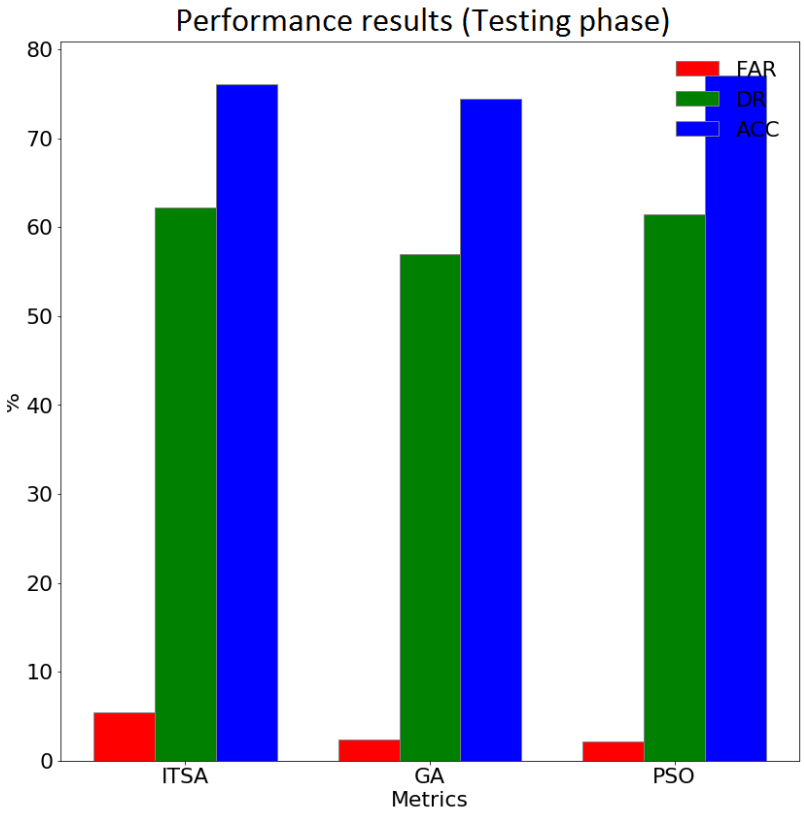

Fig III-14 Testing performance results

5. Conclusion:

This chapter introduces the followed methodology to fulfill the development of our proposed model, from parameter representation through data preprocessing toward feeding and testing it with a proportion of the NSL-KDD dataset along with a presentation of technologies employed to attain that.The performance of the proposed model shows that a trained neural network with the improved tree seed algorithm can achieve high accuracy and detection rate.

General Conclusion:

During this study, we aimed to build a new predicting model based on artificial neural network techniques to detect intrusions relying in training it on improved tree-seed algorithms, as an attempt to improve lines of defences against computer worm propagation.

Our journey starts with presenting how computer worms can be a threat to personal privacies and network infrastructures on which economies, administrative bodies and even military systems depend, through the way it finds its victims, how it infects them, as well as its stealthy methods to hide from detection systems; Last-mentioned as an important line of defence from computer worm attacks, are presented as a second part of the first chapter, we focused on the technical side of anomaly detection systems implementation as in introduction for our proposed learner technique.

The second chapter clarified needed concepts for developing an anomaly detection system from ANN to metaheuristic algorithms living it with more technical part by presenting ITSA algorithm details beside the famous NSL-KDD dataset.

Third chapter introduce the proposed model which is based on training a multi-layer perceptron with one hidden layer using the stochastic algorithm ITSA to detect anomalous behaviour, the model has been trained and evaluated with 20% proportion of the NSL-KKD dataset and its performance compared against the performance of two most widely known metaheuristic algorithms namely Genetic Algorithms (GA) and Particle Swarm Optimisation (PSO), satisfying results has been obtained considering the few number of feed data.

Despite the invested time and effort to develop new detection and counter-measure approaches, it cannot be said that the situation is stable; as a matter of fact more sophisticated and more complex generation of worms is already spreading, especially with the development of smart phones that maybe will the next target for new worm major attack.

