UNIVERSITY OF MISKOLC

FACULTY OF MECHANICAL ENGINEERING AND INFORMATICS

Fuzzy Automaton-based Early Detection Model

PhD dissertation

Mohammad Almseidin

Msc in Computer Science

'JÓZSEF HATVANY' DOCTORAL SCHOOL

OF INFORMATION SCIENCE, ENGINEERING AND TECHNOLOGY

ACADEMIC SUPERVISOR

Dr. habil. Szilveszter Kovács

Miskolc, 2019

Declaration

The author hereby declares that this thesis has not been submitted, either in the same or in a different form, to this or to any other university for obtaining a PhD degree. The author confirms that the submitted work is his own and the appropriate credit has been given where reference has been addressed to the work of others.

Miskolc, 2019. December, 01

Mohammad Almseidin

Acknowledgements

All gratitude, praise, and thanks to Allah, who has gifted me with the ability to do this work. I would like to express my enormous gratitude to my supervisor, Dr. habil. Szilveszter Kovács for his unfailing motivation and guidance throughout the course of my research, without his helpful suggestions, advice, and encouragement, this work would not have been possible. Also, I would like to express my enormous gratitude to Professor Mouhammd Al-Kasassbeh from Princess Sumaya University for Technology- Jordan for his stimulating discussion, insight, and helpful advice.

I would also like to acknowledge my gratitude to the staff—academic, administrative, technical, and support of the Department of Information Technology at the University of Miskolc for providing such a positive and nurturing environment in which to conduct my research.

Lastly, the completion of this Ph.D. would not have been possible without the support and encouragement of my family: my dear parents, who have given me all their love and support over the years, I thank them for their unwavering commitment through good times and hard times. My dear wife Nisreen Elkaraki, my lovely daughters Razan and Sara, and all my relatives and friends. My heartfelt thanks to them all. I thank all those whose names are not mentioned here, but who have helped me in some way to accomplish this work.

Table of Contents

List of Tables

List of Figures

Abbreviation

AA	Average Accuracy
ANN	Artificial Neural network
BN	Bayesian Network
CNF	Convex and Normal Fuzzy set
DFSM	Deterministic Finite State Machine
DOS	Denial Of Service
DDOS	Distributed denial of service attack
DNS	Domain Name Server
E	Entropy
FA	Fuzzy Automaton
FIS	Fuzzy Inference System
FIVE	Fuzzy rule Interpolation based on Vague Environment
FRI	Fuzzy Rule Interpolation
FTP	File Transfer Protocol
FFA	Fuzzy Finite-state Automaton
FN	False Negative
FP	False Positive
HIDS	Host Intrusion Detection system
HMM	Hidden Markov Model
IDS	Intrusion Detection system
IG	Information Gain
KDD	Knowledge Discovery and Data mining tools
KH	Kóczy-Hirota (FRI method)
MIB	Management Information Base
MLP	Multi-layer perceptron
NIDS	Network Intrusion detection system
R2L	Remote to Local
RBE-DSS	Rule Base Extension using Default Set Shapes
RMSE	Root Mean Square Error
ROC	Receiver operating characteristic
SNMP	Simple Network Management Protocol
TP	True Positive
TN	True Negative
UDP	User Datagram Protocol

1. Introduction

Nowadays, network administrators face stressful environments with an overload of network traffics. Network traffic needs to be analyzed and investigated to detect abnormalities. The IDS has benefited from the rapid growth of technology; however, intruder techniques have also adapted to the intrusion detection mechanisms' new technological developments. Intruders have continued to advance their techniques and alter their behaviors to avoid detection by recent detection mechanisms. As a result, the danger of attacks has become increasingly more difficult to combat.

Computer and network security systems face different types of sophisticated attacks. One type of sophisticated attack is the multi-step attack. The multi-step attack [1, 2] is an attack composed of several prerequisite steps leading up to the final step which launches an attack targeting the victim's security hole. The attackers follow this technique to avoid detection. The prerequisite steps resemble normal behavior and serve as a subterfuge to facilitate the execution of the final step of the attack. The multi-step attack is a constant challenge for the intrusion detection system because intruders may implement complex attack scenarios, composed of several prerequisite steps, all aimed at executing their final attack [3]. Often, there is a causal relationship between the attack steps and forecasting the next step of attack [4].

There is an increasing need to design and implement an efficient IDS detection mechanism capable of handling different attack scenarios. The IDSs face several challenges including being able to detect multi-step attacks and the boundary problem (applying the binary decisions in the detection mechanism) [5]. In terms of the multi-step attacks, there is a causal relationship between the prerequisite steps which allows for administrators to be able to predict the next step of the attack [4]. Therefore, the multi-step attacks consist of different preliminary phases that can be distinguished from one another. On the other hand, implementing an efficient detection mechanism is also challenged by the boundary problem because there are no clear boundaries and no convincing threshold for defining normal and intrusion traffics [6]. The fuzzy system extends the binary decision to the continuous space, smoothing the boundaries and offering a solution to the boundary problem. Additionally, the results generated by the fuzzy systems are more comprehensible [5].

This work proposes a novel detection method for the multi-step attack built upon Fuzzy Rule Interpolation (FRI) based fuzzy state machine. In that respect, the FRI method instruments the fuzzy state machine to be able to act on a not fully defined state transition rule-base, by offering interpolated conclusion even for situations that are not explicitly defined. The proposed detection method was able to detect the multi-step attack even within the early stages of the attack. Furthermore, it had the ability to extend the binary decision to continuous space. The proposed detection method was performed using fuzzy automaton. The reasoning part of the proposed detection method adopts the FRI method instead of classical reasoning methods. This is done in order to decrease the total number of fuzzy rules required to define the state transition rule-base (simplification) and to offer interpolated results, even when the knowledge representation is not complete.

Also, this work proposes a novel method to detect the abnormality within the network by combining the FRI reasoning with the Management Information Base (MIB) parameters. In that respect, there is no need to deal with raw traffic processing, which is time-consuming, and difficult to compute. This method also eliminates the need for creating a complete fuzzy rule base. The MIB parameters reflect the normal and abnormal nature of the network traffics.

1.1 Aims of Research

One aim of the research was to investigate the capabilities to implement the FRI (FIVE) method in the IDS application area. This investigation is practiced by producing two novel detection models built-upon FRI (FIVE) method. This method not only allows the intrusion detection system to be used in continuous spaces but makes it possible to use a sparse fuzzy rule base. This way, the overall rule base size significantly smaller. Because of its fuzzy rule base knowledge representation nature, it can be easily adapt expert knowledge, and also be suitable for predicting the level of degree for threat possibility.

Another, somewhat distinct goal, to design a novel detection model for the multi-step attack built upon the Fuzzy Rule Interpolation (FRI) based fuzzy state machine which allows the usage of sparse intrusion state transition rule-based, and permitting the system's state to have a degree of the membership function. The fuzzy rule interpolation-based fuzzy automaton (fuzzy state machine) extended with a capability to be suitable for detecting and preventing the multi-step attack in stages, where the planned attack is not fully elaborated. Furthermore, to implement and evaluate the suggested model in practice and comparing it with other detection methods, in order to highlight and discuss the difference between the proposed detection model and others.

1.2 Dissertation Guide

After the introduction, the second chapter gives an overview of the fuzzy set theory, fuzzy control systems, and an extensive introduction to fuzzy rule interpolation. The third chapter first gives a short overview about data mining algorithms, then the IDS model-based data mining algorithms are presented, and the evaluation of IDS model-based data mining algorithms are described in details, for the purpose of studying and investigating the effects of different types of intrusions and also to define the weaknesses and strong points of the algorithms. The fourth chapter presents the investigation capabilities to use FIVE based fuzzy rule interpolation model in the IDS application area, in the design and implementation of the novel detection mechanism for Distributed Denial of Service (DDOS) attacks. Firstly, the recent IDS model-based classical fuzzy inference system is presented briefly before the discussion of the developed (Fuzzy Rule Interpolation Intrusion Detection System (FRI-IDS). The method itself is explained and demonstrated with an application example. Furthermore, in chapter five, another novel method to detect the network abnormality was introduced by combining the (FIVE) FRI reasoning with the Management Information Base (MIB) parameters. The method itself is explained and demonstrated with an application example. The sixth chapter introduces the design of a novel

model for detecting the multi-step attacks built upon FRI (FIVE) based on the fuzzy automaton. Firstly, the multi-step attacks and it's detection methods are presented before the discussion of the designed FRI-based fuzzy state machine model. Additionally, in the seventh chapter, the FRI-based fuzzy state machine model is extended to be implemented in practice, and also demonstrated, and evaluated via application example.

2. Fuzzy System and Fuzzy Rule Interpolation

This chapter first gives a short overview of fuzzy set theory and fuzzy systems, then the fundamentals of fuzzy rule interpolation and some of the methods are presented.

2.1 Fuzzy System

In some computational problems, the crisp set [7] does not always satisfy the needs. There are some application areas such as the IDS system where the two-valued logic and the related binary decision could lead to inefficient solutions (detection) because there are no clear boundaries between intrusion and normal packets [5, 6]. The crisp set grants only membership of 0, or 1. Therefore, the implementation of fuzzy systems has been grown in many application areas. The fuzzy system offers the capability to handle uncertain and imprecise problems and effectively smooths the boundary. The term fuzzy logic was produced by Professor Lotfi Zadeh [8]. Fuzzy logic appears in many successful sophisticated systems in many application areas. It offers several advantages to handle binary decision problems [9]. The fuzzy logic stated in the fuzzy set theory. Fig. 1. shows the crisp set and fuzzy set membership values.

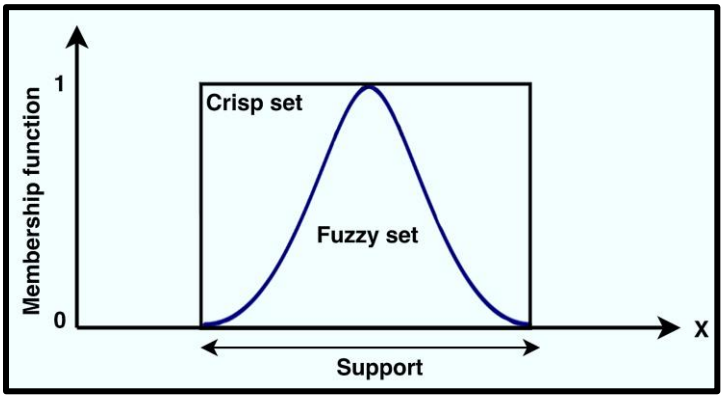

Fig. 1. The Fuzzy Set and Crisp Set

Establishing a fuzzy system (fuzzy inference system) requires several prerequisites:

- Defining the universes of the expected observations (inputs) and the possible output of the fuzzy system.
- Defining the fuzzy partitions for the inputs and outputs of the fuzzy system.
- Generating the required fuzzy rules.

Fuzzy partitions of the input values provide a significant way to define the real input value with each predefined linguistic term [7]. The linguistic terms are variable represented by words or sentences. Fig. 2. shows an example of a linguistic variable Temperature with three linguistic terms Low, Medium, and High. The linguistic values (terms): low, medium, and high are fuzzy sets for the temperature linguistic variables.

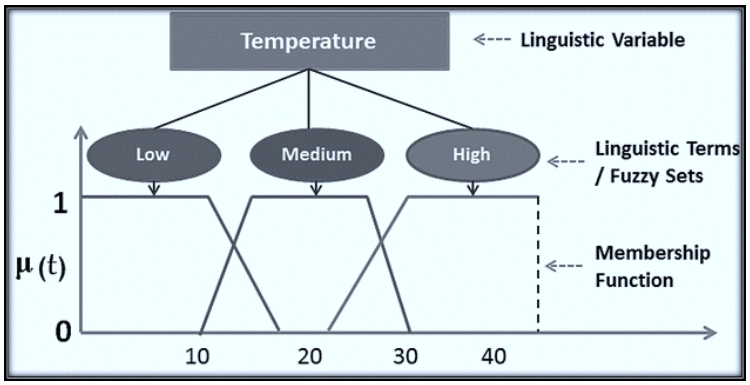

Fig. 2. The linguistic Variable Temperature

The fuzzy sets defining by its membership functions. The membership function fully characterizes the fuzzy set. Practically, the membership functions could be presented using some special shapes. Fig. 3. presents the output of the "mfdemo" script in MATLAB to show the common shapes of the membership functions.

Fig. 3. Membership Functions Example

The approximate reasoning in the fuzzy inference system is through the execution of fuzzy If-Then rules. Firstly, the crisp inputs are transformed into fuzzy sets by the fuzzifier. Then from the fuzzified input, the fuzzy inference system calculates the fuzzy conclusion through the predefined fuzzy rule base. In the end, the resulting fuzzy set is defuzzified. The defuzzified is a process transforms the output fuzzy value into a crisp value [10]. Fig. 4. shows the main structure of the fuzzy inference system. There are some reasoning approaches, the output of the inference system is crisp. Therefore, there is no defuzzification process.

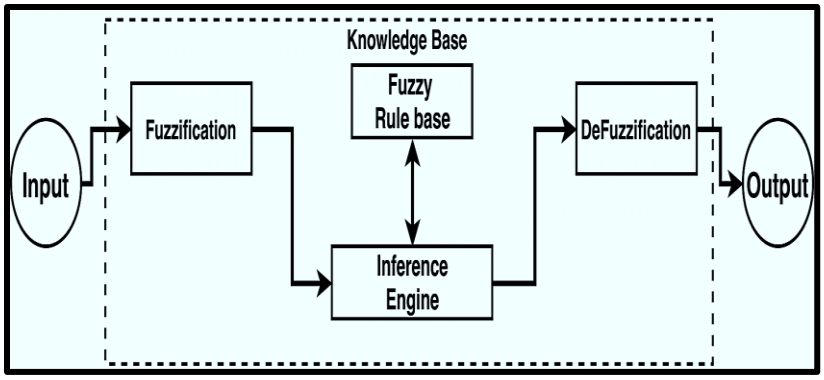

Fig. 4. The Main Structure of The Fuzzy Inference System

In a typical classical fuzzy inference system, the fuzzy rule base is extracted either from expert knowledge or it could be generated from a sample of data. To be able to handle all the possible input values, the fuzzy rule base must cover all the input universes. Therefore, the step of the fuzzy rule base considered as the most critical step during the design of the fuzzy system. Therefore, the fuzzy rule base is the core component in the fuzzy inference system. In case the fuzzy inference system has an adequate number of fuzzy rules that cover all possible input, then it mentioned as a dense fuzzy-rule base. In that respect, the conclusion could be derived directly.

In general, generating a complete (dense) fuzzy rule base in a multidimensional problem is difficult to be implemented because of the lack of information for all the possible fuzzy rules. In the case of missing rule definitions, there could be some observation which is not covered by any of the fuzzy rules. When the number of fuzzy rules is smaller and at the same time does not cover all possible inputs, then it is termed as a sparse fuzzy-rule base. In that respect, the inference engine is not straightforward, and in some cases, the results could not be obtained by the classical inference system. An example related to the density fuzzy rule base:

If X is A1 then Y is B1
If X is A2 then Y is B2
Observation: X is A1

Conclusion: Y = (B1)

From another perspective, an example related to the sparsity fuzzy rule base:

If X is A1 then Y is B1
If X is A2 then Y is B2
Observation: X is A^*

Conclusion: Y = B^* (??)

In the following sub-chapter, the fuzzy rule interpolation presented. The fuzzy rule

interpolation methods had the capability to work in the sparse rule base.

2.2 Fuzzy Rule Interpolation

The classical reasoning methods, i.e. Mamdani [11] and Takagi-Sugeno [12], demands a complete fuzzy rule base to generate the desired output. Therefore, the classical reasoning methods could not infer the conclusion for any observation that is not defined in the fuzzy rule base [13]. if there is a completed fuzzy rule base, then the union of an antecedent part of fuzzy rules was covering all of the input universes as Equation 1 presented.

$$\bigcup_{i=1}^{k} supp(Ai) = X \tag{1}$$

supp refers to support. The support of a fuzzy set indicates the set of all elements within the universe of discourse in which their degree of membership is greater than zero. Further, X_i is the i^{th} input universe of discourse, A_{i_k} is the k^{th} set of the partition of X_i. Fig. 5. illustrates the case of complete rules, when the observation x appeared and was covered by the fuzzy rule base.

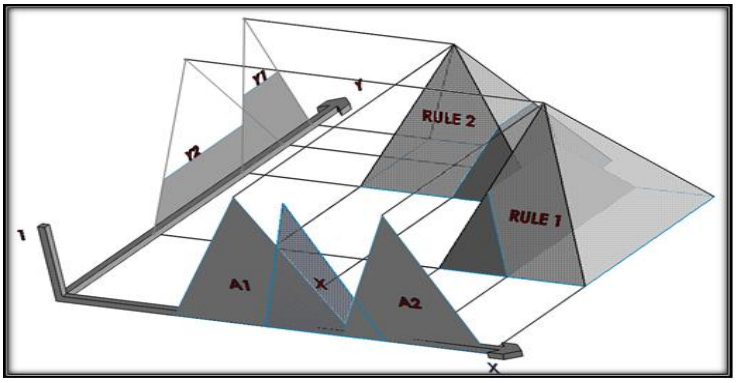

Fig. 5. Complete Fuzzy Rule-base

However, in such application areas that have multidimensional problems, it could be difficult to obtain a dense rule base that covers all possible input parameters. Regardless of the way of obtaining a dense rule base, the rule base size grows exponentially with the number of the observed input parameters. Therefore, a dense rule base in a complex system could be considered as impracticable [14, 15]. For more precise, in the IDS application area, it is challenging to generate a complete fuzzy rule base capable of handling all possible expected observations. As a result, it is imperative to implement a fuzzy concept, especially for the IDS application area, that benefits from extending the binary decision to the continuous space and at the same time can efficiently handle the situation of the sparse rule base.

One effective solution that has the capability to handle a sparse rule base is the FRI methods. The FRI methods were introduced to extend the capability of using the fuzzy system in more complex systems when in case of the sparse rule base. If there a sparse rule base, then the union

of an antecedent part of fuzzy rules was not covering all of the input universes as Equation 2 presented [14]. The $supp$ refers to support, X_i is the i^{th} input universe of discourse, A_i is the set of the partition of X_i, then the union of an antecedent part of fuzzy rules was not covering all of the input universes.

$$\bigcup_{i=1}^{k} supp(Ai) \subset X \rightarrow \bigcup_{i=1}^{k} supp(Ai) = 0 \quad (2)$$

Fig. 6. presents a case when an observation X exists, but not covered by any rules of the fuzzy rule base. Hence, the classical reasoning method could not offer any conclusion; it is a method that is not always capable of satisfying the needs of some application areas (requiring a conclusion for all the possible observations). In some application areas, where there is a large number of unexpected observations and the expert knowledge base cannot cover all the observation domain, it could be difficult to present the complete fuzzy rule base [5].

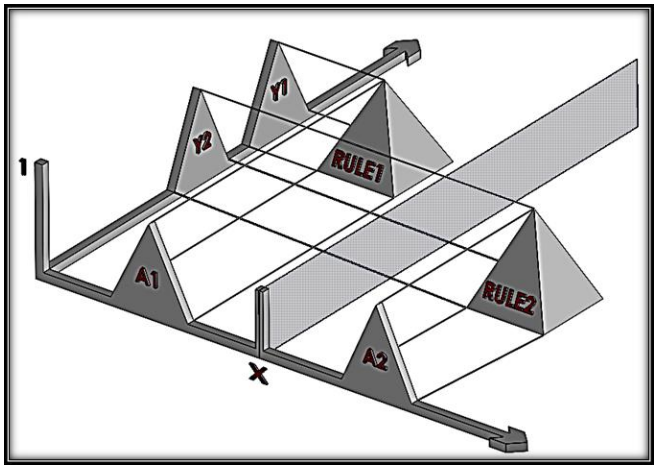

Fig. 6. Sparse Fuzzy Rule-base

Therefore, the FRI methods were introduced to overcome the demand for the complete fuzzy rule base. They generate the possible inference even in cases where there is no complete fuzzy rule base. Furthermore, the FRI methods reduce the number of necessary fuzzy rules which could be beneficial for both decreasing the complexity of the fuzzy system, and the computation time of large systems. Here is a simple example explained the case of sparse rule-base, suppose that there are two rules for a specific fuzzy system as follows:

*[**Rule1:** If you eat over diet Then you become a fat person.]*
*[**Rule2:** If you eat less diet Then you become a skinny person.]*

Applying the classical inference system, there could be an observation between these two rules that have no result.

*[**Input:** if you eat a decent diet]*

This input is followed by a missing fuzzy rule:

*[**Rule*:** If you eat a decent diet Then....]*

Fig. 7. presents the general structure of the fuzzy sparse rule base systems based on the fuzzy rule interpolation.

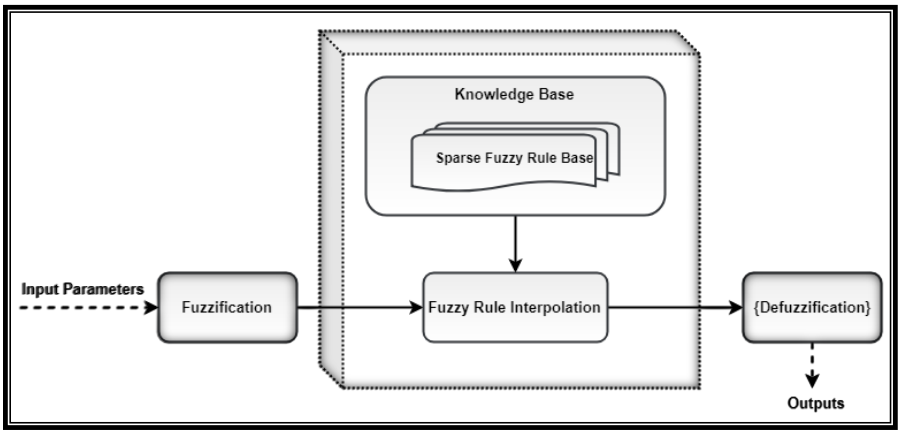

Fig. 7. Fuzzy Rule Interpolation System

There are many FRI methods were proposed in order to handle the sparse rule base, (e.g.[7, 72, 73, 74, 92]). One of the first FRI methods produced by Koczy and Hirota in [7], and it is termed as KH method. The KH method is applicable to work with convex and normal fuzzy sets (CNF). It infers the result using it's α-cuts in such a way that the ratio of distances between the conclusion and the consequent should be identical with the ones between the observation and the antecedents for all important α-cuts. A good summary of FRI methods presented in [15].

2.3 Fuzzy Interpolation based on Vague Environment (FIVE)

The FIVE (Fuzzy rule Interpolation based on Vague Environment) method was originally introduced in [33], [34] and [35]. It serves the deducible conclusions even in case if some situations are not explicitly defined in fuzzy rule-based knowledge representation. Furthermore, it is produced to serve many application areas such as IDS solution, which is served a crisp observation and at the same time required a crisp conclusion. It is worth mentioning that since using the FRI (FIVE) method as an inference engine there is no need for an additional defuzzification step. The concept of the vague environment, introduced by Klawon in [92], refers to the basis of similarity or indistinguishability of the elements as shown in Fig. 8. The concept of the vague environment can be expressed by a scaling function (s). The proper scaling function (s) which describes all the fuzzy sets of a fuzzy partition, should be implemented to produce a vague environment. According to [72, 73], the scaling function (s) is suitable for describing the shapes of all fuzzy set of a fuzzy partition. In the vague environment, the level of similarity between two fuzzy sets illustrates the

fuzzy membership function M(x). Therefore, the most critical task is to find the approximate scaling function.

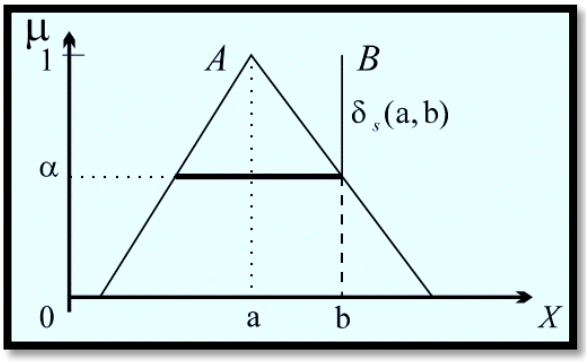

Fig. 8. The α-cuts of μA(x) contain the elements that are (1- α)-indistinguishable from α, where A is α fuzzy set and B is α singleton fuzzy set [35]

The vague environment for the antecedent and the consequent parts can be produced beforehand. This speeds up the method because in course of the inference only the interpolation needs to be done. The FIVE method had the ability to deal with Multi-Input Single Output (MISO) cases, moreover, it is application-oriented because it is fast and easy, and that's why the FIVE method would be used in this work. From another perspective, the lack of fuzziness on the observation part could restrict the applicability of the FIVE method. The FIVE method is implemented by determining the connection between the similarity of two fuzzy sets and the vague distance of points in a vague environment. Then, generating a vague environment from the fuzzy partitions of the linguistic terms within the fuzzy rules. After that, deciding the approximate scaling function. Finally, FIVE method calculates the conclusion by approximating the vague points of the rule base [154].

3. Intrusion Detection System

Regarding the rapid development of technology, the number of intrusions increased and developed continuously. Every day, there are large amounts of financial loss, privacy violations, and information transfers in an illegal way as a result of succeeding intrusions implementation. There are different types of intrusions threatening networks, computer information, and resources. Many types of intrusions exist, such as user to root intrusion, where the goal of this type of intrusion is to have full right permission of computer and network resources. Probing intrusion is another type of intrusion where the goal is to determine the weaknesses of computer and network resources based on scanning techniques. The previous types of intrusions could be implemented to be prerequisite steps of the Denial Of Service (DOS) intrusions. This type of intrusion heading to consume various resources in order to close-down several services for legal users [5].

Governments and organizations employed various systems to protect their data, such as firewalls, IDS, logging systems, vulnerability assessments, anti-virus software's and so on [16]. Commonly, the firewall system is widely used to protect the network, they are adapted the filtration process to detect unwanted external packets. Practically, firewalls did not handle all types of intrusions, especially for the internal packets. The attackers could be from the outside network or legitimate users of the network. Therefore, there is an increasing need to add another security layer that had the ability to detect different types of intrusions regardless if it is internal or external packets. [17].

The intrusion-detection system is one of the solutions used to detect and prevent any abnormal behavior. It had the ability to detect intrusions within inbound and outbound traffics [16]. The term intrusion [18] defined as any attempt to affect integrity, confidentiality or availability of system resources. An IDS system must be constantly updated with all the latest techniques to deal with intruder attacks in order to preserve service availability. IDS is presented as a monitoring and prevention device, where an IDS collects network traffic and then a pre-processing procedure starts collecting network traffic. Intrusion recognition then starts operating to detect and classify packets, as illustrated in Fig. 9. In the following sub-chapter, different types of IDS are presented briefly.

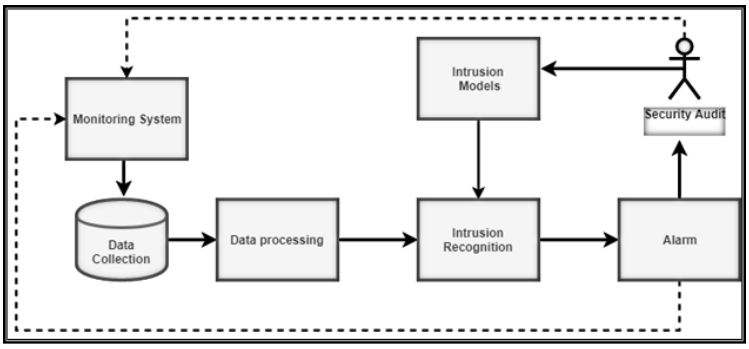

Fig. 9. IDS Detection Processes

3.1 Types of Intrusion Detection System

There are many security solutions could be implemented to prevent a different type of attacks, such as Intrusion Prevention System (IPS) and IDS. The IDS is a motioning system, whereas the IPS is a control system. An IDS is one of the solutions used to prevent intruders from implementing attacks within the protected network. An acceptable IDS can detect a new intrusion in a faster time, comparing with another IDS that required human effort to update frequently the repository of the attack patterns.

An IDS system is organized into two types [19], as follows:

- Host Intrusion Detection System (HIDS): this type of IDS can be implemented on network devices or workstations. HIDS techniques can be used to prevent DDoS attacks on selected devices; a HIDS technique does not support monitoring of the whole network. Fig. 10. presents the HIDS structure.
- Network Intrusion Detection System (NIDS): this type of IDS can be implemented as a security strategy within a protected network. NIDS can be used to detect and classify all network traffic from all devices within a network. Fig. 11. presents the NIDS structure.

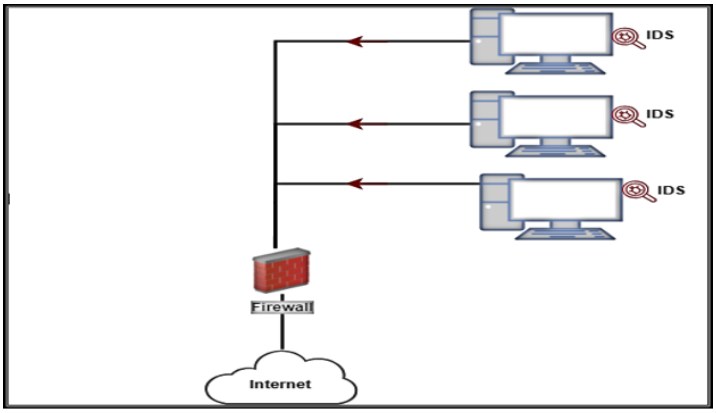

Fig. 10. HIDS Structure

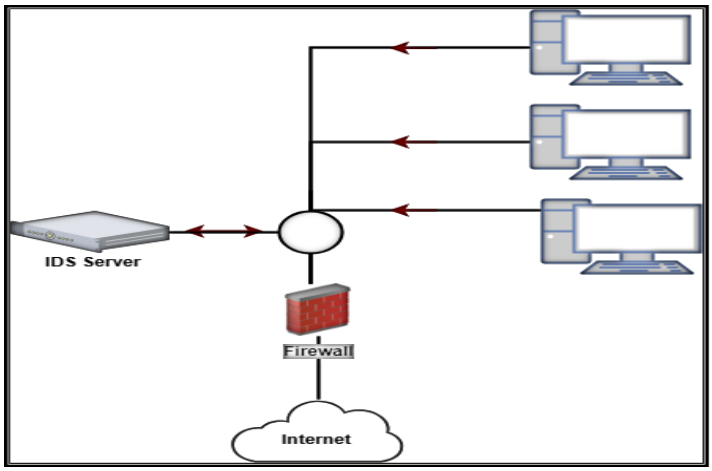

Fig. 11. NIDS Structure

IDS can be applied either anomaly-based or signature-based to detect and classify network traffic. Anomaly-based compares network traffic behavior with historical baseline data so that training data are required for it to work intelligently. Signature-based focuses on each independent packet and compares it with a store of signature or well-known intruder attacks. Detection time for signature-based is faster than anomaly-based, as this requires training data, while signature-based requires a store of signatures [20]. Fig. 12. shows the general IDS classification.

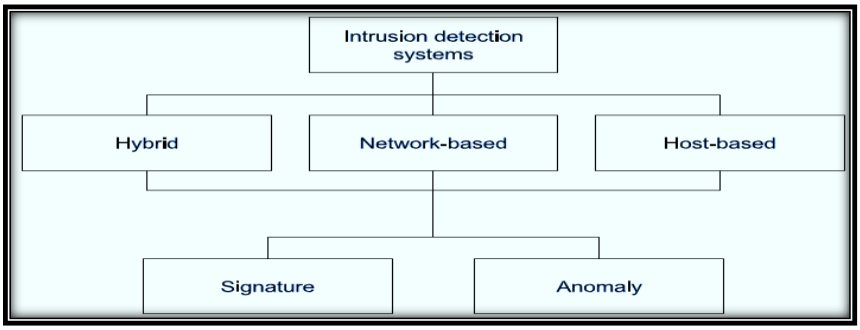

Fig. 12. General Classification of IDSs

3.2 IDS Implementation Techniques

This subsection presents a comprehensive review of IDS detection techniques and also presents briefly an overview of the different types of attacks. The IDS can be defined as a monitoring system built to analyze network communication and detect the intrusions. Any kind of unauthorized activates that caused information damage considered intrusions. Recently, there are many techniques have been implemented to improve the effectiveness of intrusion detection, and also reducing the false alerts. To provide recent IDS detection techniques, the recommendations in the recent IDS surveys [136, 137] are used. IDS detection techniques could be categorized as follows:

- Statistical based technique: the aim of this technique is to build a distribution model that has a normal behavior profile, then detects the potential intrusions. This technique forces on some of the statistical metric such as median and standard deviation of packets within the normal behavior. The main difference between this technique and other techniques that, rather than inspecting network traffics based on flow information, the statistical-based technique is used to identify the normal behavior using statistical metrics. Typically, statistical-based IDS implemented using one of the following models:
    - Univariate: this model used in the case of the data has only one variable. In other words, when there is only one profile of normal behavior [138].
    - Multivariate: on contrary to the univariate model, this model used in the case of the data has more than one variable. Therefore, there is more than one normal behavior of the protected system. This model effectively detects the abnormality if there is a relationship between two normal behaviors [138].

    - Time series model: the main idea of this model is to detect the intrusion based on the series of observations over a certain time. The intrusion appears within this model as a new observation with the probability of occurring at that time is too low [139].
- Knowledge-based techniques: it is also known as an expert system method. The main idea of this technique is to create a normal network profile using the knowledge base. The intrusion appears within this technique as any activity which differs from the normal profile. Contrary to other techniques, the normal profile created as a set of rules based on human knowledge. This technique characterized by its ability to reduce effectively the false positive alerts. From another perspective, IDS based knowledge-based requires frequently updated the rules of normal behavior. Typically, knowledge-based based IDS implemented using one of the following models:
    - The Finite State Machine (FSM): FSM is defined as a computational model. The IDS based FSM presented in the form of states, activities, and transitions. The model checks the state history data to define the intrusion, any variation in the input data could be detected as an intrusion. Therefore, any illegal transition did not follow the standard normal behavior considered as an attack [140].
    - Description Language: The main idea of this technique is to formulate the attack pattern as predefined syntax rules. These rules could be formalized using such kinds of description language such as N-grammars [141].
    - Signature analysis: This technique considers the earliest model applied in the IDS as a detection mechanism. The main idea of this technique relies on string matching. Therefore, the inbound packet is investigated in order to define any matching with the stored attack pattern [142].

- IDS based on Machine Learning (ML) techniques: there are different machine learning algorithms such as Naive Bayes, neural network, clustering, and fuzzy system were implemented in the IDS application area. These algorithms were applied for the sake of discovering the knowledge within intrusion datasets or real-time data. The aim behind adapting the machine learning algorithm for intrusion detection is to generate an IDS that had an acceptable accuracy rate with less requirement for human knowledge. In the following, some of ML algorithms are used to improve the effectiveness of intrusion detection [143].
    - Decision trees: This algorithm could be used to improve the efficiency of intrusion detection. It had three basic elements. The first element is the decision node, which is used to recognize the test features. The second element is a branch, where every branch represents the possible decision. The third element is the leaf which indicates the classes. There are many different decision tree algorithms including ID3 [144].

- Genetic algorithms (GA): GA is a heuristic algorithm for optimization based on the concept of evaluation. Every possible solution presented as a sequence of genes or chromosomes. Adapting the GA in case of intrusion detection requires an encoding process. Commonly there are two types of encoding process: the first is belong to the clustering for the sake of generating the binary chromosome coding. The second coding method is to specify the cluster center using an integer coding chromosome [145].
- Fuzzy logic: this technique is based on adapting the degrees of uncertainty instead of the typical binary decision. Using the fuzzy logic the form of IDS alerts were changed and represented in a more comprehensive form. Also, it permits an instance to belong, possibly partially to multi-class at the same time. Therefore, the administrator will have more choices to reconfigure network baselines. The fuzzy logic could be a good inference method in case of vagueness and problem border between normal and abnormal states. Adapting fuzzy logic decrease effectively the false IDS alerts [146].
- Hidden Markov Model (HMM): HMM is a statistical Markov model in which the IDS detection approach modeled by Markov process. Adapting HMM in the IDS application area required fully understood of the domain problem, which type of intrusion would be detected to extract the requires conditions. Also, the parameter optimization should be applied to optimize the unstructured values. Different parameter optimization algorithms could be applied with HMM such genetic algorithm and Baum Welch algorithm [147].
- K-Nearest Neighbors (KNN): typically, this algorithm is non-parametrically applied in machine learning. The main idea of this algorithm is to name the unlabeled data to the class of its K nearest neighbor. In the IDS application area applying KNN where point X represents an instance of unlabeled data intrusion or normal packet which needs to be classified. Based on the values of K the model considers the packets either as a normal packet or intrusion packet.
- Support Vector Machines (SVM): SVM is a discriminative algorithm defined by a splitting hyperplane. In the intrusion detection, the kernel function is used to map the training data into a higher dimensioned space so that intrusion is linearly classified. There are different types of separating hyperplanes could be used such as linear, polynomial and Gaussian radial function. Typically, in the case of intrusion detection, there are could irrelevant features which affect directly spreading data points to the correct class using SVM. Therefore, SVM applied besides features selection algorithm [148], Fig.13. summarized the recent IDS implementation techniques.

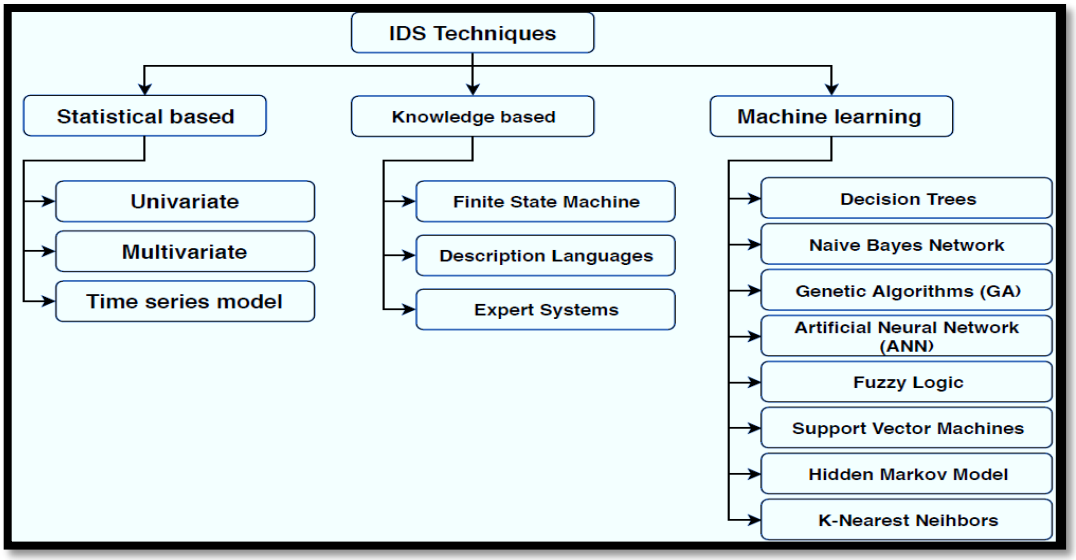

Fig. 13. IDS Implementation Techniques

Protecting computer networks from different types of attacks is not an easy task. Also, implementing an acceptable IDS is not a straightforward procedure. IDSs have to deal with different types of attacks. There are different types of attacks could be categorized based on the aim of attackers, and based on their activities. Each type of attack can be classified into one of the following four classes [149]:

- Denial of Service attack: where the aim of this type of attack is to deny the normal services that are delivered by the computer, network to the end-users.
- Probing attack: the aim of this type of attack to collect the required information about the desired victims such as operating systems, open ports, and IP addresses.
- User to Root: the aim of this type of attack is to have root access or admin access on a specific system or computer application.
- Remote to Local attack: the aim of this type of attack is to obtain privileged which an end-user could have on the computer system.

Different methods and approaches such as fuzzy clustering, genetic algorithm, artificial neural network (ANN) have been conducted on the prevention and avoidance of different types of attacks. Loukas et al. [150] began their research with timeline significant DOS. In September 1996 an "SYN Flood" attack was discovered. This type of attack sends SYN messages to the victim machine, and these SYN messages appear as normal messages. The main idea of the SYN flood attack is that it will never send a final ACK message to the victim server so that the victim server will not accept any new request. Some attackers use a combination of an SYN flood and a PING to stop the server service. Another type of simple DoS attack is the PING attack where the victim

machine is flooded with TCP/ICMP messages. A Smurf attack started in January 1998, where the attackers used ICMP messages to send to the broadcast addresses. This type of attack works by sending an ICMP echo request to the intermediary (slave machine). In most situations, the intermediary (slave machine) does not filter ICMP messages, so that many clients on the network who receive this ICMP echo request send ICMP to replay back.

Furthermore, the teardrop and Bonk techniques started as DoS attacks and focused on Microsoft Windows operating systems. These two types of attacks were successful in disrupting services on Windows operating systems that did not have up-to-date security patches. Teardrop attacks the Internet Protocol (IP) that is required to be fragmented, and where the packets are too large, each fragment packet has a value that will be used later to reassemble the packets. A teardrop attack generates the wrong value to the next fragment so that the packets will not reassemble correctly.