Bibliography:

Adetunmbi A.Olusola., Adeola S.Oladele and Daramola O.Abosede Analysis of KDD ’99 Intrusion Detection Dataset for Selection of Relevance Features [Conference] // World Congress on Engineering and Computer Science. - San Francisco, USA : [s.n.], 2010. - Vol. 1.

Alba Erique and Marti rafael Metaheuristics procedures for training neural networks [Book]. - [s.l.] : Springer science , 2006.

Ali Mohammed [et al.] A New Intrusion Detection System Based on Fast Learning Network and Particle Swarm Optimization [Journal] // IEEE Access. - 2018.

Ali Sulieman Sulieman Mohamed and Fadlalla Yahia A Detecting Zero-day Polymorphic Worm: A Review [Journal] // IEEE. - 2008.

Almufti Saman M Historical survey on metaheuristics algorithms [Journal] // International Journal of Scientific World. - 2019. - 1 : Vol. 7.

Avinash Wadhe, Rahul Suryawanshi and Suryawanshi Rahul Novel Approach for Worm Detection using Modified Crc32 Algorithm [Conference] // National Conference on Innovative Paradigms in Engineering & Technology / ed. Applications International Journal of Computer. - 2012.

Basheer Imad and Hajmeer M. Artificial neural networks: fundamentals, computing, design, and application [Journal] // Journal of Microbiological Methods. - 2000. - Vol. 43. - pp. 3–31.

Ben Amor Nahla, Benferhat Salem and Elouedi Zied Naive Bayes vs Decision Trees in Intrusion Detection Systems [Conference] // ACM symposium on Applied computing. - 2004. - pp. 420-424.

Bhuyan Monowar H, Bhattacharyya D. K and Kalita J. K Network Anomaly Detection: Methods, Systems and Tools [Journal] = 01 // IEEE COMMUNICATIONS SURVEYS & TUTORIALS. - 2014. - 16. - pp. 303-336.

Cinar Ahmet Cevahir Training Feed-Forward Multi-Layer Perceptron Artificial Neura lNetworks with a Tree-Seed Algorithm [Journal] // Arabian Journal for Science and Engineering. - [s.l.] : springer, 2020.

Clerc M and Kennedy J The particle swarm-explosion, stability and convergence in a multi dimensional complex space. [Journal] // IEEE Trans Evolut Comput. - (2002). - 2 : Vol. 6. - pp. 58–73.

Dhanabal L and Shantharajah S.P A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms [Journal] // International Journal of Advanced Research in Computer and Communication Engineering. - 2015. - 6 : Vol. 4.

Gallo Crescenzio artificial neural networks Tutorial [Journal] // Encyclopedia of Information Science and Technology. - [s.l.] : IGI Global, 2015. - 3. - pp. 6369-6378.

Ghanem Waheed [et al.] An Efficient Intrusion Detection Model Based on Hybridization of Artificial Bee Colony and Dragonfly Algorithms for Training Multilayer Perceptrons [Journal] // IEEE Access. - 2020.

Ghanem Waheed and Jantan Aman New approach to improve anomaly detection using a neural network optimized by hybrid ABC and PSO algorithms [Journal] // Pakistan Journal of Statistics. - 2018. - Vol. 34. - pp. 1-14.

Ghanem Waheed and Jantan Aman Swarm Intelligence and Neural Network for Data Classification [Journal] // Proceedings - 4th IEEE International Conference on Control System ICCSCE 2014. - 03 2015. - pp. 196-201.

Gideon Creech and Jiankun Hu A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguous and Discontiguous System Call Patterns [Journal] // IEEE TRANSACTIONS ON COMPUTERS. - APRIL 2014. - 04 : Vol. 63. - pp. 807-819.

Glover Fred Tabu search [Journal] // INFORMS journal on computing. - 1990. - Vol. 2. - pp. 4-32.

Holland John adaptation in natural and artificial systems [Book]. - [s.l.] : MIT press, 1992.

Jantan Aman and Ghanem Waheed Training a Neural Network for Cyberattack Classification Applications Using Hybridization of an Artificial Bee Colony and Monarch Butterfly Optimization [Journal] // Neural Processing Letters. - [s.l.] : Springer, 2020. - Vol. 51.

JANTAN AMAN, GHANEM WAHEED and GHALEB SANAA USING MODIFIED BAT ALGORITHM TO TRAIN NEURAL NETWORKS FOR SMAP DETECTION [Journal] // Journal of Theoretical and Applied Information Technology. - 2017. - 24 : Vol. 95. - pp. 6788-6799.

Kanani Pratik [et al.] Survey on Computer Worms [Journal] // International Journal on Recent and Innovation Trends in Computing and Communication. - August 2017. - Vol. 5.

Kennedy J and Eberhart R Particle swarm optimization [Conference] // Proceeding of IEEE international conference on neural networks. - 1995. - pp. 1942-1948.

Khraisat Ansam [et al.] Survey of intrusion detection systems:techniques, datasets and challenges [Journal] // Cybersecurity. - 2019. - 1 : Vol. 2. - pp. 1-22.

Khreich Wael, Granger Eric and Sabourin Robert A survey of techniques for incremental learning of HMM parameters [Journal] // Information Sciences. - August 2012. - pp. 105-130.

Kiran Mustafa Servet [et al.] An Improved Tree Seed Algorithm for Optimization [Journal] // International Journal of Machine Learning and Computing. - 02 2018. - 1 : Vol. 8.

Kiran Mustafa Servet TSA: Tree-seed algorithm for continuous optimization [Journal] // Expert Systems with Applications. - 11 2015. - 19 : Vol. 42. - pp. 6686-6698.

Manasi Gyanchandani, J.L Rana and R.N Yadav Taxonomy of Anomaly Based Intrusion Detection system : a review [Journal] // International Journal of Scientific and Research Publications. - 2012. - 12 : Vol. 02. - pp. 174-187.

Mao Jianchang, Jain A.K and Mohiuddin K.M Artificial neural networks: a tutorial [Journal] // Computer. - [s.l.] : IEEE, 1996. - 3 : Vol. 29. - pp. 31 - 44.

Martin Alejandro, D. Menendez Hector and Camacho David Genetic Boosting Classification for Malware Detection [Conference] // Congress on Evolutionary Computation (CEC). - [s.l.] : IEEE, 2016.

Martins Simone and Ribeiro Celso Metaheuristics and Applications to Optimization Problems in Telecommunications. [Book Section] // HANDBOOK OF OPTIMIZATION IN TELECOMMUNICATIONS. - 2006.

McKinney Wes Data structures for statistical computing in python [Conference] // Proceedings of the 9th Python in Science Conference. - [s.l.] : Sefan van der Walt and Jarrod Millman , 2010. - Vol. 445.

Midhunchakkaravarthy Divya and Ganapathi Padmavathi Malicious Traffic Detection and Containment based on Connection Attempt Failures using Kernelized ELM with Automated Worm Containment Algorithm [Journal] // Indian Journal of Science and Technology. - 2016.

Mirjalili Seyedali, Mirjalili Seyed Mohammad and Mohammad Seyed Multi-Verse Optimizer: a nature-inspired algorithm for global optimization [Journal] // Neural Computing and Applications. - [s.l.] : Springer, 2015. - 3 : Vol. 26.

O. Adetunmbi Adebayo [et al.] NETWORK INTRUSION DETECTION BASED ON ROUGH SET AND K-NEAREST NEIGHBOUR [Journal] // International Journal of Computing and ICT Research. - June 2008. - 1 : Vol. 2.

Peddabachigari Sandhya, Abraham Ajith and Thomas Johnson Intrusion detection systems using decision trees and support vector machines [Journal] // International Journal of Applied Science and Computations. - 2004.