Additionally, the HTTP flood started in 2004. Researchers suggest that complete protection architecture is by means of detection, where detection can be either anomaly-based or signature-based, or a hybrid of these two, which researchers prefer. Classifications such as neural networks, radial basis functions, and genetic algorithms are increasingly used for DOS detection because of the automatic classification that they can offer. The protection system either drops the attacking packets in a timely fashion or renders them inoperable. Traffic rate, SYN and URG flags, as well as some specific ranges of ports, are the most significant in the identification of a DoS attack, as researchers tested in their survey. DDoS attacks are very harmful to OSI layer protocol; each OSI layer serves the type of protocol just as the application layer serves HTTP. The network layer protocol serves ICMP. Sujay et al. [29] emphasize the application-layer DDoS attacks because such attacks are becoming more of a problem and are growing rapidly. The attacks work with a low bandwidth which is very difficult to detect because the attackers act as legitimate users. The major type of DDoS in this layer is the HTTP flood. Sujay and others categorized flooding attacks on the application layer as follows:

- Reflection-based flooding attacks where attackers send a request to the slave machine that acts as a reflector to send a flood based on domain name system (DNS) amplification.
- HTTP flood attack where attackers consign an HTTP GET or HTTP POST request to the victim server and that causes server overload.

Sujay et al. [29] conclude that the Naïve Bayes (NB) algorithm provides faster learning/training speed than other machine learning algorithms. In addition, it has a more accurate rate of classification and detection of layer seven attacks. Lu et al. [151] proposed a framework to efficiently detect DDoS attacks and identify attack packets. The framework exploits the spatial and temporal correlation of DDoS attack traffic. These techniques can accurately detect DDoS attacks and identify attack packets without modifying existing IP forwarding mechanisms at routers. The researchers claimed a 97% detection probability using the proposed framework. Their approach was able to detect different types of DDoS attacks, such as UDP flood, ICMP Flood, DNS flood. After analyzing the network traffic, the proposed framework was stated to have the following advantages:

- The proposed framework is able to detect and identify normal flows.
- Detect and identify the packet without modifying the IP mechanism.
- It only identifies legitimate traffic that will be forwarded.

Another type of attack is the UDP flood. It is one of the most common types of DDoS attacks, where attackers use random ports on the victim machine to flood with UDP packets. Another type of DDoS is an SYN flood attack, where the three-way handshake TCP connections are used. The maximum packet size that can be sent using the Ping command is (65535 bytes). Ping of Death (POD) is a type of DDoS attack that is more harmful because the intruder can send larger than the maximum allowed packet size (65535 bytes). Reflected attack (RA) is another type of DDoS attack, RA attack works by creating forged packets that will broadcast to all the clients in the network. After infected clients receive messages they will replay as spoof addressed that will route current replay messages to the victim server [137]. Other researchers proposed their detection approaches using fuzzy systems to detect the probing attack which is could be used as a part of the remote to local attack [152, 153]. Typically, this type of attack used to collect the required information about the expected victims. In [152], Naik et al. proposed a detection approach using fuzzy rule interpolation for the purpose of detecting the probing attack. Naik et al. integrate the fuzzy rule interpolation for the reasoning part besides the snort IDS. The aim behind this integration is to extract the required network parameters using snort in the sniffing mode. Experimental results conclude that the proposed approach had the ability to detect the probing attack, moreover, effectively reduced both of the false positive and false negative alerts. Due to the rapid growth of technology, there are many contemporary tools and frameworks could be used to execute different types of attacks [87] such as:

- Trinoo: this tool could be used to execute and launch a UDP flood to the targeted machine. Based on the number of master and slave clients, attackers may use the maximum number of Trinoo controllers.
- Ten tool: this tool could be used to execute and launch an SYN Flood, UDP Flood, and ICMP Flood; the attack is also based on the number of master and slave clients.
- Ten 2K tool: this tool could be used to encrypt ICMP flooding and Smurf attacks.
- Stacheldraht: this tool could be used to execute a TCP flood, a Smurf attack, and ICMP flooding.
- Trinity: this unencrypted tool can be used to implement TCP flooding, and it is a method of communication.

3.3 IDS Performance Metrics

An IDS generates an alarm in case of intrusion is appeared, this alarm forwarded to the network administrator for providing a better understanding of the current network security status. Practically, not every alarm generated by IDS considered as an attack. In some cases, IDS

generates a large number of false alarms, where there are normal traffics and appeared as intrusion traffics. In this case, administrators face stressful environments with an overload of different types of alarms. These alarms need to be analyzed and investigated to detect abnormalities. The performance of an IDS could be evaluated using the overall detection rate and the false alarm. The overall detection rate indicates the total number of successfully detection process, and the false alarm indicates the total number of the failure detection process [21]. The acceptable IDS, which is obtained the lower false alarm besides higher detection rate.

The most important items in the output result of the IDS, one of which is the confusion matrix [22]. The confusion matrix contains information about real and predicted traffics as carried out by the IDS. The performance metrics of such systems are commonly evaluated using the information in the matrix [5, 19, 23]. Table 1 shows the successful detection process alarms and failure detection process alarms for the IDS based on the confusion matrix. where TP, FP, TN, and FN are true positive, false positive, true negative and false negative respectively.

Table 1. Hit and Miss IDS Confusion Matrix

	Predicate Class		
Actual Class		**Attack**	**Normal**
	Attack	TP (Hit Rate)	FN (Failure Detection)
	Normal	FP (Miss Rate)	TN (Successfully Detection)

Commonly, the main performance metrics that could be extracted from the confusion matrix are as follows [19]:

- Average Accuracy (AA): this is a primary indicator, which calculates average accuracy for all experiments. Equation 3 shows how AA is measured:

$$\text{AA} = \frac{TPR+TNR}{TPR+TNR+FPR+FNR} \tag{3}$$

- True Positive Rate (TPR): this is another primary indicator to calculate the proposition of detecting a network intrusion as an intrusion (correctly classified). Equation 4 shows how TPR is calculated:

$$\text{TPR} = \frac{TP}{TP+FN} \tag{4}$$

- True Negative Rate (TNR): this is an indicator for the proposition detecting normal packet as not an intrusion. Equation 5 shows how TNR is calculated:

$$\text{TNR} = \frac{TN}{FP+TN} \tag{5}$$

- False Positive Rate (FPR): this is a performance indicator for detecting a normal network packet as an intrusion packet (incorrectly classified). Equation 6 shows how FPR is calculated:

$$\text{FPR} = \frac{FPR}{FPR+TNR} \tag{6}$$

- False Negatives Rate (FNR): this is a performance indicator detecting an intrusion as normal network activity. Equation 7 presents how FNR is calculated:

$$\text{FNR} = \frac{FNR}{TPR+FNR} \tag{7}$$

- Precision (P): this is the primary performance indicator for a number of records that are correctly classified, as follows: intrusion packet as intrusion packet, normal packet as a normal packet [24]. Equation 8 illustrates how P is calculated:

$$\text{P} = \frac{TPR}{TPR+FPR} \tag{8}$$

- Root mean squared error (RMSE): this provides information on the efficiency that indicates the difference between the outputs and the targets. Lower values of the RMSE indicate more evaluation that is accurate. Zero means no error.
- F-Measure: this is a combined measure for recall and precision and it is calculated using Equation 9 below:

$$\text{F} - \text{Measure} = \frac{2TP}{2TP+FP+FN} \tag{9}$$

4. IDS Model-based Data Mining Algorithms

This chapter first gives a short overview of data mining algorithms, then the IDS model-based data mining algorithms are presented, and the evaluation of IDS model-based data mining algorithms are described in detail, which is incorporated in the work presented further on.

4.1 IDS Model-based Data Mining Algorithms

4.1.1 Artificial Neural Network (ANN)

ANN was the oldest detection method used with an IDS system [25], and the following types of ANN were implemented with the IDS: ANN supervised learning, ANN unsupervised and hybrid ANN. Each IDS system includes three main common components:

- Data collection module.
- Analyzer module.
- Response modular.

The most common and most well-known Feedforward Neural Network (FFNN) model is called the Multi-Layer Perceptron (MLP). MLP has been successfully applied in a number of applications, including regression, classification and time series prediction problems, using simple auto-regressive models. MLP permits the data flow to travel one way, from input to output. There is no feedback; it tends to be straight-forward networks that companion inputs with outputs. MLP permits the data flow to travel one way, from input to output. There is no feedback; it tends to be straight-forward networks that companion inputs with outputs.

According to [26, 27], any MLP network can be distinguished by a number of performance characteristics, which can be summarized in three points:

- Neural Network Architecture: Overall, MLP architecture can be clarified as the pattern of connections between the neurons in different layers. The architecture consists of three layers: the input layer, hidden layers, and the output layer. Two nodes of each end-to-end layer are connected. Furthermore, MLP is always fully connected. Each link has a weight, which is limited based on the training algorithm. Architectures that are more complex have more layers.
- Training Algorithm: The method of selecting one model from a set of models, which determines the weights of the connections.
- Transfer Function: This is applied by each neuron to its net input to determine its output signal. This function is usually non-linear. The sigmoid function is one of the most commonly used transfer functions. The use of the sigmoid function has an advantage in neural networks trained by a backpropagation learning algorithm.

According to [135] the MLP network advantages:

- MLP had the ability to work with incomplete data.
- MLP had the fault tolerance, therefore, the corruption of one or more cells of ANN doesn't prevent it from generating the result.

- Parallel processing capability.
- Information is storing in the entire network.
- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience

However, one of the limitations of the MLP that the determination of the suitable network structure, there are no pre-defined rules for determining the artificial neural networks. Also, it suffers from presenting the problem task to the network. Furthermore, the behavior of the network is unexplained.

4.1.2 Random Forest Algorithm

Random forest algorithm was developed by Leo Breiman and Adele Cutler in [28], combining tree algorithms to predict new unlabeled data. The predictor depends on the number of (A) that represent the number of trees in the forest; the attributes are selected randomly, and each number of trees represents a single forest, and each forest represents a prediction class for new unlabeled data. In this algorithm, random feature selection will be selected for each individual tree [29].

A random forest algorithm ensemble learning algorithm for detection of the abnormality is based on an individual number of trees [23]. Using the random forest algorithm, many trees will be generated, and each individual tree is constructed by different parts of the general dataset. After each tree detects an unlabeled class, the new object will be implemented and each tree will vote for a decision. The forest chosen as the winning class is based on the highest number of recorded votes. The random forest algorithm summarized as follows:

- If there is a dataset, we need to split n samples from the whole dataset, giving (n samples= number of trees).
- Each dataset sample needs to be regressed or classified; for each record, this is randomly split among all predictor classes to reach an approximately optimal split. Bagging can be learned as a special scenario when m (tries) = P (number of predictors).
- Predict unlabeled classes based on a reassembled number of aggregation prediction numbers of trees.

The accuracy rate and error rate for random forest tuning parameters can be measured either by splitting the whole dataset, for example by testing 40% and for training 60%, or by dividing the data into 10s or 20s, etc. After a random forest built model test, 40% can be used to calculate error rate, and accuracy rate can be measured based on comparisons of correctly detected instances with incorrectly detected instances [30]. Out of the bag (OOG) is another way of calculating the error rate in this technique; there is no need to split the dataset because calculation occurs within the training phase. The following parameters need to be adjusted correctly to reach the highest accuracy rate with a minimum error rate:

- The number of trees.
- The number of descriptors that occur randomly for the present candidate's *m* (tries).

After analyzing and studying many cases, 500 trees are needed within the descriptor that may be desired. Even if there are great numbers of trees that will not achieve the highest accuracy rate, except for wasting training time and resources [24], random forest tuning parameters are a hot research area that needs to be fine-tuned. Fig. 14. shows random forest architecture.

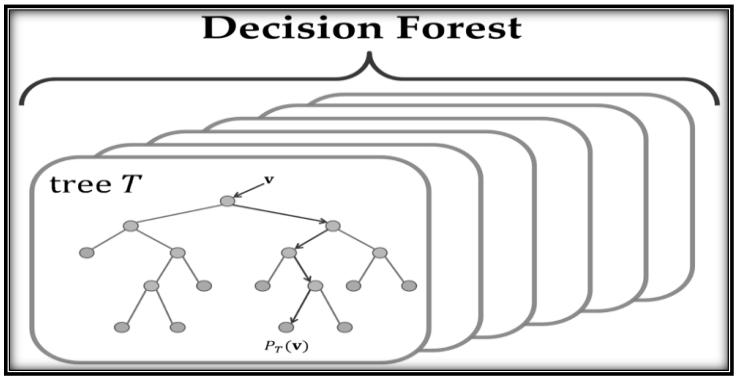

Fig. 14. Forest Tree Architecture

Advantages of the random forest:

- Random forest could be used for both classification and regression problems.
- Random forest is suitable to be used with a large dataset with higher dimensionality.
- Random forest could be used in case of the incomplete dataset where there is missing data.
- Random forest deals with a sampling of training data with a replacement called bootstrap sampling. A third of data (validation).

One of the limitations of the random forest algorithm that it doesn't give the precise continuous nature prediction in case of the regression problem. Also, it acts as a black-box approach.

4.1.3 Naive Bayes Algorithm

Naive Bayes is a simple probabilistic algorithm that returns p (y | x), and calculates probabilistic for each class in a dataset and determines discriminative learning to predict the values of the new class. The main formulation for Naive Bayes may be found in [31]. A Naive algorithm links the dataset attributes x ∈X that are used as inputs to the class labels Z ∈ 1,2, C, where X is the attribute space and Z is the class space. Let X = IRD where D is a real number. The Naive classifier may be used with discrete and continuous attributes. This model is called a multi-label problem. The learning function that directly computes class is called a discriminates model. The main aim is to learn the conditional class that is used for non-linear and multi-label problems. The advantages of the Naïve Bayes algorithm could be summarized as follows:

- It could be implemented easy and fast.
- Naïve Bayes required less training data to build an acceptable model.
- Naïve Bayes handles the issues of discrete and continuous data.

- Irrelevant features had less effect on the Naïve Bayes model.
- Probabilistic predictions.
- Naïve Bayes could be used for both binary and multi-class problems.

One of the limitations of the Naïve Bayes that, it assumes the attributes are independent of each other. However, in some cases features depend on each other. Another limitation point that, it makes a very strong assumption on the shape of data distribution. Additionally, it suffers from zero conditional probability problems.

4.1.4 J48 Tree Algorithm

J48 tree first introduced by Breiman in [132]. It is the most common classifier used to manage database for supervised learning that gives a prediction about new unlabeled data, J48 creates Univariate Decision Trees. J48 based used attribute correlation based on entropy and information gain for each attribute [12]. It has been used in many fields of study, such as data mining, machine learning, information extraction, pattern recognition, and text mining. It has many advantages; it is capable of dealing with different input data types: numeric, textual and nominal. The J48 decision tree is an extension of the algorithm ID3. It has an advantage over ID3 in that it can build small trees. It follows a depth-first strategy and a divide-and-conquer approach. A decision tree consists of several elements: root, internal nodes, and leaves. The internal nodes represent the conditions in which the value of the parameters will be tested. Based on these values and the condition, the flow of the tree will be decided (along which branch the decision tree must go). Leaf nodes represent the decision or the class. Fig. 15. shows a typical decision tree structure.

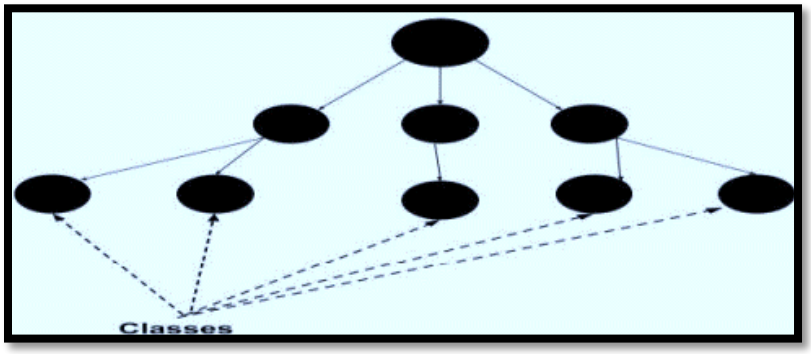

Fig. 15. J48 Structure

The tree is constructed by following these three main steps:

- Ensure that all of the grouped inputs are of the same class. Then ensure that the tree is labeled with the class.
- Calculate some parameters for each attribute, such as information gain.
- Choose the best split attribute based on the criteria that have been set. Entropy comes from information theory; it indicates the amount of information that is held.

The main advantages of the J48 algorithm could be summarized as follows:

- J48 could be easily interpreted by humans as rules.
- J48 could be implemented simply and easily.
- Addressing non-linearity.
- Construct the tree is fast and also it has a fast prediction process.

One of the limitations of the J48 is overfitting. It is sensitive to the precise layout of point, and if there is less data, j48 can fit the noise of data. J48 is not suitable to be implemented with a smooth boundaries dataset. It effectively works with the discontinuous pice-wise model.

4.1.5 Bayesian Network

It is a classifier for supervised learning that uses assumptions of independent features. It uses the theory of learning that represents the distribution naïve Bayesian classifier. It uses various search algorithms and different quality measurement methods. Bayes Network is an enhancement for Naïve Bayes [133]. A Bayesian network is very useful because it helps us to understand the world we are modeling. BayesNet may be the best in various areas of life, we're modeling a mysterious fact and in the state of decision nets, wherever it is good to make intelligent, justifiable and quantifiable decisions that will enhance the performance of classification. In brief, BayesNet is helpful for diagnosis, prediction, modeling, monitoring, and classification [134]. The main idea of the Bayesian classifier consists of two phases: In the first, if an agent has an idea and knows the class, in this case, it can predict the values of the other features; in the second, if the agent does not have an idea or does not know the class, in this case, the Bayes rule is used to predict the class given. The aims behind using the Bayesian Network as a classifier are:

- Probabilistic learning, which calculates clear probabilities for the assumption.
- Incremental, which is a prior knowledge and possible to be added to data viewing.
- Probabilistic prediction, which can predict more than one hypothesis, weighted by the probabilities.

The advantages of Bayesian network could be summarized as follows:

- it could easily handle incomplete datasets.
- It adapts learning the causal relationship.
- It readily facilitates the use of prior knowledge.

One of the limitations of the Bayesian network that it is extremely computationally expensive. Furthermore, Bayesian networks tend to perform poorly on high dimensional data. Bayesian network models could be difficult to interpret. In the following sub-chapter, several experiments have been performed and evaluated to assess various data mining algorithms as IDS model detection.

4.2 Intrusion Detection Using Data Mining Algorithms

Information technology had a rapid development in the last two decades. Computer networks are widely used by industry, business and in various fields of human life. Maintaining the reliability of networks became an essential task of the IT administrators. On the other hand, rapid development also produces several challenges and the question of network reliability became a very difficult task. There are many types of attacks threatening the availability, integrity, and confidentiality of computer networks. The Denial of Service attack (DoS) considered one of the most common harmful attacks.

The aim of DoS attacks is to temporarily deny services for the end-users. In the most common case, it consumes the network resources and overloads the system with undesired requests. For this reason, the DoS acts as a large umbrella of naming for all types of attacks that aim to consume computer and network resources. In 2000 Yahoo was the first victim of a DoS attack, which was also the date, when the DoS recorded its first public attack [32]. Nowadays web services and social websites are the main targets of DOS attacks [33]. From another vulnerability perspective, the remote to local (R2L) attacks are another common type of attack which are designed to gain local access permissions remotely in case if some network resources (e.g. servers) are protected by allowing access only for local users. There are several types of R2L attacks e.g. SPY and PHF. These types of attacks aim to prepare illegal remote access to network resources [34].

Related to the illegal access to the network and computer resources, the type of User to Root (U2R) attacks aim to switch the attacker access permission from a normal user to the root user, who has full access rights to the computers and network resources [35]. The main challenge is that attackers are always keeping up-to-date their tools and techniques for exploiting any kind of vulnerabilities appearing to be known. Hence, it is very difficult to detect all types of attacks based on single fixed solutions. For that Intrusion Detection System (IDS) became an essential part of network security. It is designed to monitor the network traffic and generate alerts when any attacks appear. IDS can be implemented to monitor network traffic of a specific device (host IDS) or to monitor all the network traffics (network IDS) which is the most common type used.

Conceptually there are two types of IDS, Anomaly-based IDS, and Misuse based IDS. Anomaly-based IDS implemented to detect attacks based on the recorded normal network behavior. It compares the current real-time traffics with the previously recorded normal traffics. This type of IDS is widely used because it has the ability to detect the new (previously unknown) type of intrusions, too. On the other hand, conceptually it registers the largest values of false-positive alarms too, for the situations, which is normal, but not recorded among the "normal network behavior" samples (e.g. there is an uncommonly large number of normal packets considered to be attacking traffic).