Pedregosa F [et al.] Scikit-learn: Machine Learning in Python [Journal] // Journal of Machine Learning Research. - 2011. - Vol. 12. - pp. 2825-2830.

Prahlad Fogla [et al.] Polymorphic Blending Attacks [Journal] // Technical Report GIT-CC-05-09, College of Computing, Georgia Tech. - 2005.

Pratama Andhika and Adi Rafrastara Fauzi Computer Worm Classification [Journal] // International Journal of Computer Science and Information Security. - April 2012. - 4 : Vol. 10.

Protić Danijela REVIEW OF KDD CUP ‘99, NSL-KDD AND KYOTO 2006+ DATASETS [Journal] // MILITARY TECHNICAL COURIER. - 2018. - 3 : Vol. 66.

Rajesh B., Janardhan Reddy and Reddy Dillip Kumar A Survey Paper on Malicious Computer Worms [Journal] // International Journal of Advanced Research in Computer Science & Technology. - 2015.

Reddy E.Kesavulu Neural Networks for Intrusion Detection and Its Applications [Journal] // Lecture Notes in Engineering and Computer Science. - 2013.

Saeed Imtithal, Abuagoub Ali and Selamat Ali A Survey on Malware and Malware Detection Systems [Journal] // International Journal of Computer Applications. - 2013. - 16 : Vol. 67. - pp. 25-31.

Sevaux Marc [et al.] METAHEURISTICS [Book]. - [s.l.] : Springer, 2010. - 3.

Sheikhan Mansour and Jadidi Zahra Flow-based anomaly detection in highspeed links using modified GSA-optimized neural network [Journal] // Neural Computing and Applications. - 2014. - Vol. 24.

Singh Raman, Kumar Harish and Singla Ravinder Kumar Review of Soft Computing in Malware Detection [Journal] // Special Issues on IP Multimedia Communications. - January 2011.

Smith Craig and Matrawy Ashraf Computer Worms: Architectures, Evasion Strategies, and Detection Mechanisms [Journal] // Journal of Information Assurance and Security. - 2009.

Srinivasu P and Avadhani P.S Genetic Algorithm based Weight Extraction Algorithm for Artificial Neural Network Classifier in Intrusion Detection [Journal] // Procedia Engineering. - [s.l.] : elsevier ltd, 2012. - Vol. 38. - pp. 144-153.

Studnia Ivan [et al.] A language-based intrusion detection approach for automotive embedded networks [Journal] // International Journal of Embedded Systems. - [s.l.] : Inderscience, January 2018. - 1 : Vol. 10.

Szor Peter The Art of Computer Virus Research and Defense [Book]. - [s.l.] : symantec press, February, 2005.

TANG Yong [et al.] Concept, Characteristics and Defending Mechanism of Worms [Journal] // IEICE Transactions on Information and Systems. - May 2009.

Tavallaee Mahbod [et al.] A Detailed Analysis of the KDD CUP 99 Data Set [Journal] // Computational Intelligence for Security and Defense Applications. - [s.l.] : IEEE Symposium, July 2009.

Vinod Kumar and Om Prakash Sangwan Signature Based Intrusion Detection System Using SNORT [Journal] // International Journal of Computer Applications & Information Technology. - 2012. - 03 : Vol. 01. - pp. 35-41.

Wang Tian, Wei Lihao and Ai Jieqing Improved BP Neural Network for Intrusion Detection Based on AFSA [Book]. - [s.l.] : Atlantis Press, 2015.

Weaver Nicholas [et al.] A taxonomy of computer worms [Journal] // Proceedings of the 2003 ACM Workshop on Rapid Malcode, WORM 2003,. - Washington, DC, USA : [s.n.], January 2003.

Xu Jie [et al.] The Generalization Ability of SVM Classification The Generalization Ability of SVM Classification [Journal] // IEEE TRANSACTIONS ON CYBERNETICS. - 2014.

Yan Guanhua, Xiao Zhen and Eidenbenz Stephan Catching Instant Messaging Worms with Change-Point [Journal] // Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats. - april 2008.