Misuse intrusion detection systems are implemented to detect attacks based on a repository of attack signatures. Conceptually it has no false positive alarms but a new type of attack (which signature is missing from the repository) can succeed to pass-through as normal traffic. According to [36], attack detection considered a classification problem because the target is to clarify whether

the packet either a normal or an attack packet. Therefore, an IDS can be built based on the methodology of machine learning algorithms.

To compare the IDS performance of different data mining algorithms, the following algorithms have been selected: J48, Random Forest, Random Tree, Decision Table, Multi-layer Perceptron (MLP) and Naive Bayes algorithm. For the model formation and evaluation, the publicly available KDD-99 benchmark dataset was applied. The studied attack types were DOS, R2L, U2R, and PROBE.

4.2.1 IDS and Benchmark Dataset

The commonly available KDD-99 is the data set used at The Third International Knowledge Discovery and Data Mining Tools Competition [37] for the task of building a network intrusion detector. The competition was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining in 1999. Although the KDD-99 dataset is rather old, it is still widely used in academic research for testing and comparing IDS performance [38]. Because of its unceasing popularity, for comparing and discussing the IDS performance in case of different intrusion types in this work also the KDD-99 dataset is chosen. In [33], Nguyen et al. made a deep survey of IDS and the KDD-99 dataset. They extracted 49596 instances of the KDD-99 dataset to implement several machine learning algorithms e.g. Naive Bayes and MLP. The authors succeeded to propose two models for detecting intrusions types of the KDD-99 dataset. In [39], Lahre et al. applied a MATLAB implementation of the Support Vector Machine (SVM) algorithm for IDS. They used the KDD-99 dataset as IDS benchmark data. They claimed that the SVM algorithm needs long training time and hence the usability of SVM is limited. In [40], Haddadi et al. preprocessed the KDD-99 dataset, symbolized and normalized the attributes to the [-1, 1] range. Then a feed-forward neural network was applied in two experiments. They concluded that the neural network is not efficient for detecting R2L and U2R attacks but it has an acceptable accuracy rate in detection DOS and PROBE attacks.

In [31], Zhang et al. are implemented Fuzzy ARTMAP, Radial-basis Function, Backpropagation (BP) and Perceptron-back propagation-hybrid (PBH) IDS. The four algorithms evaluated and tested on the KDD-99 dataset, in which the BP and PBH algorithms achieved the highest accuracy rate. Another research direction focuses on attributes selection algorithms in order to reduce the cost of the computation time. In [41], Alsharafat et al. are focusing on selecting the most significant attributes to design IDS that have a high accuracy rate with low computation time. They implemented the IDS based on extended classifiers and neural networks to reduce false positive alarm as much as possible.

In [42], Bhargava et al. implemented the information gain algorithm to be an effective attribute selection method for improving DoS intrusion detection. The genetic algorithm (GA) was also implemented to enhance the detection of different intrusion types. In [34], Paliwal et al. proposed a methodology to derive the maximum detection rate with the minimum false positive rate. The GA was applied to generate a number of effective rules to detect intrusions. They achieved 97%

accuracy on the KDD-99 dataset. In [43], Fleizach et al. applied the Naive Bayes algorithm to detect all intrusions types of the KDD-99 dataset. The authors concluded that the detection rate is unacceptable if they apply only a single IDS algorithm. Some IDS research is focusing on a specific type of attack. In [44], Alkasassbeh et al. produced a new Distributed Denial of Service (DDoS) dataset from the samples of HTTP, smurf, SiDDoS, and UDP attack data. The DDoS dataset then tested with different IDS algorithms. For detecting the DDoS intrusions, the MLP algorithm achieved the highest accuracy rate (98.36%). Another example of applying the KDD-99 dataset for evaluating different IDS methods can be found in [45], where the performance of 20 different classifiers was compared to different attack categories. Regarding the implemented experiments the Multivariate Adaptive Regression Splines (MARS) algorithm getting a higher accuracy rate. Furthermore, the fuzzy logic obtained an acceptable accuracy rate compared with other implemented algorithms. Moreover, the lowest accuracy rate recorded by Partial Decision Tree (PART) algorithm. Additionally, The acceptable IDS should perform with an accepted average accuracy rate and lowest possible false negative value.

The KDD-99 dataset still provides a reasonable benchmark environment for testing and evaluating various machine learning algorithms. It is also important to note, that a single machine learning algorithm could not provide an acceptable detection rate. One solution to this problem is the application of different IDS algorithms for detecting the various type of attack threats. In the following, seven types of Machine Learning and Data mining Algorithms (J48, Random Forest, Random Tree, Decision Table, MLP, Naive Bayes, and Bayes Network) will be implemented, tested, compared and evaluated based on KDD-99 dataset. Our interest is directed to the most important performance parameters, like false negative and false positive attack detections. We would like to select the most promising IDS methods which could achieve an acceptable accuracy rate with the minimum false negative detections.

4.2.2 Preprocessing the KDD-99 Dataset

The KDD-99 dataset can serve as a good sample for several intrusion behaviors, and a good benchmark for testing and evaluating intrusion detection algorithms. The KDD-99 dataset first published by the MIT Lincoln labs at the University of California in 1999 and still available in UCI Machine Learning Archive [46]. It includes 4898431 instances with 41 attributes. The first step of the IDS model generation is the preprocessing of the dataset. For this reason in our case the KDD-99 dataset was first imported to an SQL server 2008, then various statistical measurement values e.g. distribution of instances records, attack types and occurrence ratios were calculated. Fig. 16. presents the main preprocessing steps of the KDD-99 dataset.

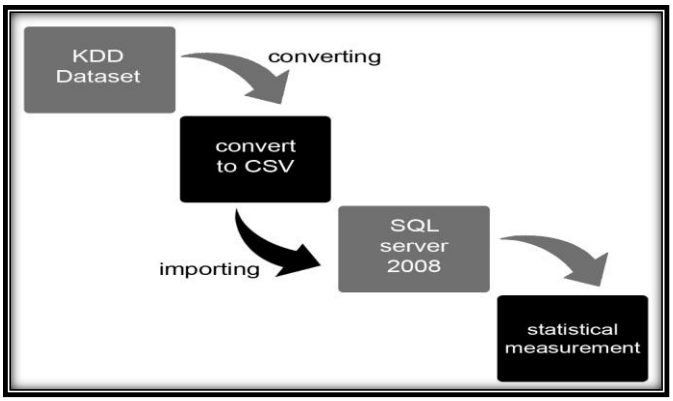

Fig. 16. Preprocessing steps of the KDD-99 dataset

Statistical measurements provide a deep understanding of this dataset in order to extract impartial experiments. Table 2 illustrates the distribution of attack types within the KDD-99 dataset. There are 21 types of attacks, which can be categorized into four groups with a different number of instances and occurrences. 79% of the instances are related to DOS attacks, 19% belong to normal packets and 2% can be categorized as other attack types. Based on these values the KDD-99 appears to be an unbalanced dataset. The packets have 41 attributes.

Table 2. The distribution of the attack types within the KDD-99 Dataset

Categories of Attack	Attack name	Number of instances
DOS	SMURF	2807886
	NEPTUNE	1072017
	Back	2203
	POD	264
	Teardrop	979
U2R	Buffer overflow	30
	Load Module	9
	PERL	3
	Rootkit	10
R2L	FTP Write	8
	Guess Passwd	53
	IMAP	12
	MulitHop	7
	PHF	4
	SPY	2
	Warez client	1020
	Warez Master	20
PROBE	IPSWEEP	12481
	NMAP	2316
	PORTSWEEP	10413
	SATAN	15892

These attributes are basic information that can be collected during the TCP/IP connection [35]. Table 3 illustrates these fundamental TCP /IP attributes. One important contribution of the KDD-99 dataset, that it also contains 32 experts suggested attributes which can help the understanding of the behavior of an attack type. I.e. the most significant attributes of the four attack groups (DOS, R2L, U2R, and PROBE) are also included.

Table 3. The fundamental attributes of a TCP /IP connection

Attributes	Type
The total duration of connections in second	continuous
The total number of bytes from sender to receiver.	continuous
Total number of bytes from receiver to sender	continuous
Total number of wrong fragments	continuous
Total number of urgent packets	continuous
Protocol type	discrete
Type of service	discrete
The status of the connection (normal or error)	discrete
Label (1) if the connection established from the same host. Otherwise label (0)	discrete

4.2.3 The Applied Data Mining Algorithms

Data mining algorithms can be categorized as supervised and unsupervised algorithms [47]. Supervised algorithms learn for predicting the object class from pre-labeled (classified) objects. The unsupervised algorithm finds the natural grouping of objects given as unlabeled data. In our IDS study supervised learning algorithms will be applied, as the imported KDD-99 dataset includes predefined classes.

J48 Classifier: This classifier is designed to improve the implementation of the C.4.5 algorithm, which is introduced by Ross Quilan [48] in 1993. The output of this classifier is in the form of decision binary trees, but with more stability between computation time and accuracy than the original C.4.5 [49]. The decision about the expected output is the leaf node of the decision tree structure. Decision table classifier: the main idea of this classifier is to build a lookup table for identifying the predicted output class. There are several algorithms e.g. breadth-first search, genetic algorithm, and cross-validation can be implemented to generate an efficient decision table [50]. The lookup table includes a set of conditions and the expected classes. These are the rules of the decision table classifier, which are predicting the classes for the incoming inputs [51]. The rules of the decision table can also be fuzzified, this case the decision table classifier can also handle uncertainties of the inputs and classes.

Multi-layer Perceptron (MLP) Classifier: MLP is one of the most common algorithms that proved its effectiveness to deal with several application areas e.g. time series classification and regression problems [52]. During the implementation, the testing phase can be short, but the training phase typically needs a long time. MLP algorithm can be implemented with various

transfer functions e.g. Sigmoid, Linear, and Hyperbolic. During the implementation the number of outputs or expected classes is straightforward, but the number of the hidden layer neurons should be correctly defined for having an effective MLP classifier. In the beginning, every node within the neural network had its random weight and bias values, the large weight values in the input layer present the most effective attributes within a dataset, and on the contrary, the small weight values present the least effective attributes within a dataset.

Naive Bayes Classifier: This classifier refers to the group of probabilistic algorithms. It implements Bayes theorem for classification problems. The first step of the Naive Bayes classifier is to determine the total number of classes (outputs) and calculate the conditional probability for each dataset class. After that, the conditional probability is calculated for each attribute. The standard formula of Naive Bayes can be found e.g. in [31]. Furthermore, it has the ability to work with discreet and continuous attributes too. On the contrary to the MLP classifier, Naive Bayes can be implemented within a short period of time [43]. The Naive Bayes Classifier can be represented as a Bayesian Network (BN) or a Belief Network. BN presents independent conditional probabilities based on the understanding framework. In general, BN is an acyclic graph between expected class (output) and a number of attributes [53].

Random Tree Classifier: It is one of the classification tree algorithms. A random tree classifier is a finite group of decision trees. The number of trees must be fixed in advance. Each individual tree represents a single decision tree. Each individual tree has randomly selected attributes from the dataset. The entire dataset is predicted from several decision trees outputs and choose the winner expected class based on total numbers of votes [54]. Random Forest Classifier: It is one of the ensemble learning algorithms. The main goal of this algorithm is to enhance tree algorithms based on the concept of the forest. Random forest algorithms [28] have an acceptable accuracy rate. It can be implemented to be able to handle noise in the dataset. It is averaging multiple decision trees, trained on different parts of the same dataset. The number of trees must be fixed in advance. Each individual tree within a forest predicts the expected output. Then, the expected output selected by a voting technique [28].

4.2.4 Generating the Classification Systems

There are 21 types of attacks appearing in the KDD-99 dataset. These attacks are categorized into four groups (DOS, R2L, U2R, and PROBE). Each attack type has a different number of instances and occurrences in the dataset. After the preprocessing of the KDD-99 dataset, and because it includes a large number of instances only 148753 instances of records have been extracted to an SQL server. This number of instances (148753) present 10% of each attack type. This labeled data serves as a training set for the further IDS model generation. The attack categories and types with the number of instances are presented in Table 4. Based on the analysis of the KDD-99 dataset the occurrence distribution of the different attack types was recorded. 79% of the extracted data present DOS attacks, 19% is related to the instances of normal traffic and 2% is related to other types of intrusions (U2R, R2U, and PROBE). The KDD-99 dataset includes

different features such as duration, source byte, and destination contains high variations, as a result, the performance of the studied algorithms could be degraded. Therefore, at the early stage of preprocessing, the KDD-99 dataset was normalized using the WEKA filter preprocessing stage, this filter normalizes all values in the given dataset to the default range of [0.0, 1.0]. It is worth mentioning that the KDD-99 features (41) were taken into consideration, in other words, the implemented models were generated using all KDD-99 features (41).

Table 4. The Training Model Dataset.

Categories of Attack	Attack name	Number of instances
DOS	SMURF	85983
	NEPTUNE	32827
	Back	70
	POD	10
	Teardrop	30
U2R	Buffer overflow	10
	Load Module	2
	PERL	1
	Rootkit	5
R2L	FTP Write	2
	Guess Passwd	10
	IMAP	4
	MulitHop	2
	PHF	1
	SPY	1
	Warez client	31
	Warez Master	7
PROBE	IPSWEEP	382
	NMAP	70
	PORTSWEEP	318
	SATAN	487
Normal		28500

The experiments were performed on an Ubuntu 13.10 platform, Intel R, Core(TM) i5-4210U CPU @ 1.70GHz (4CPUs), 6 GB RAM. The applied machine learning tool was the Waikato Environment for Knowledge Analysis (WEKA) [55]. It is an open-source tool written in JAVA and available for free. It provides all the classifiers referred to this work. These are the J48, Random Forest, Random Tree, Decision Table, Multilayer Perceptron (MLP), Naive Bayes and Bayes Network. Based on the preprocessed 148753 instances according to the labeled attack categories (see attack types and categories in Table 4.). For the creation of the classifiers, all the labeled data were processed as training data. In order to evaluate the generated models using different validation processes, the sample splitting and the 10-fold cross-validation process were used. Then the classifiers were saved for a comprehensive study introduced in the following. In the sample splitting technique, the dataset is usually split into training data and test data. The

training set contains a known output and the model learns on this data in order to be generalized to other data later on. The test dataset (or subset) was used in order to test the generated model prediction on this subset. Fig. 17 shows the structure of the sample splitting technique.

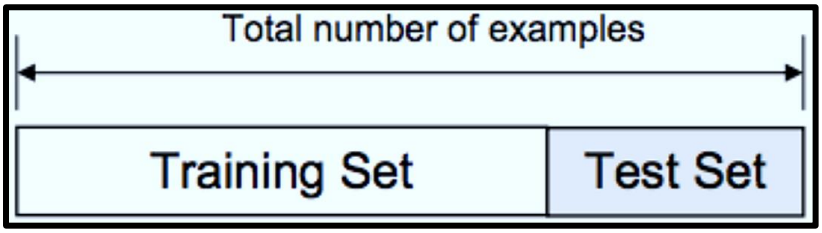

Fig. 17. Sample Splitting Strcutre

4.2.5 Performance of The IDS Implementation

After the IDS model generation, the next step is the comparative study of the models. In order to implement a fair testing phase fully randomized 60000 instances have been extracted from the preprocessed database for the sample splitting validation process. Regarding the cross-validation process (148753) instances were used for train, test and validate the generated models. The extracted testing data included all the 21 attack types of the KDD-99 dataset and labeled according to the attack categories introduced in Table 4. There are several metrics that can be used for evaluating the efficiency of the IDS model. In this work, the confusion matrixes were generated for each classification algorithms. Furthermore, the following performance metrics [44] were computed:

- True Positive (TP): This value represents the correct classification of the attack packets as attacks.
- True Negative (TN): This value represents the correct classification of the normal packets to be normal traffic.
- False Negative (FN): This value represents an incorrect classification, where the attack packet classified as a normal packet. A large FN value presents a serious problem of confidentiality and availability of network resources because the attacker succeeded to pass through the IDS.
- False Positive (FP): This value represents incorrect classification, where the normal packet classified as an attack. The increasing of FP value increases the computation time, but it is considered less harmful than the increased FN value.
- Precision: Is one of the primary performance indicators. It presents the total number of records that are correctly classified as attacks divided by a total number of records classified as an attack.

In addition, the number of both correctly and incorrectly classified instances is recorded with respect to the time taken for the proposed training model. During the testing phase, the following classification parameters were applied:

- J48 tree classifier: confidence factor = 0.25; numFolds = 3; seed = 1; unpruned = False, collapse tree = true and subtree rising =true.
- Random forest classifier: number of trees =100 and seed =1.
- Random tree classifier: min variance = 0.001 and seed = 1.
- Decision Table classifier: Best First Search (BFS) and cross value = 1.
- MLP classifier: search learning rate=0.3, momentum =0.2, validation threshold=20.

Table 5 presents the TP rate and the Precision values of the selected data mining algorithms during the experiments for both validation processes. It can be concluded that in both validation processes (sample splitting and cross-validation), the random forest classifier achieved the highest TP rate, and the random tree classifier achieved the lowest TP rate. I.e. the random tree classifier has the lowest correct attacks classification value. Furthermore, in the case of the sample splitting validation process, the decision table classifier reached the lowest 94.4% precision value. This indicates that the decision table classifier suffers from an increasing false positive value. Therefore, there is a large number of normal packets classified as attack packets.

Table 5. The True Positive Rate and the Precision

Classification Algorithms	Validation Process			
	Sample Splitting	Cross-Validation	Sample Splitting	Cross-Validation
	TP	TP	Precision	Precision
J48	0.931	0.992	0.989	0.991
Random forest	0.938	1	0.991	1
Random tree	0.906	0.987	0.992	0.991
Decision table	0.924	0.990	0.944	0.991
MLP	0.919	0.993	0.978	0.992
Naive Bayes	0.912	0.990	0.988	0.995
Bayes Network	0.907	0.997	0.992	0.991

In general, the TP rate and precision values are important performance parameters for a common intrusion detection system, but from another perspective, the most serious performance parameters are the FP rate and the FN rate. The goal of this study is to decrease both of these parameters, as much as possible, especially the FN parameters. The FP and FN performance parameters of the IDS tests are summarized in Table 6. It can be concluded, that the random tree classifier achieved the highest 0.093 FN rate in case of a sample splitting validation process. Meanwhile, the random forest classifier obtained zero false alerts in case of a cross-validation process, that's means, it effectively detected the normal packets. Hence there is a large number of attacks classified as a normal packet. On the contrary with the decision table classifier which is achieved the lowest 0.002 FN rate. At the same time, the decision table classifier reached the highest 0.073 FP rate. It means that there is a large number of normal packet classified as attack packets.

Table 6. The False Positive Rate and the False Negative Rate

Classification Algorithms	Validation Process			
	Sample Splitting	Cross-Validation	Sample Splitting	Cross-Validation
	FP Rate	FP Rate	FN Rate	FN Rate
J48	0.005	0.008	0.063	0.006
Random forest	0.001	0	0.061	0.008
Random tree	0.001	0.004	0.093	0.02
Decision table	0.073	0.01	0.002	0.006
MLP	0.014	0.004	0.066	0.01
Naive Bayes	0.002	0.003	0.085	0.02
Bayes Network	0.001	0.014	0.092	0.011

Table 7 presents the Root Mean Square Error (RMSE) and area under the Receiver Operating Characteristic (ROC). RMSE presents the difference between the actual and the desired outputs based on the confusion matrix. The model with a lower value of RMSE indicates better output prediction efficiency, on the contrary, the large value of RMSE indicates lower prediction efficiency. The ROC represents a trade-off between detection rate and false alarm. It shows the probability of detection provided by the IDS at a given false alarm probability. The large value of ROC indicates that the model has better intrusion detection ability, while the lower value presents the weakness of the model.

Table 7. The RMSE and the Area under the ROC

Classification Algorithms	ROC Area	Root Mean Squared Error
J48	0.969	0.0763
Random forest	0.996	0.0682
Random tree	0.953	0.0763
Decision table	0.984	0.0903
MLP	0.990	0.0813
Naive Bayes	0.969	0.0872
Bayes Network	0.997	0.0870

It is worth mentioning that, the area under ROC could be obtained based on calculating the AUC curve on the basis of integrating the areas of small trapezoidal bins from the ROC curve. The basic principle is that observations segment the entire integration interval into multiple sub-intervals. Each subinterval will form a closed area [155]. The trapezoidal rule formula is as follow:

$$AUC_{(t_i - t_{i-1})} = (t_i - t_{i-1}) X \frac{f(t_i) + f(t_{i-1})}{2}$$

According to the results in Table 8, the Bayes network classifier achieved the highest 0.997 ROC value, while the random tree classifier achieved the lowest 0.953 value. Furthermore, the

random forest classifier had the lowest 0.0682 RMSE value, while the decision table presented the highest 0.0903 value. After the classification of 60000 instances of the KDD-99 dataset, the total number of incorrectly classified records for each selected classifier, and the average accuracy rate is presented in Table 8. The average accuracy rate is calculated according to Equation 11.

$$Average\ Accuracy\ Rate = \frac{TP+TN}{TP+FN+FP+TN} \quad (11)$$

Table 8. Average Accuracy Rate

Classification Algorithms	Sample Splitting	Cross-Validation
	Accuracy Rate	Accuracy Rate
J48	93.10%	98.9118 %
Random forest	93.77%	99.1845 %
Random tree	90.57%	97.1097 %
Decision table	92.44%	98.3005 %
MLP	91.90%	98.1208 %
Naive Bayes	91.23%	97.9664 %
Bayes Network	90.73%	97.5099 %

It is important to mention, that it could take a long time to build the IDS model. Based on the experiments, the random tree classifier model is the fastest, while training the MLP classifier was taken about 176 minutes. In our experiments, it was the longest model generation time. From the results of the tests, we can conclude the followings:

- The Random forest achieved the highest accuracy rate with the smallest RMSE value and false-positive rate in both of the validation processes.
- The Random tree classifier reached the lowest average accuracy rate with the smallest ROC value for both validation processes.
- Regarding the average accuracy rate, there is no big difference between the MLP classifier and the Naive Bayes classifier in case of a sample splitting validation process.
- All classification algorithms present acceptable precision rates for detecting normal packets for both validation processes.
- Bayes network classifier recorded the highest value for detecting correctly the normal packets.
- There are no big differences between the MLP and the J48 algorithms based on FN parameters.
- Despite that, the decision table classifier did not reach the highest accuracy rate, but it had the lowest FN rate. The model generation time was also acceptable.
- All of the tested classification algorithms had acceptable model generation time, except the MLP.
- It can be concluded that the rule-based algorithm (decision table) is presented with an acceptable accuracy rate with the lowest FN rate, which is increasing the confidentiality

and the availability of the network resources.

4.3 Summary

The practical task of information reliability and security is effective intrusion detection and prevention. Open systems are vulnerable. Intruders are always keeping updated information about the current technology and generate new intrusion methods. There are several defense solutions against intrusions. In this chapter, the commonly available KDD-99 dataset was used for comparing and discussing the IDS performance in case of different intrusion types. The IDS performance of the J48, Random Forest, Random Tree, Decision Table, Multi-layer Perceptron (MLP) and Naive Bayes algorithms compared based on the average accuracy rate, precision, false positive and false negative performance in case of DOS, R2L, U2R, and PROBE attacks.

According to our experiments, from 60000 randomly chosen testing records, the random forest algorithm achieved the highest 93.77% accuracy value. During the same test, it has 3735 incorrectly classified records. The random tree algorithm achieved the lowest 90.57% accuracy value with 5655 incorrectly classified records. Regarding the root mean squared error values, also the random forest algorithm achieved the lowest 0.0682 value, while the decision table algorithm had the highest 0.0903 value. The Naive Bayes algorithm needed the shortest model generation time, while the MLP algorithm reached the longest 176 minute training time. All the seven studied algorithms achieved acceptable precision for detecting normal packets. The decision table algorithm had the lowest 0.002 false-negative value, which means that it can detect various intrusion types of the KDD-99 dataset successfully. The effectiveness of any IDS always suffers from false negative values. The acceptable IDS should perform with the lowest possible false negative value.

The scientific results in this chapter were published in [56, 57, 58] for in-depth details.

5. Investigating The Capabilities of The FRI in The IDS Application Area

This chapter presents the capabilities to use FIVE based fuzzy rule interpolation model in the IDS application area, in the design and implementation of the novel detection mechanism for Distributed Denial of Service (DDOS) attacks. Firstly, the recent IDS model-based classical fuzzy inference system is presented briefly before the discussion of the developed FRI-IDS (introduced in: [5]).

5.1 Classical Fuzzy Inference System For IDS

This subsection presents some relevant works related to the application of the fuzzy system for intrusion detection. It also provides a brief overview of different methods and approaches that are used for intrusion detection.

There are different challenges in implementing sufficient IDS. One of these challenges is the binary decision in detection techniques. The typical detection mechanisms of IDS had a boundary problem [6]. The fuzzy system offers several advantages to handle boundary problems. It also presents the detection degree level of intrusions which could be more readable for the security engineer. In [59], Idowu et al. implemented architecture to detect DDOS attacks using Fuzzy Reasoning Spiking Neural-P (FRSN-P). It is a type of membrane computing system. The neurons within this system communicate based on electrical spikes (impulses). The Knowledge Discovery Databases (KDD-99) dataset imported into the proposed system. KDD-99 dataset was prepared by the University of California [60]. The constraint was on a synchronization flood. The authors extracted the synchronization flood attack from the KDD-99 dataset. After extracting the required records of the desired attack, the fuzzy reasoning spiking neural-P was implemented and evaluated. According to the presented results, the proposed system was able to reach 0.02% false-negative and 0.25% false-positive detection rate.

In [61], Thakare et al. proposed a network IDS based on automatic fuzzy rule base generation. The single length of frequent item approach was implemented as a preprocessing step to generate the required fuzzy rule base automatically. KDD-99 dataset was used to evaluate the proposed system. The implemented experiments demonstrated that the proposed system obtained 90% as an overall accuracy rate. The work of Mkuzangwe et al. in [62], focuses on detecting and preventing the Neptune attack. It is a type of TCP flood attack and belongs to DOS attacks. The enhanced release of the KDD99 dataset (NSL-KDD) was imported to test and evaluate the proposed fuzzy system [35]. According to the implementation of the proposed fuzzy system, a feature ranking algorithm was implemented to select the relevant features for the detection approach. The proposed fuzzy system was evaluated with a decision tree algorithm using the NSL-KDD dataset. It was succeeded to obtain 0.93% as the overall average accuracy rate for detecting Neptune attack. Compared with the decision tree algorithm, the proposed fuzzy system had the highest average accuracy rate.

There are many IDS methods implementing feature selection algorithms. A large number of features can be collected using several network tools. However, not all of them are relevant to the

detection mechanism. The primary aim of feature selection algorithms is to reduce the computation time by reducing the size of data. It can reduce the problem space by reducing the required feature set to a minimum [63].

In [6], Shanmugavadivu et al. proposed a network IDS based on the fuzzy system. The feature selection algorithm was applied for reducing the originally large (41) number of features of the KDD-99 dataset. The authors divided the imported dataset into two parts, the first part for the training phase and the second part for the testing phase. The fuzzy rules generated based on the instructions of a knowledge expert. The proposed fuzzy system was able to achieve a 0.95% average accuracy. Other algorithms were also used in combination with a fuzzy system to detect and prevent intrusions. In [64], Danane et al. proposed a hybrid approach of genetic algorithm and fuzzy system for detecting DOS attacks. The main purpose of implementing a genetic algorithm was the automatic fuzzy rule generation, as a preprocessing step of the IDS construction. The testbed environment was the KDD-99 dataset. The proposed system achieved a 0.94% average detection rate.

There are other hybrid fuzzy IDS solutions too. In [22], Wang et al. combined the neural network and fuzzy system to enhance the detecting rate of intrusion detection. The fuzzy rule base is generated based on the expert knowledge base. The neuro-fuzzy IDS achieved a 0.93% average detection rate. In [65], Feizollah et al. proposed a combination of a fuzzy system and a decision tree algorithm. Features selection was applied to reduce the size of feature space. Moreover, the decision tree algorithm was applied to extract automatically the fuzzy rule base. The proposed system reached a 0.99% average accuracy rate. In [66], Einipour et al. focused on enhancing the detection rate of IDS based on combining the fuzzy system and Particle Swarm Optimization (PSO) method. The PSO was applied for generating the fuzzy rule base in order to detect DDOS attacks. As a result of the PSO generated fuzzy rules, the proposed fuzzy system was able to reach 0.93% as a detection rate average.

The past works provided convincing contributions and supporting the idea that the fuzzy rule-based model could be a useful device for IDS implementation. In the following sub-chapter, we introduced the benefits of the FRI application at the IDS application area, mainly tackling the distributed denial of service type attacks. The aim behind using the FRI is the simplification of the expert rule base and the extension of the binary decision problem to continuous truth value, in which conclusions like ”the level of the attack” can be also simply defined.

5.2 Fuzzy Rule Interpolation in The IDS Application Area

Fuzzy rule interpolation methods can serve deducible (interpolated) conclusions even in case if some situations are not explicitly defined in fuzzy rule-based knowledge representation. This property can be beneficial in partial heuristically solved applications; there the efficiency of expert knowledge representation is mixed with the precision of the machine learning method. The implementation of IDS Model-based fuzzy rule interpolation was divided into three main steps:

- To identify observable features suitable for IDS and the way they can handle the intrusion

boundaries problem.

- To implement the FRI-IDS model as a detection mechanism for DDOS attacks.
- To compare the FRI-IDS model with other literature's results, which had used the same test-bed environment with different classification algorithms for detecting DDOS attacks.

5.2.1 DDOS Attacks Possibilities

The DDOS attack is treated as one of the most harmful types of attacks. Nowadays, DDOS attacks are considered as a continuous challenge for both users and organizations. The first serious DDOS attack appeared in 2000 against Yahoo [32]. The main purpose of DDOS attacks is to consume different types of resources such as network bandwidth, CPU, memory utilization, etc. Any consumption of these resources will increase the overloading and as a result, different services would be unavailable for legal users.

In 2017, according to the Kaspersky security report [67], the approximate cost of the DDOS attack was 52000$ for small businesses and around 440,000$ for enterprise businesses. In 2016, the dangerous effect of DDOS reached 80 countries around the world [68]. There different types of (DDOS) attacks such as smurf attack where an intruder sends large numbers of Internet Control Message Protocol (ICMP) echo packets to the intended victim. In most situations the intermediary (slave) machine does not filter ICMP messages, therefore, many clients on the network who receive this ICMP echo request send ICMP to replay back. Another type of DDOS attack is the User Datagram Protocol (UDP) flood attack. It is one of the most common types of DDOS attacks, where the intruders send a large number of UDP traffics to the victim within a specific period of time. From another perspective, the HTTP-flood attack is considered as a difficult one to detect. According to HTTP-flood attack, the intruder sends completely normal posted messages with a very slow rate in a systematic way, this type of DDOS is difficult to detect because its behavior seems like a normal behavior [19]. Another type of modern DDOS attack is a Simple Query Language (SQL) Injection Distributed Denial Of Service (SIDDOS). According to this type of DDOS, the intruder inserts a malicious SQL statement as a string in the browser side, then it is forwarded to the victim as an executed statement [69].

5.2.2 DDOS Dataset

There are several open-source datasets that exist which include intrusions related data. These datasets provide a convenient environment for research purposes. In this work, the DDOS dataset of [19] was used as a test-bed environment for testing the FRI inference based IDS solutions. The DDOS dataset includes intrusions related data, such as HTTP flood and SIDDOS. This dataset can be downloaded freely for research purposes from [19]. The distribution of the recorded attack types within the DDOS dataset summarized in Table 9.

Table 9. Distribution of DDOS Dataset Attacks

Attack	Number of Records
SIDDOS	6665
HTTP Flood	4110
UDP Flood	201344
SMURF	12590

The discrete and continues features appearing in the DDOS dataset are presented in Table 10 and Table 11 respectively.

Table 10. The Discrete Features of DDOS Dataset

Index	Features	Description
6	Pkt Type	Packet Type Based on Used Protocol
8	Flags	7-Digit Flag Strings
11	Node Name From	Client Name That Sends the Packet
12	Node Name To	Client Name That Receives the Packet
28	Pkt Class	The Class of Packet

Table 11. The Continues Features of DDOS Dataset

Index	Features	Description
1	Src Add	Port of Source Address
2	Des Add	Port DestinationAddress
3	Pkt Id	Packet Identifier
4	From Node	Define client sending packet
5	To Node	Define client receiving packet
7	Pkt Size	The Packet Size in bytes
9	Fid	Flow Identifier
10	Seq Number	Sequence Number
13	Number Of Pkt	Total Number of Packets
14	Number Of Byte	Total Number of bytes
15	Pkt In	Total Time of Packet Inside Queue
16	Pkt Out	Total Time of Packet Outside Queue
17	Pkt R	Time of Packet Received
18	Pkt Delay Node	Time Packet Delay Within Node
19	Pkt Rate	Average Packet Rate
20	Byte Rate	Average byte Rate
21	Pkt Avg Size	Average Packet Size
22	Utilization	Bandwidth Utilization
23	Pkt Delay	Total Time of Packet Delay
24	Pkt Send Time	Time of Sending Packet
25	Pkt Received Time	Time of Receiving Packet
26	First Pkt Sent	Time of FirstPacket Sent
27	Last Pkt Received	Time of the Last Packet Received

5.2.3 Dataset Preprocessing And Features Selection

The DDOS dataset consisted of a large number of connection records. Because of that, we extracted 10% of the total number of intrusions records. The extracted DDOS dataset listed in Table 12.

Table 12. The Extracted DDOS Dataset

Class Name	Number of Connection Records
SIDDOS	676
HTTP Flood	441
UDP Flood	20135
SMURF	1260
Normal	193000

The typical IDS detects the packet based on predefined rules such as SNORT [70]. The rules which are responsible for distinguishing intrusion from normal packets must be created based on the features given in the intrusion dataset. There is a large number of features could be recorded during the collection of the intrusions dataset. These features could be recorded using any network monitoring tools. Generally, most of them are not relevant features. It means that features are not relevant in the detection of a given type of intrusion. Features selection considered an important step because if there are irrelevant features then it could decrease the performance of the final IDS.

One of the simple and quick features selection algorithms is the Information Gain (IG) algorithm. In this work, the IG algorithm was chosen as a features selection algorithm because of its simplicity and quick validation of the entropy factors. The IG algorithm is based on the concept of entropy. According to [71], the entropy parameter computed to characterize the relevantly of each feature. Suppose that, $E = (E, P)$ be a discrete probability space, where $E = \{E_1, E_2, \ldots, E_n\}$ is the finite set of the selected features. Each of the selected features had the following probabilities $P_i, i = 1,2, \ldots, n$. The entropy for each features computed as Equation 12 illustrated.

$$Entropy(E) = -\sum_i p_i \log_2(p_i) \tag{12}$$

After the entropy parameters calculated for each feature of the DDOS dataset, the next step is to calculate the IG values. IG computed based on the predefined collected entropy parameters and the set of all possible values for the feature, Equation 13 presents the calculation formula of IG.

$$\mathrm{IG} = \mathrm{Entorpy(E)} - \sum \frac{\mathrm{E_k}}{\mathrm{n}} \mathrm{Entorpy(E_k)} \tag{13}$$

Where n presents the total number of instance of records, E_k denotes the total number of instance of records that belongs to the class k. Table 13. summarizes the top ten relevant features using the IG algorithm.

Table 13. The Relevant Features Using IG Algorithm

No	Features	IG Values
1	Pkt Rate	0.3811300
2	Byte Rate	0.3809683
3	Utilization	0.3809683
4	Pkt Size	0.3803146
5	Pkt Avg Size	0.3786557
6	Number Of Pkt	0.3740723
7	Pkt Delay	0.3630512
8	Number Of Bytes	0.3630512
9	First Pkt Sent	0.3513514
10	Pkt Delay Node	0.3321921

5.2.4 FRI-IDS Model Generation

In this subsection, the full architecture of the FRI IDS model discussed including its main functions and interactions. Typically, in the classical fuzzy system, the inferring of consequences could not be deduced in case if some situations were not explicitly defined in a fuzzy rule-based. Therefore, the inferring of the consequences of the fuzzy system required a completed fuzzy rule base. In the case of the sparse rule base which was not covered all of the possible situations, FRI methods offer the capability to generate the possible inference, even in case of lack definitions and information of existing knowledge representation. This benefit could be beneficial in partial heuristically solved applications. The FRI-IDS model was a description of the problem domain (IDS application area). It constraints of the key features (relevant features) to detect the intrusion. There are four major components needed to implement the FRI-IDS model as a detection mechanism:

- Setup the input and output of the FRI-IDS model. (The input parameters of the FRI-IDS model were the relevant features of the dataset which are mentioned in Table 13, the output of the FRI-IDS model supposed to be the level of attack instead of binary decision).
- Setup the fuzzy sets for each input/output of the FRI-IDS model. (This component introduced in details in subsection (5.2.6)).
- Setup the fuzzy rules for all the possible events of normal and intrusion. (This component introduced in details in subsection (5.2.6)).
- Testing and validating the FRI-IDS model.

The inference engine of the FRI-IDS model was performed by the Fuzzy Interpolation based on the Vague Environment method (FIVE). It was introduced by Kovacs in [72, 73, 74] in 1996. The FIVE method serves the deducible conclusions even in case if some situations are not explicitly defined in fuzzy rule-based knowledge representation. It is produced to serve many application areas such as IDS solution, which is served a crisp observation and at the same time required a crisp conclusion. It is worth mentioning that since using the FRI (FIVE) method as an inference engine there is no need for an additional defuzzification step.

The architecture of the FRI-IDS model was shown in Fig. 18. starts by data filtration phase where the network traffics (training data) analyzed in order to extract and determine the relevant features. During the data filtration phase, the irrelevant features were removed. It should be known that the existence of irrelevant features could decrease the performance of the FRI-IDS model because it could give an incorrect indication about the existence of attacks. In the modeling phase, the sparse fuzzy model identification [75] was performed, it was introduced by Johanyák in 2008. The modeling phase had several actions, this includes the estimation of fuzzification and membership functions, fuzzy rules generation besides deducing the consequences and tuning methods.

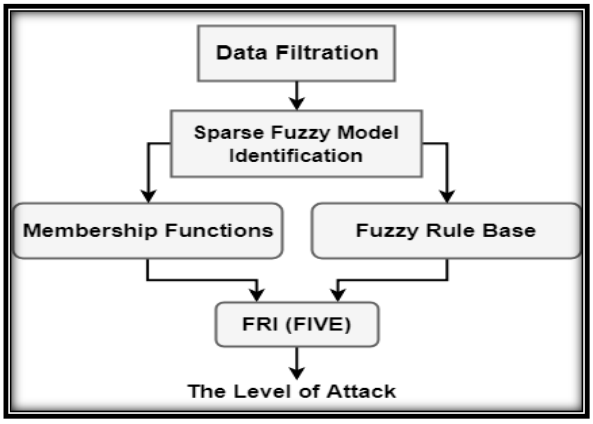

Fig. 18. The Architecture of The FRI-IDS Model

5.2.5 Data Filtration Phase

The dataset was divided into a training part and testing part. The training data consisted of 10000 records with 5000 normal cases and 5000 intrusion cases. The test data consisted of 10000 records with 5000 of normal cases and 5000 of intrusion cases. In order to increase the efficiency of IDS, it is important to identify the observable features that are relevant to detect intrusions from the network traffic data [59].

These are some of the relevant features: utilization, packet rate, byte rate, pkt size, and pkt delay. For the sake of reducing the possible number of fuzzy rules and low complexity system, the highest three relevant features according to the IG algorithm were used as input parameters of the FRI IDS model. These relevant features as found in Table 13 are the packet rate, byte rate, and utilization. The anomaly-based and misuse based detection techniques detect the attacks based on the predefined rule base (i.e. rules for normal and intrusions). For this, sorting the normal and intrusions cases of the training data is required [7]. Algorithm 1 presents the sorting and feature extraction of the training data.

Algorithm 1: Sorting and Feature Extraction

Input: The training data
Input: Two pools of the dataset (normal and intrusion)
The training data two pools of the dataset (normal and intrusion)

1: **while** Termination Condition Not Met **do**
2: Classify whole test-bed dataset into" normal" and "attack" class
3: Check for missing entry for all records
4: Extract the suitable features for IDS based o IG algorithm
5: Remove all irrelevant features
6: Store the values of normal pool
7: Store the values of intrusions pool
8: **end while**

The outputs of sorting and feature extraction algorithm were two pools of normal and intrusion records. These two pools consisted of only the relevant observable features that are suitable for the FRI-IDS model (packet rate, byte rate, and utilization) features, where all other values are removed.

5.2.6 Modeling Phase

The part of fuzzy modeling considered as one of the important parts of the fuzzy system. The FRI-IDS model was constructed by using the sparse fuzzy model identification [75]. The training data of the FRI-IDS model had three input parameters (packet rate, byte rate, and utilization). These parameters were chosen according to the IG algorithm to infer the DDOS attack. In order to generate the optimized fuzzy rules and fuzzy sets, the Rule Base Extension using Default Set Shapes (RBE-DSS) method which is introduced by Johanyák in [76] was applied. According to [76, 77] the main steps of RBE-DSS method can be summarized as follows:

- In the early stage of modeling the fuzzy system, the RBE-DSS method generates two rules that covered (fit) the minimum and maximum of the output.
- In the next step, the hill-climbing tuning algorithm started. It is adjusting the previous parameter values one by one. For each iteration, the fuzzy system is evaluated with different parameter values based on training data. The retrieved parameter values ensure that the fuzzy system belongs to the better performance index for the later iterations.
- The performance index is computed in each iteration to compare the obtained results with different parameter values. The relative root main square error was chosen as a performance index for tuning the FRI-IDS model.
- On the assumption, the increasing of fuzzy system performance appeared too slow or interrupted (i.e. fuzzy system obtained the local minimum) then, the new fuzzy rule generated to increase the possibilities of fuzzy system enhancement.
- The new fuzzy rule created where the difference between the value of actual output and computed output is maximum.
- The tuning process stopped either in case if the predefined performance index value

obtained or when the number of the iterations is reached.

As a result of applying the RBE-DSS method, Fig. 19. shows the support of antecedent fuzzy sets of the tuned FRI-IDS model based on the training data.

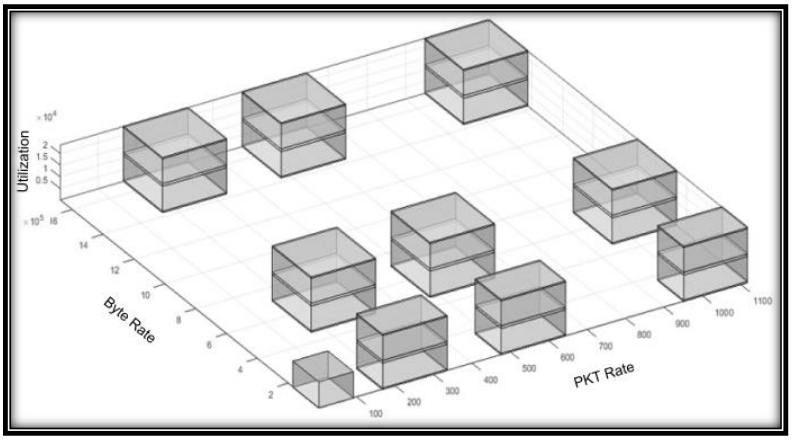

Fig. 19. Support of The Antecedent Fuzzy Sets of FRI-IDS Model

It is worth mentioning that, the generated fuzzy rules by the RBE-DSS method were sparse, these fuzzy rules for the fuzzy interpolation and if it is implemented for classical fuzzy reasoning there is no result could be obtained. Out of 28 fuzzy rules were generated in order to detect the DDOS attack based on the training data. Table 14 presents the generated fuzzy rules, Subsequently, of the modeling phase, the obtained fuzzy sets were represented by trapezoidal membership functions. The byte rate and utilization input parameters have three trapezoidal membership functions and the packet rate input parameter has four trapezoidal membership functions. Table 15 presents the optimized values of fuzzy sets for the FRI IDS model based on the training data.

Table 14. The Obtained Fuzzy Rules

No.	Packet Rate	Byte Rate	Utilization	Consequences
1	L	L	L	FA
2	L	L	M	FA
3	L	L	H	FA
4	L	M	L	FA
5	L	M	M	FA
6	L	M	H	FA
7	L	H	L	A
8	L	H	M	A
9	L	H	H	A
10	M	L	L	FA
11	M	L	M	FA
12	M	L	H	FA

No.	Packet Rate	Byte Rate	Utilization	Consequences
13	M	M	L	FA
14	M	M	M	FA
15	M	M	H	FA
16	M	H	L	A
17	M	H	M	A
18	M	H	H	A
19	H	L	L	A
20	H	L	M	A
21	H	L	H	FA
22	H	M	L	A
23	H	M	M	A
24	H	M	H	A
25	H	H	L	A
26	H	H	M	A
27	H	H	H	A
28	VL	L	L	A

Table 15. The Obtained Fuzzy Set Parameters Of FRI-IDS Model

Packet Rate	**Very Low**	**Low**	**Medium**	**High**
	[1 1 35.92 91.78]	[166.81 222.66 278.51 334.36]	[475.73 531.58 587.43 643.28]	[950.67 1006.52 1062.37 1118]
Byte Rate	**Low**	**Medium**	**High**	
	[55 55 83268.03 167136.28]	[461330.38 545198.63 629066.88 712935.13]	[1425835.73 1509703.98 1593572.23 1677420]	
Utilization	**Low**	**Medium**	**High**	
	[3 3 594.18 11235.33]	[594.18 11235.33 12417.68 23058.83]	[12417.68 23058.83 23650 23650]	

FRI-IDS model serves crisp values and at the same time generates a crisp conclusion. Therefore, each observation within the training data in the example of this work presented as a fuzzy singleton. Fuzzy systems had the capability to extend the binary decision to the continuous truth value which is more readable and easier to be understood and analyzed. Suppose that, there are two observations within the training data, the first observation had the following crisp values (packet rate= 200, byte rate = 55943 and utilization = 11560). The second observation had the following crisp values (packet rate= 900, byte rate = 1190251 and utilization = 22029). The inferred consequence of the FRI-IDS model for the previous two observations was shown in Fig. 20. and Fig. 21. respectively where the first observation presents the normal event and the second observation shows the DDOS attack event. FRI-IDS model can serve the interpolated conclusions even in case if some observations are not covered directly by fuzzy rules as Fig. 21. presented. The interpolated conclusions mean that the FRI-IDS offers the ability to generate the required comprehensive alert, and present it based on the FRI-IDS output membership functions (Normal or DDOS attack).

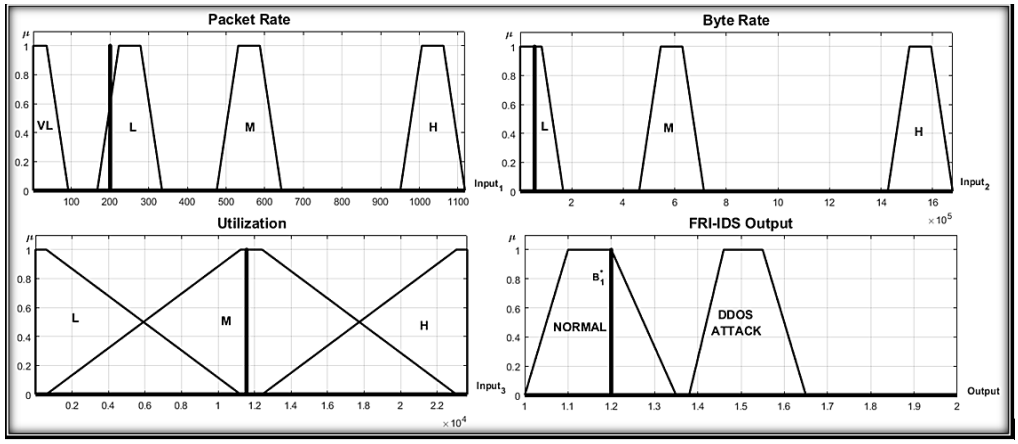

Fig. 20. FRI-IDS Output Response in Case of Normal

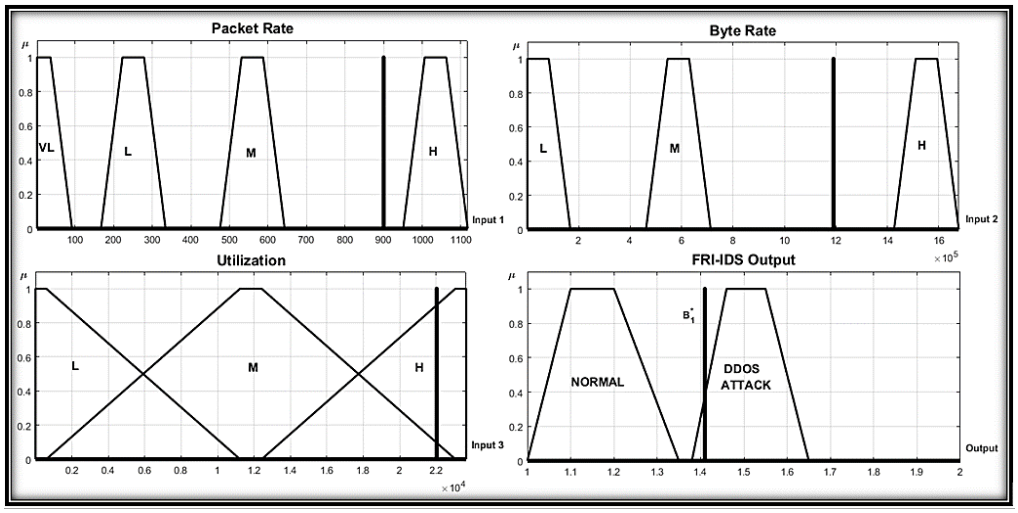

Fig. 21. FRI-IDS Output Response in Case of Attack

5.2.7 Experiments and Discussion

This subsection illustrates the testing and validating of the FRI-IDS model using the test-bed dataset. Thereupon, all experiments were conducted using MATLAB [10] and FRI toolbox [78]. The inference engine of the FRI-IDS model was performed using the FIVE method. Out of 28 fuzzy rules were generated in order to detect the DDOS attack. It is worth mentioning that the FRI-IDS was tuned using RBE-DSS. The tuning process stopped when the predefined performance index value obtained or when the number of iterations is reached. The tunning process of FRI-IDS stopped criteria was the maximum number of iterations. The code of FIVE method and other FRI methods can be used through the FRI toolbox which can be downloaded freely from [78]. The overall process of testing and validating the FRI IDS model was shown in Fig. 22.

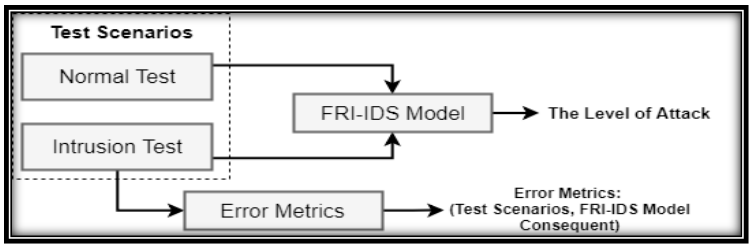

Fig. 22. The Testing and Validating Process of FRI-IDS Model

The FRI-IDS model was tested and evaluated based on two test scenarios:

- The first test titled as the normal test scenario where 5000 instances of normal cases is used as input parameters of the FRI-IDS model. The extracted 5000 instances of normal cases presented as a matrix of the normal test.

Normal Test Matrix

$$\begin{Bmatrix} 328 & 505434 & 23632 \\ 125 & 124944 & 5961 \\ 658 & 379060 & 3092 \\ . & . & . \\ . & . & . \\ \text{Pkt Rate} & \text{Byte Rate} & \text{Utilization} \end{Bmatrix}$$

- The second test scenario was titled as an intrusions test scenario where 5000 instances of intrusion cases are used as input parameters of the FRI-IDS model. The extracted 5000 instances of intrusion cases presented as a matrix of intrusion test.

Intrusion Test Matrix

$$\begin{Bmatrix} 1118 & 1677420 & 14377 \\ 963 & 1444030 & 12381 \\ 328 & 18067 & 845 \\ . & . & . \\ . & . & . \\ \text{Pkt Rate} & \text{Byte Rate} & \text{Utilization} \end{Bmatrix}$$

The previous two matrices of normal and intrusion test scenarios were chosen as a two-input testing file of the FRI-IDS model. The evaluation of the FRI-IDS model carried through the computed error metrics of the test scenarios. The inferred consequence of the FRI-IDS model was compared along with the actual values of normal and intrusion. According to [57], the error metrics of the test scenarios were extended to the following performance metrics: TP, FP, TN, and FN parameters.

Furthermore, the previous mentioned performance metrics offer the capability to compare the FRI-IDS model results with other algorithms that have been implemented for detecting DDOS attacks. As a result of the test scenarios of the FRI-IDS model, 10000 cases were tested

successfully. These cases were split as 5000 of normal cases and 5000 of intrusion cases. During the normal test scenario, only 3 records of normal cases were inferred incorrectly by the FRI-IDS model. Besides, during the intrusion test scenario, 332 of intrusion cases were inferred incorrectly by the FRI-IDS model. The obtained results besides the error metrics values presented in Table 16.

Table 16. The Result of The Test Scenarios Cases

	Normal	**Intrusion**	**Total**
Normal	4997	3	5000
Intrusion	332	4668	5000
Total	5329	4671	10000

According to the obtained results and the error metrics values of Table 16, the confusion matrix parameters of the FRI-IDS model presented in Table 17.

Table 17. Confusion Matrix Of FRI-IDS Model

Alert Response	**Intrusion Packet Prediction**	**Normal Packet Prediction**
Intrusion	TPR = 0.93	FNR = 0.06
Normal	FPR = 0.0006	TNR = 0.999

The implemented experiments have demonstrated that the FRI-IDS model obtained 96.65% as an overall detection rate. The computed performance metrics concluded that the FRI-IDS model obtained an acceptable value for the detection rate, and it decreases effectively the false positive rate. Decreasing the false positive rate helps to reduce a large amount of IDS alerts. To summarize the aforementioned results, the FRI-IDS model could be a suitable approach to be implemented as a detection mechanism for the following reasons:

- The FRI-IDS model offers an extension of the binary decision problem to continuous truth value, in which the inferred consequence like “the level of intrusion”, which makes the response result more readable and clearly analyzed rather than a binary decision.
- It is difficult to identify a clear boundary between normal and intrusion packets. Therefore, the fuzzy system effectively smooths the abrupt break of normal and intrusion.
- FRI methods can serve deducible (interpolated) conclusions even in case if some situations are not explicitly defined in fuzzy rule-based knowledge representation.
- The implemented experiments show that the FRI-IDS model obtained an accepted value for the detection rate and false-positive rate.

5.2.8 Difference From Prior Works

From another perspective, there are several convincing efforts of literatures to implement different classification algorithms in order to prevent DDOS attacks. Therefore, to support the idea of implementing the FRI-IDS model as a detection mechanism could be a suitable approach. The obtained results of the FRI-IDS model compared with Alkasassbeh [19] and Irfan Sofi [69] results.

They have employed different classification algorithms to detect DDOS attacks using the same test-bed environment. Table 18 summarized the comparison result of the FRI-IDS model with other classification algorithms.

Table 18. FRI-IDS Model Vs Data Mining Algorithms

Authors	**Algorithms**				**FRI-IDS**
Irfan Sofi *et al.* [69]	Neural Network	Naive Bayes	Decision Tree	Support Vector Machine	**FRI-IDS**
Detection Rate	98.91%	96.89%	98.89%	92.31%	**96.65%**
Alkasassbeh *et al.* [19]	Neural Network	Naive Bayes	Random Forest		**FRI-IDS**
Detection Rate	98.63%	96.91%	98.02%		**96.65%**

According to the results, the overall detection rate of the FRI-IDS is in pairs with other methods. On the example dataset, it outperforms the detection rate of the support vector machine. FRI IDS reduced effectively false positive rate value which reduced a large number of IDS false alerts. Moreover, FRI-IDS offers the extension of the binary decision problem to continuous truth value, in which conclusions like "the level of the attack" can be also simply defined. Nevertheless, these methods desire dense fuzzy rules as a major requirement. Regarding the large number of network connections, it could be very hard to comply with the dense fuzzy rules. However, the FRI-IDS model is characterized to offer the attack alert generation in case of a lack of information and definition of the existing knowledge base. It can generate their approximate conclusions either directly from inputs by an interpolating fuzzy function, or by interpolating a new fuzzy rule which overlaps the input.

5.3 Summary

IDS is one of the effective solutions to detect and prevent intrusions occurrence. According to a large amount of financial loss and privacy violation of intrusions, IDS has become a fundamental solution to network security. There are different challenges in implementing sufficient IDS. One of these challenges is the binary decision in detection techniques. The typical detection mechanisms of IDS had a boundary problem.

This chapter has investigated the capabilities to use the FRI methods in the IDS application area. This investigation is practiced by implementing the FRI-IDS model as a detection mechanism for the DDOS attacks. The FRI-IDS model was constructed using the sparse fuzzy model identification. The fuzzy rules of the FRI-IDS model were generated and optimized using the RBE-DSS method. In the example of the chapter as a test-bed environment, an open-source DDOS dataset was used. The implemented experiments have demonstrated that the FRI-IDS model obtained an accepted detection rate. It has reduced effectively the false positive rate value which decreased a large amount of IDS alerts.

Additionally, the FRI-IDS model can serve the interpolated conclusions even in case if some observations are not covered directly by fuzzy rules. The obtained results of FRI-IDS model compared with other literature's results which they employed different algorithms to detect the

DDOS attacks using the same dataset, FRI IDS model outperforms the detection rate of support vector machine algorithm in the example of DDOS dataset, where other algorithms (neural network, random forest and decision tree) recorded lightly higher detection rate. Consequently, the FRI-IDS model could be a suitable approach for detecting intrusions if it is implemented as a detection mechanism. It is characterized by offering the capability to present the detection level of intrusion and permits the attack alert generation in case of a lack of information and definition of the existing knowledge base.

Thesis.I: The FIVE based fuzzy rule interpolation model can be used in the IDS as a suitable inference method. Furthermore, the FRI inference system has yielded promising results when implemented as an IDS detection mechanism. Additionally, during the studies test application, the FRI inference system effectively decreased the rate of false positive values. Moreover, because of it's tendency for fuzzy rule based knowledge representation, it can easily adapt to expert knowledge, and be suitable for predicting the potential threat level.
The results introduced in this chapter are supporting the statement of Thesis I and published in [5].

6. FRI and SNMP-MIB for Emerging Network Abnormality

This chapter is introducing a novel method to detect abnormalities by combining the (FIVE) FRI reasoning with the Management Information Base (MIB) parameters. In that respect, there is no need to deal with raw traffic processing, which is time-consuming, and difficult to compute. This method also eliminates the need for creating a complete fuzzy rule base. The MIB parameters reflect the normal and abnormal nature of the network traffics. The implementation of the proposed FRI-SNMP detection approach was constructed into three main phases:

- To identify how the SNMP-MIB parameters can be used as a useful data source for detecting the abnormality.
- To implement the proposed detection approach based on the strength of the fuzzy rule interpolation and the SNMP-MIB parameters.
- To highlight and discuss the difference between the proposed detection approach and other approaches that detect intrusions based on SNMP-MIB parameters.

6.1 Detection Approaches Based on SNMP-MIB Parameters

This subsection presents some relevant works related to the application of the detection mechanism for intrusion detection. It also provides a brief overview of different methods and approaches that are used for intrusion detection using the SNMP-MIB parameters. Typically, the SNMP is used to collect information from different data sources such as switches, routers, etc. This information is used to manage and troubleshot different network devices. The typical IDS detection mechanism uses the raw traffic to assess the threats within connected devices. Raw traffic requires extensive investigative pre-processing to extract the required information. This investigative pre-processing is a time-consuming task for the IDS detection mechanism [79]. Therefore, the SNMP-MIB parameters offer a solution that provides the required wealth of information without needing to extensively investigate and pre-process a large amount of raw traffic.

A network attack [80] is any process used to perform malicious actions against any host inside a network with the intention of compromising the security of that network. Alkasassbeh et al. in [81, 47], proposed a Mobile Agent (MA) to read the SNMP-MIB data from the local nodes that use the MIBs to store that traffic data locally. The MA was used to overcome the limitations of the centralized management system or the IDSs. The above-mentioned works reflected that the statistical methods, based on Wiener filter and MA technology, could be joined to detect network intrusions. The suggested model intended to detect all the attacks. The MIB variables were chosen from the IF and IP groups, and the studied scenarios were the decoy port-scan, the buffer overflow, the brute force attack, and the null session attack. The work of Cabrera et al. presented in [82] is one of the first attempts to apply the SNMP-MIB parameters as the data source for the IDS detection mechanism. The authors proposed a detection mechanism for Distributed Denial of Service (DDOS). Three types (Targa3, UDP Flood ad Ping Flood) of attacks were detected using the proposed detection mechanism. Altogether 90 SNMP-MIB parameters were used to detect the

previous types of DDOS attacks. The SNMP-MIB parameters were collected from the simulated test-bed environment. The proposed detection mechanism was able to successfully detect the predefined DDOS attacks.

In [79], Yu et al. propose a lightweight IDS detection mechanism by adapting the machine learning algorithm to the SNMP-MIB parameters. The proposed detection mechanism avoided the raw traffic analyzation and used the statistical MIB parameters to recognize the degree of abnormality within connected devices. The authors applied the features selection algorithm to decrease a large number of SNMP-MIB parameters. The relevant parameters were determined using the correlation feature selection algorithm and the network traffic was classified using the support vector machine. Furthermore, the SNMP-MIB parameters were extracted from real-time experiments. The proposed approach achieved a fast detection time and a high rate of accuracy. The work of Yu et al. in [8] focuses on using SNMP-MIB parameters to detect flooding attacks. The authors designed the flooding attack detection mechanism by adapting the C4.5 algorithm. The SNMP-MIB parameters were collected from the simulation environment operating the flooding attack. Then, the C4.5 algorithm starts detecting and classifying the traffics based on the recorded SNMP-MIB parameters. The proposed approach was able to obtain a 93.0% detection rate.

In [83], Hsiao et al. proposed a detection mechanism for an ARP spoofing attack. The SNMP-MIB parameters were used instead of the raw traffic. The proposed detection approach was contracted into three parts. The first part was adapted to the Naive Bayesian algorithm. The second part applied the support vector machine and the last part applied the C4.5 algorithm. The authors recorded their findings and highlighted both the weak and strong points for each of these parts that were used for detecting the ARP spoofing attack based on SNMP-MIB parameters. Typical performance metrics such as accuracy rate, false-positive rate, and missing rate were recorded for the implemented algorithms. The implemented experiments demonstrated that the C4.5 achieved the highest accuracy rate. The lowest value of false alarms was recorded by the support vector machine algorithm. The Naive Bayesian algorithm had the lowest accuracy rate within the implemented experiments.

From another perspective, the decentralized detection mechanism based on the clustering algorithm and SNMP-MIB parameters was proposed by Cerroni et al. in [84]. The proposed decentralized mechanisms were divided into the monitoring phase and the traffic detection phase. In the monitoring phase, the SNMP-MIB parameters were gathered from several agents. These parameters were forwarded to the distributed data mining algorithms for the sake of classifying the observation as either normal or abnormal. The proposed decentralized detection mechanism was tested and evaluated using the SNMP-MIB dataset and was able to detect the plausible intrusions within the dataset that related to the decentralized detection mechanisms.

In [85], Cerroni et al. introduced a new distributed data mining method in order to detect the intrusion based on SNMP-MIB parameters. The proposed method has been tested for decentralized testbed environments. The SNMP-MIB parameters were collected from the simulated network environment. Fourteen SNMP-MIB parameters were used to detect the specific type of DDOS

attack. These parameters related to the IP and TCP groups. The experiments conducted reflect that the proposed mechanism obtained an acceptable detection rate.

Some other works were used in a hybrid approach, in conjunction with the SNMP-MIB parameters, to detect for abnormalities within the network traffic. In [86], Namvarasl and Ahmadzadeh proposed a hybrid approach to detect DDOS based on SNMP-MIB parameters. The proposed approach consisted of three modules; the first module was constructed for the features selection of the SNMP-MIB parameters. In the second module, the detection mechanism was generated based on high ranked SNMP-MIB parameters and the C4.5 and RIPPER were implemented to detect the intrusions. The proposed approach was tested and evaluated based on the SNMP-MIB dataset. The imported dataset consisted of 66 SNMP-MIB parameters. It also had the following type of attacks: UDP flood attack, ICMP flood attack, and TCP-SYN flood attack. The proposed approach was able to detect different types of attacks within the imported SNMP-MIB dataset.

The previous works provided plausible contributions and, at the same time, supported the idea that the SNMP-MIB parameters are instrumental in detecting and recognizing the intrusion. Using SNMP-MIB parameters avoids the need for the time-consuming analysis of massive amounts of raw traffics. Previous works also shared common issues such as the difficulties associated with the detection mechanisms which suffered from a lack of clear boundaries for distinguishing between normal and abnormal traffic. Furthermore, the previous detection mechanisms did not determine the level of degree of abnormality; they only applied a binary decision to recognize normal and abnormal traffic.

In response to these issues, in this thesis, a novel approach for detecting and preventing abnormalities by combining the (FIVE) FRI reasoning with the SNMP-MIB parameters is suggested. The FRI approaches are implemented primarily to avoid binary decisions, and instead, establishing a gradient scale for distinguishing the normal and abnormal traffics. Additionally, they generate results (Detection Decision) in a clear and understandable form. Contrary to the classical fuzzy systems, the FRI approaches do not require a large number of fuzzy rules (i.e. expert knowledge this case) for determining the level or degree of abnormality within the protected network. Finally, the FRI approaches can produce results, even in case of lacking defined knowledge representation (sparse rule base).

6.2 IDS Model-based FRI (FIVE) and SNMP-MIB

This subsection introduces the full architecture of the proposed detection approach in detail according to its main functions and prerequisites.

6.2.1 The Studied Attack Types

The MIB parameters are characterized by the wealth of beneficial information they can offer for defining abnormality and reflect the normal and abnormal nature of the network traffics. The MIB dataset applied as an example in this work is introduced by Al-Kasassbeh et al. [87]. This

dataset was originally generated to target DoS attacks. A DoS attack is blocking legitimate user requests for services the server can provide. Such attacks can be carried out by flooding the chosen server with a high volume of traffic, thereby consuming all of the server's resources and, consequentially, preventing the server from responding to genuine requests. These attacks can be generated either from a local or remote node in a different network. DoS attacks are usually difficult to assess and prevent [88], making them one of the most challenging type threats. An even more severe threat is the DDoS attack, which is a type of flooding attack that is generated from various nodes simultaneously [89].

In the dataset, seven classic DoS flooding attacks are studied. The first one is the TCP-SYN attack. This attack abuses the susceptibility of the three-way handshake mechanism (SYN, SYN-ACK, and ACK) operating between the host and the server when establishing the TCP/IP protocol connection. During the process, the attacker sends an SYN control packet. The server on the other side responds to the SYN request by sending an SYN-ACK packet. Meanwhile, the server stores and reserves all the resources and waits for an ACK from the sender. While the server is waiting for the ACK packet, the request remains in the memory stack. The server will not receive ACK packets from the attacker, and the attacker will send more SYN requests within a short amount of time to exhaust the server's resources until it is unable to respond to any new requests [90].

The second attack is the UDP flood attack. This type of attack sends UDP packets to random ports on the victim server. When the server deals with these packets and discovers that the packets are empty, the server will then send back an error message through ICMP protocol to the sender. The server's resources, such as bandwidth which is very important for the performance of the network will be exhausted by the volume of useless or empty packets, and therefore will not be able to respond to any other requests. The UDP flood attack is typically very effective in smaller networks [91].

The third attack is the ICMP-ECHO attack. This type of attack floods the victim's bandwidth thus preventing new connections from being initiated. The PING command is used to test whether, or not the host is alive on the network. When a device receives a PING request, it will automatically reply with a message informing the sender of its status. This type of attack tricks the system by crafting a large number of ICMP packets using a spoofed source IP address as the victim's IP address in order to reply directly to it later. Then it sends these packets through a network broadcast address which directs numerous hosts to send their replies to the same victim's IP address at the same time. Eventually, the high volume of reply messages will overwhelm the system and exhaust the victim's resources [87].

The fourth attack is the HTTP flood attack. This attack targets a web server and consumes the victim's resources, such as memory, CPU, bandwidth, etc. The attacker sends a huge number of valid HTTP requests (GET or POST) to a web server. Typically, these requests are generated by hosts called botnets. Each one of the bots sends a large number of legal requests at once. If there is a large number of botnets, the request rate will be higher than that is usually generated by typical users. This attack may be one of the most dangerous threats because it is hard to differentiate between normal and abnormal HTTP traffic [91]. The fifth attack is the Slowloris attack, whereby

the attacker sends sessions with a high load of requests by opening multiple connections to the victim server and trying to keep these connections open as long as possible. In this case, the requests are partial HTTP requests. The attack lasts until all available sockets are reserved by the HTTP requests, causing the server to freeze in response to any legitimate connection [7]. The final attack is the Slowpost attack. Similar to the previous attack, the attacker hereby sends a complete, rather than a partial, HTTP header request, including the content-length field in the post message body. The data fills the message body at the rate of one byte every two minutes. At the same time, the server remains to wait for each message body to be completed, leading to a denial of services [87].

SNMP-MIB data are rich sources providing clear statistical information about the current network device status. The SNMP-MIB is a widely deployed protocol in most network devices, and available without any additional new hardware or software investment. By reading the MIB data, some of the major challenges of intrusion detection can be avoided. The dataset we used contains 4998 connection records. The data is distributed into eight main classes as described in Table 19.

Table 19. Traffic Type and Number of Generated Record

No.	Type of Traffic	Number of Records
1	Normal	600
2	TCP-SYN	960
3	UDP flood	773
4	ICMP-ECHO	632
5	HTTP flood	573
6	Slowloris	780
7	Slowpost	480
8	Brute Force	200
Total	4998	

This MIB dataset has been collected from a router. The dataset has 34 MIB variables from 5 MIB groups in MIB-II. The groups are IF, IP, TCP, UDP, and ICMP. The groups and their variables are listed in Table 20.

Table 20. The SNMP MIB Parameters

Interface Group	TCP group	IP Group	ICMP Group	UDP Group
IF In Octets	TCP Out Rsts	IP In Receives	ICMP In Msgs	UDP In Datagrams
IF Out Octets	TCP In Segs	IP In Delivers	ICMP In Dest nreachs	UDP Out Datagram
IF out Discards	TCP Out Segs	IP Out Requests	ICMP Out Msgs	UDP In Errors
IF In Ucast Pkts	TCP Passive Opens	IP Out Discards	ICMP Out Dest reachs	UDP No Ports
IF InN Ucast Pkts	TCP Retrans Segs	IP In Discards	ICMP In Echos	
IF In Discards	TCP Curr Estab	IP Forw Datagrams	ICMP Out Echo Reps	
IF Out Ucast Pkts	TCP Estab Resets	IP Out No Routes		
IF OutN Ucast Pkts	TCP Active Opens	IP In Addr Errors		

6.2.2 IDS Model-based FRI and SNMP-MIB

The proposed IDS approach is adapting the FIVE method as an inference engine. The concept of the vague environment, introduced by Klawon in [92], the vague environment can be defined on the basis of similarity or indistinguishability of the elements. The concept of the vague environment can be expressed by a scaling function (s). The proper scaling function (s) which describes all the fuzzy sets of a fuzzy partition, should be implemented to produce a vague environment. According to [72, 73], the scaling function (s) is suitable for describing the shapes of all fuzzy set of a fuzzy partition. In the vague environment, the level of similarity between two fuzzy sets illustrates the fuzzy membership function M(x). In the vague environment, two values are ε-distinguishable if their distance is greater than ε:

$$\varepsilon > \delta_s(X_1, X_2) = \left| \int_{X_2}^{X_1} s(x)dx \right| \tag{14}$$

Likewise, $\delta_s(X_1, X_2)$ represents the vague distance for the values X1 and X2.

The general structure of the proposed detection approach, as it is shown in Fig. 23., is initiated by the data-cleaning stage. This stage is responsible for assembling the required information using the SNMP agents. This information is then forwarded to the SNMP manager which consists of a repository of MIB parameters. During the data cleaning stage, the MIB parameters were evaluated to determine their relevant parameters. The cleaning stage aims to reduce a large number of MIB parameters by eliminating those that are irrelevant.

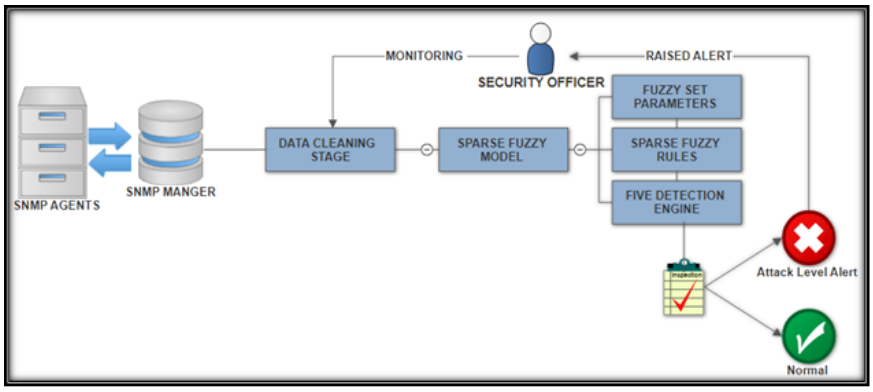

Fig. 23. The General Structure of The Proposed Detection Approach

The imported MIB dataset consists of a large number of MIB parameters. To simplify the process, the top five relevant MIB parameters [93, 87] were chosen as input parameters for detecting the abnormality in the proposed detection approach. These MIB parameters are the IP-Out-Discards and IP-In-Discards from the IP MIB group. IF-In-Discards and IF-Out-Discards from the interface MIB group and ICMP-Out-Dest-Unreachs from the ICMP MIB group. The training part is designed to generate two repositories. The first repository consists of only the

intrusion traffics, and the second repository includes only the normal traffics. Consequently, 2970 instances of normal and abnormal traffic were stored in two repositories. These instances had only the top five relevant MIB parameters.

The detection stage consists of several operations including fuzzification, sparse rule base generation and adapting the inference engine. The proposed detection approach was designed and constructed using the SFMI [75]. Before constructing the proposed detection approach, the top-five relevant MIB parameters were forwarded to the SFMI. As mentioned in subsection (5.2.6), the fuzzy rule generation and fuzzy sets optimization are the necessary modeling steps for constructing the proposed detection approach. The fuzzy rule generation and the fuzzy sets optimization were adapted using the RBE-DSS method which is introduced by Johanyák in [76]. As a result of adapting the RBE-DSS method, the FRI methods effectively reduce the total number of fuzzy rules, having 245 fuzzy rules for the FIVE method to detect the abnormality based on five MIB parameters. Table 21 presents a sample of the sparse rule base that was generated by the RBE-DSS method.

Table 21. The Sparse Rule-base on The MIB Parameters

No.	IFoutDiscards	IFInDiscards	IPOutDiscards	IPInDiscards	ICMPOutDestUnreachs	Consequence
1	Low	Low	Very Low	Very Low	Low	Normal
2	Low	Low	Very Low	Very Low	Medium	Normal
3	Low	Low	Very Low	High	High	Normal
4	Low	Low	Medium	Medium	Medium	Normal
5	Low	Medium	Very Low	Very Low	Low	abnormal
6	Medium	High	Very Low	Very Low	Low	abnormal
7	High	Low	Very Low	Very Low	Low	abnormal
8	High	High	Very High	Very Low	Low	abnormal
9	Medium	Low	Very High	Very Low	Low	Normal
10	Medium	High	Very Low	High	Medium	abnormal
11	Medium	High	Medium	Medium	High	abnormal
12	Medium	Medium	Very High	Very Low	Low	Normal
13	High	Low	Medium	Medium	Low	abnormal
14	Medium	High	Very Low	Very Low	Low	abnormal
15	Low	High	Very High	Very Low	High	abnormal

In the RBE-DSS method, the trapezoidal membership functions were chosen to apply during the fuzzy set parameters optimization. The ICMP-Out-Dest-Unreachs, IF-Out-Discards and IF-In-Discards MIB parameters have three membership functions. These membership functions are classified into the following linguistic terms: Low, Medium and Large. The ip-In-Discards MIB parameter has four membership functions classified into the following linguistic terms: Very Low, Low, Medium and Large. Finally, the IP-Out-Discards MIB parameter represents five membership functions which are classified into the following linguistic terms: Very Low, Low, Medium, Large and Very Large.

The RBE-DSS method optimizes the values of fuzzy set parameters to the maximum performance of the fuzzy IDS. Fig. 24. presents the proposed detection approach's antecedent

partitions. The proposed detection approach could offer the conclusion (detection result) even in situations where some MIB parameters are not explicitly defined in the generated fuzzy rule base. Table 22 lists the fuzzy set values optimized by the RBE-DSS method.

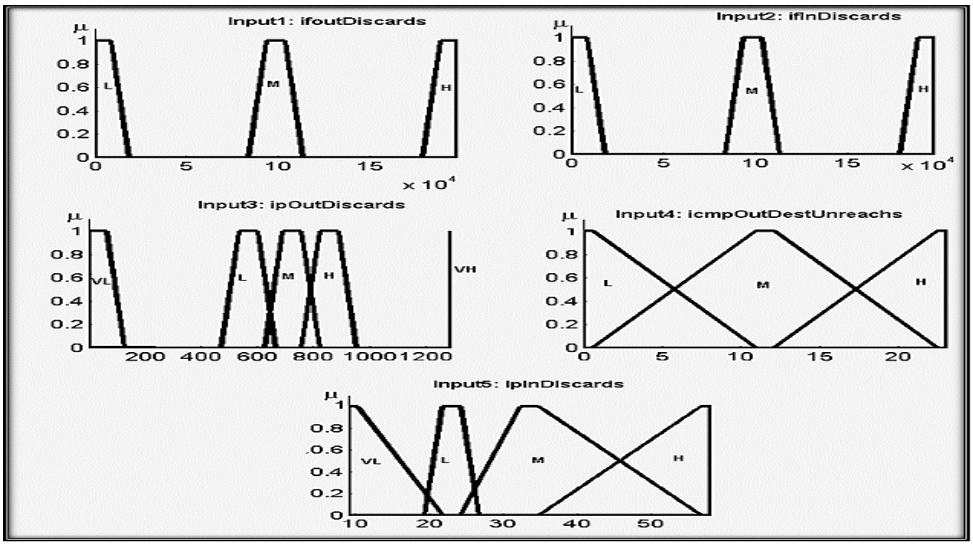

Fig. 24. The Antecedents Partitions of The Proposed Detection Approach

Table 22. The Optimized Fuzzy Set Parameters

IpOutDiscards	**Very Low**	**Low**	**Medium**	**High**	**Very High**
	[1 1 64.8 129.1]	[472.55 536.85 601.15 665.45]	[627.93 692.23 756.53 820.83]	[758.75 823.05 887.35 951.65]	[1287 1287 1287 1287]
IpInDiscards	**Very Low**	**Low**	**Medium**	**High**	
	[9 9 10.23 21.78]	[19.33 21.78 24.23 26.68]	[24.23 32.28 34.73 56.78]	[34.73 56.78 58 58]	
icmpOutDestUnreachs	**Low**		**Medium**	**High**	
	[0 0 0.58 10.93]		[0.58 10.93 12.08 22.43]	12.08 22.43 23 23	
ifinDiscards	**Low**		**Medium**	**High**	
	[0 0 8602.57 18434.06]		[83567 93399 103230 113062]	[178195 188027 196630 196630]	
ifoutDiscards	**Low**		**Medium**	**High**	
	[0 0 8602.57 18434.06]		[83567 93399 103230 113062]	[178195 188027.44 196630 196630]	

6.2.3 Simulation and Results

This subsection introduces the simulation and discusses the results of the proposed detection approach in detail. As detailed in subsection (5.3.3), the total number of training data consisted of 2998 instances of normal and abnormal traffics. The training data were used to construct and optimize the proposed detection approach. The rest of the MIB dataset consisted of 1998 instances

that were used for the validation process. It is worth mentioning that, every observation within the SNMP-MIB dataset was presented as a fuzzy singleton. The proposed detection approach was able to generate intelligible results due to its fuzzy nature, subsequently allowing the degree of abnormality to be determined.

Fig. 25. presents the output response of the proposed detection approach in the case of the abnormal instance with the parameters which are listed in Table 23. The data conclude that the degree of abnormality has been determined. Subsequently, the results, which are now more concise, serve to help administrators for better understanding the current security status.

Table 23. Abnormal MIB Parameters Example

MIB Parameters	Value
IF-Out-Discards	7270
IF-In-Discards	7270
IP-Out-Discards	1287
IP-In-Discards	9
ICMP-Out-Dest-Unreachs	0

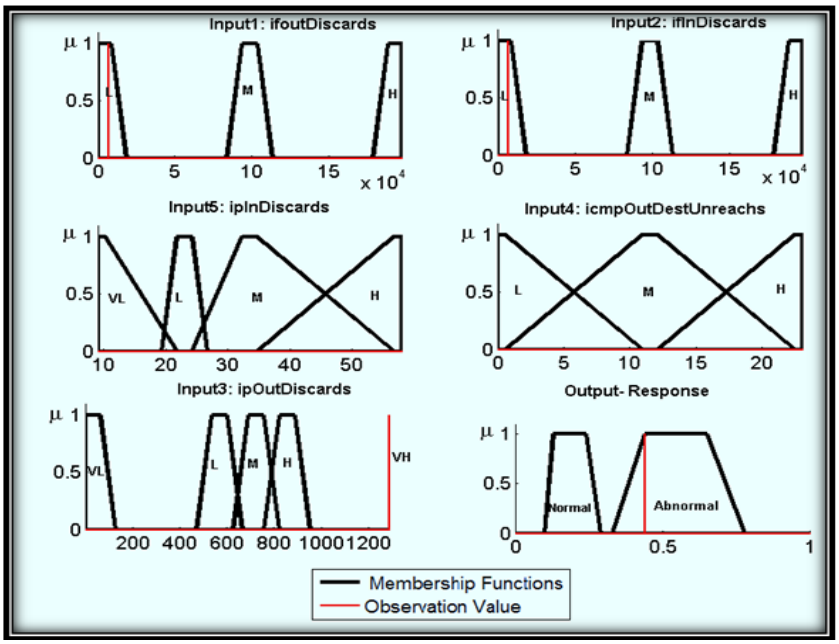

Fig. 25. The Output Response of The Proposed Detection Approach

The proposed detection approach was evaluated in a two-phase process. The first phase evaluated the normal repository and the second phase was evaluated the abnormal repository. A total of 1998 MIB parameter instances were tested and evaluated. Fig. 26. displays the results from both phases (normal and abnormal) of the detection approach's evaluation process.

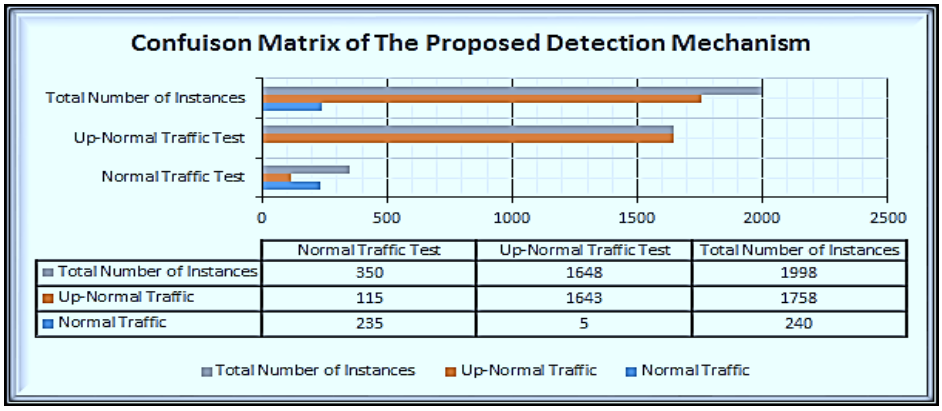

Fig. 26. The Confusion Matrix of The Evaluation Process

It was concluded that five instances of normal traffic were inferred incorrectly and 115 instances of abnormal traffic were inferred incorrectly. The obtained results have been carefully analyzed and investigated to highlight the strengths of the proposed detection approach. Table 24 presents the performance metrics for the proposed detection approach.

Table 24. The Performance Metrics For The Proposed Approach

Performance Parameter	Value	Formula
Sensitivity	0.9346	TPR = TP / (TP + FN)
Specificity	0.9792	SPC = TN / (FP + TN)
Precision	0.9970	PPV = TP / (TP + FP)
False Positive Rate	0.0208	FPR = FP / (FP + TN)
False Negative Rate	0.0654	FNR = FN / (FN + TP)
Accuracy	0.9399	ACC = (TP + TN) / (P + N)

To summarize the aforementioned results, the performance of the proposed detection approach achieved satisfactory values and, at the same time, supports the idea that implementing the fuzzy rule interpolation methods for the reasoning part together with the SNMP-MIB parameters could be a promising approach in the IDS application area. Moreover, the results obtained from the proposed detection approach were compared with other literature results [93] in which the same MIB dataset and the same number of relevant MIB parameters (top-5) were applied in combination with neural network, support vector machine and Bayesian network algorithms. Fig. 27. compares the results between the proposed detection approach and other algorithms (neural network, support vector machine and Bayesian network).

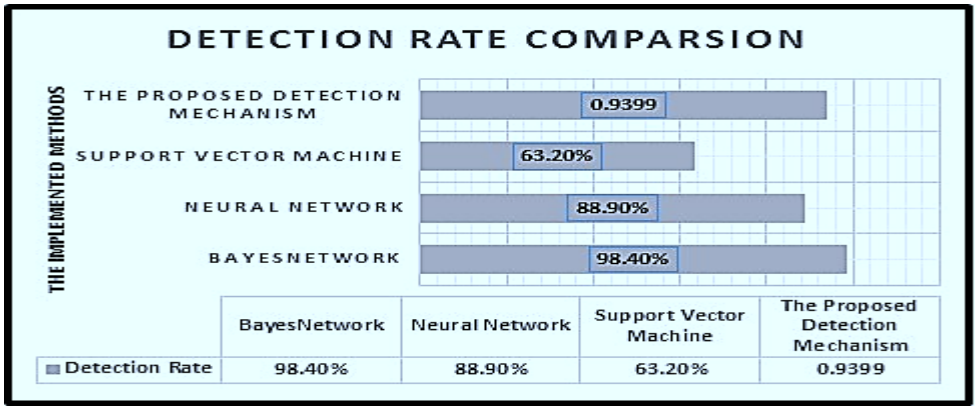

Fig. 27. The Detection Rate Comparison Results

Consequently, the implemented experiments demonstrated that the proposed detection approach achieved an acceptable accuracy rate. Moreover, it effectively reduced the false positive rate parameter. The conventional detection approaches focus on adapting the typical data mining algorithms, or the classical fuzzy reasoning methods, to be used with raw network traffics. Although FRI methods have been implemented in the IDS application area, these methods are still under investigation. Nevertheless, current research has yielded satisfactory results. The strength of FRI methods is derived from the combination of the fuzzy concept and interpolation techniques. Therefore, the FRI methods could pose an effective solution for the boundary problem and could also handle the deficiencies of the knowledge-base representation. The strength of the proposed detection approach is based on combining the MIB parameters with the fuzzy rule interpolation reasoning method. Thus, there is no need to deal with raw traffics which are time-consuming and difficult to compute. Furthermore, this method eliminates the need for the complete fuzzy rule base.

6.3 Summary

It is difficult to implement an efficient detection approach for IDS and many factors contribute to this challenge. One such challenge concerns establishing adequate boundaries and finding a proper data source. Typical IDS detection approaches deal with raw traffics. This traffic needs to be studied in-depth and thoroughly investigated in order to extract the required knowledge base. Another challenge involves implementing a binary decision. This is because there are no reasonable limits between normal and attack traffics patterns. The typical IDS detection mechanism uses the raw network traffic as a data source to detect abnormalities within the network. Dealing with raw traffics requires in-depth analysis and review to extract the information relevant for helping attack detection [79]. The SNMP-MIB parameters can also offer the required information, yet reducing the extensive processing time necessary for analyzing the raw traffics. The SNMP-MIB can be considered as a rich data collection fetched from a series of devices for producing realistic information about the health of a network, which can be also beneficial for detecting attacks.

This chapter has introduced a novel method to detect abnormalities by combining the Fuzzy Interpolation based on the Vague Environment (FIVE) FRI reasoning with the Management Information Base (MIB) parameters. In that respect, there is no need to deal with raw traffic processing, which is time-consuming, and difficult to compute. This method also eliminates the need for creating a complete fuzzy rule base. The MIB parameters reflect the normal and abnormal nature of the network traffics. Using SNMP-MIB parameters avoids the need for the time-consuming analysis of massive amounts of raw traffics. The proposed detection approach was tested and evaluated using open-source MIB parameters dataset. The conducted experiments reflect that the proposed detection approach could effectively detect the abnormal traffics within the selected SNMP-MIB parameters dataset with 93.9% accuracy.

Thesis **II**.: Applying FIVE based fuzzy rule interpolation for SNMP-MIB data achieving acceptable results in an IDS detection mechanism. I concluded that using this method there is no need to deal with raw traffic processing, which is time-consuming, and difficult to compute. The MIB parameters reflect the normal and abnormal nature of the network traffics. Furthermore, expert knowledge can be easily adapted by eliminating the need for creating a complete fuzzy rule base.
The results introduced in this chapter are supporting the statement of Thesis II and published in [129].

7. Fuzzy Automaton Based Detection Model

This chapter introduces the design of a novel model for detecting multi-step attacks by the application of the FRI based fuzzy automaton. First, the multi-step attacks and it's detection methods are presented briefly, then the discussion of the designed FRI based fuzzy state machine model follows.

7.1 Taxonomy of Multi-step Attacks

This subsection presents some different types of the well-known multi-step attacks for clarifying the sequence steps for those types of attack. It also shortly describes the main prerequisite steps, characteristics and events structure of the multi-step attacks.

Nowadays, network administrators face stressful environments with an overload of network traffics. This traffic needs to be analyzed and investigated to detect abnormalities. The IDS has benefited from the rapid growth of technology; however, intruder techniques have also adapted to the IDS detection mechanisms' new technological developments. Intruders have continued to advance their techniques and alter their behaviors to avoid detection by recent detection mechanisms. As a result, the danger of attacks has become increasingly more difficult to combat.

Computer and network security systems face different types of sophisticated attacks. One type of sophisticated attack is the multi-step attack. The multi-step attack [1, 2] is an attack composed of several prerequisite steps leading up to the final step which launches an attack targeting the victim's security hole. The attackers follow this technique to avoid detection. The prerequisite steps resemble normal behavior and serve as a subterfuge to facilitate the execution of the final step of the attack. As detailed in the security report of the Chinese network security organization [2], two types of multi-step attacks (denial of service and warms) recorded 60% of the total number of attacks around the world. As a result, multi-step attacks have become a constant challenge for both users and organizations. In 2017, the Kaspersky global security report [94] revealed that 91% of enterprise businesses are affected by these types of sophisticated attacks, the largest proportion of which are the denial of service attacks. The well-known types of multi-step attacks such as DoS Mstream, File Transfer Protocol (FTP) bounce and DoS Domain Name Server (DNS) were executed based on a sequence of prerequisite steps [95].

The multi-step attack is a constant challenge for the IDS because intruders may implement complex attack scenarios, composed of several prerequisite steps, all aimed at executing their final attack [3]. Often, there is a causal relationship between the attack steps and forecasting the next step of attack [4]. There is an increasing need to design and implement an efficient IDS detection mechanism capable of handling different attack scenarios. The IDSs face several challenges including being able to detect multi-step attacks and the boundary problem (applying the binary decisions in the detection mechanism) [5]. In terms of the multi-step attacks, there is a causal relationship between the prerequisite steps which allows for administrators to be able to predict the next step of the attack [4]. Therefore, the multi-step attacks consist of different preliminary phases that can be distinguished from one another. On the other hand, implementing an efficient

detection mechanism is also challenged by the boundary problem because there are no clear boundaries and no convincing threshold for defining normal and intrusion traffics [6]. The fuzzy system extends the binary decision to the continuous space, smoothing the boundaries and offering a solution to the boundary problem. Additionally, the results generated by the fuzzy systems are more comprehensible [5].

7.1.1 Denial of Service (DoS) Mstream Multi-Step Attack

DoS attacks are considered one of the most harmful types of attacks. They directly affect the confidentiality, integrity, and availability of network services. This attack aims to prevent several network and computer services [5]. The attackers perform several techniques to disrupt services such as consuming resources (network bandwidth, CPU, memory utilization, etc). Any consumption of these resources increases the system overload and, after a while, the service slows down or becomes unavailable for end-users [19]. At the early stages of the DoS-Mstream attack, attackers attempt to perform a sequence of prerequisite steps to launch their final goal. The DoS-Mstream attack has five prerequisite steps [96] to reach the desired goal successfully. This sequence of steps is summarized as follows:

- The attacker executes one of the probe tools (i.e. IP sweep), these tools are used to discover and collect some required information such as live IP addresses, operating system version, services, and opening ports.
- From the collected live IP addresses, the attacker searches for the hosts that had enabled the service of "sadmind" using "ping" command options.
- As a result, the attacker is able to generate a list of intended victims. The attacker collects the victims to implement the root access login using the Remote SHell (RSH) access script. The aim of this step is to give the attacker permission over the victims' systems.
- DoS-Mstream installation begins by infecting victims with the root access login shell.
- Once infected, the DoS-Mstream multi-step attack is successfully executed.

As a result of the executed DoS-Mstream multi-step attack, the system service is disrupted and the protected data are exposed to illegal access. Attackers do not typically launch their attacks blindly. They begin their attacks with legal steps set up to uncover host information, services, etc. After, they execute the remaining steps of the attack. It is worth mentioning that these probe tools are designed for authorized users to discover and troubleshoot network and computer devices. However, attackers exploit these tools to execute their attacks. Fig. 28. presents the event sequence of the DoS-Mstream multi-step attack.

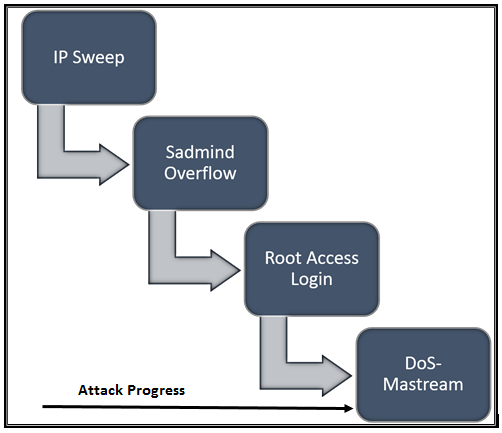

Fig. 28. The Sequence Events of The DoS-Mstream Attack

7.1.2 File Transfer Protocol (FTP) Bounce Multi-step Attack

The FTP bounce multi-step attack is executed by exploiting weaknesses in the FTP protocol. The standard FTP specifications include features that could be exploited by attackers. The main purpose of the FTP bounce attack is to transfer prohibited data within network ports [97]. The attackers exploit the FTP server's passive mode to illegally send and receive data within network ports. In the FTP server's passive mode, the trusted client initiates the commands and data sessions. The attackers exploit the initiated sessions to launch a Remote SHell (RSH) message against the FTP server which possesses a trusted client record [98]. According to [98, 97], the FTP bounce multi-step attack is carried out as follows:

- The attacker uses one of the vulnerability tools to uncover and collect some required information such as the server's live IP addresses, the FTP server version, opening ports and services.
- The attacker prepares a list of vulnerable victims that are running a RSH shell.
- The attacker uploads the malicious file to the infected victims now running the RSH shell and uses the port commands to initiate the data transfer.
- If the previous step is completed successfully, the attacker then forwards the FTP server output to the RSH shell port.
- The infected victim accepts the forwarded files and the attacker begins executing them as a sequence of commands. Fig. 29. presents the sequence of events for the FTP bounce multi-step attack.

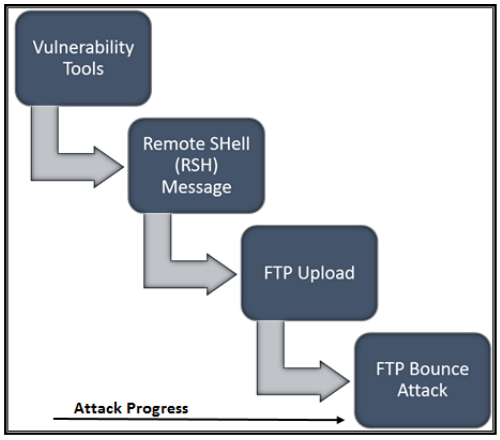

Fig. 29. The Sequence of Events of The FTP-Bounce Attack

7.1.3 DoS on Domain Name Server (DNS) Multi-step Attack

Internet service has been available now in different application areas. It serves several applications such as bank transactions, mail systems, social networks and more. Website services are also targeted by security threats, leading users to be concerned about service availability and access to their personal information. The DNS service is considered a critical component of internet infrastructure. It consists of the formal database of the public IP addresses and their hostnames. It also offers official mapping between the IP addresses and domain names [95]. Attackers execute the DoS-DNS multi-step attack by exploiting security weaknesses. This attack aims to prevent the DNS server services from being reached by end-users. The DoS-DNS multi-step attack [99] is implemented as follows:

- The attacker defines the expected DNS victim by using the nslookup which is a legal command that could be used by the authorized administrator. The output of the nslookup command is the current valid DNS server.
- The attacker verifies the primary active DNS server by using the ping command.
- The attacker initiates the DNS probe tools to define the DNS version, opening ports and the current running services.
- The attacker executes the DoS-DNS attack scripts such as WinNuke [99] or HyenaeFE [100]. Fig. 30. presents the sequence events of the DoS-DNS multi-step attack.

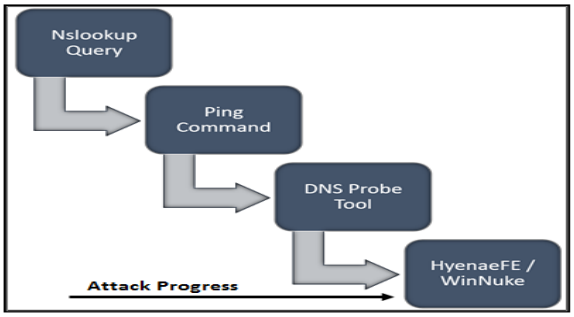

Fig. 30. The Sequence of Events of The DOS-DNS Attack

7.2 Multi-step Attacks Detection Methods

Multi-step attacks pose a constant challenge for protecting the network and computer resources. The typical IDS detection mechanism effectively detects low-level attacks (single-stage attacks used to obtain the target). These types of attacks can be detected using either a common convincing threshold or based on pre-defined rules [101]. On the other hand, the multi-step attack performs several prerequisite steps leading up to the execution of the final step. It has different, distinguishable preliminary phases. The state machine detection mechanisms effectively detect multi-step attacks [96, 102, 103].

This subsection presents some of the recent relevant works related to detect and prevent the multi-step attacks using state machine detection mechanisms. It also provides a brief overview of different methods and approaches that are used as a detection mechanism for IDS.

7.2.1 Hidden Markov Model Against Multi-step Attacks

HMMs [104] are implemented based on the probabilistic finite state machine to generate a predictive model for the sequence of events. The HMMs consist of two parts: observable events and hidden states. The probabilistic models can be implemented in several domain problems such as IDS, signal processing, pattern recognition and more. Typically, effective HMMs depend on two fundamental steps [105]:

- Full understanding of the domain problem to characterize the possible events.
- Parameter optimization (there are different tuning algorithms implemented with HMMs such as Genetic Algorithms (GA) and Baum-Welch (BW) algorithm.

In [106], Shrijit et al. use the HMMs to implement an approach for detecting DoS multi-step attacks. The mathematical model of the urn and ball was implemented to extract the required events. The expected observations were defined as source bytes, destination bytes, duration, host login, and guest login. HMM, parameters were tuned using the standard BW algorithm. The simulation environment shows that the proposed HMM approach obtained a 79% detection rate. The work of Zhang et al. in [2] introduces two different HMMs for detecting the multi-step attacks. The first HMM model was implemented and optimized using the BW algorithm. The second HMM model was designed and implemented without a training and optimization phase. The two proposed models were tested and evaluated using the Defense Advanced Research Projects Agency (DARPA) multi-step attacks dataset [107]. The results obtained reflected that the optimized HMM model effectively decreased the false positive alerts. Meanwhile, it outperformed the untrained HMM model in the detection and prediction of multi-step attacks.

There are other hybrid HMMs solutions too. Devarakonda et al. [108], propose a model to detect multi-step attacks based on HMMs and Bayesian networks. The proposed hybrid model was divided into two phases. In the first phase, the Bayesian network algorithm was used to extract the required HMM states from the Knowledge Discovery Databases (KDD99) [60]. The second phase was performed based on the extracted state's transition of the Bayesian network algorithm. The validation process for the proposed hybrid model reflects that the model detected the multi-step

attacks with a high detection rate. In [109], Aneetha et al. propose the use of a hybrid model of clustering algorithm and a HMM for detecting the multi-step attacks. The proposed probabilistic model was divided into two parts. The first part was performed to define the states, based on a clustering algorithm. The extracted states were forwarded to the second part of the proposed probabilistic model. In the second part, the state transitions probability matrix was generated with the initial distribution matrix. The proposed hybrid probabilistic model detects the multi-step attack even within the early stages of the attack and achieved a 95% detection rate.

The work of Devarakonda et al. in [110] focuses on detecting and preventing the multi-step attack at its onset (before it poses a severe risk). The proposed detection approach was adapted using a hybrid HMM. The Bayesian network algorithm was used to extract the system states. The proposed approach was optimized using the BW algorithm. The state transition tables for both the normal and multi-step attacks were generated by the Bayesian network algorithm. The simulated environment (DARPA dataset) reflected that the proposed approach could detect the multi-step attack within the DARPA dataset even within the early stages of the attack. The system states are essential for implementing the HMMs as a detection mechanism. According to the literature's [109, 110, 111, 106], the system states could be determined using various methods including the Bayesian network, clustering algorithms, and Non-Nested Generalized Exemplars (NNGE).

7.2.2 Deterministic Finite State Machine Against Multi-step Attacks

At the present time, research has been conducted to implement the DFSM as a mechanism for detecting multi-step attacks. The DFSM [112] is a computational model of system behavior that has a restricted number of states. In other words, those systems that have states which could be represented as disjoint sets. The beneficial effect of additions to the DFSM against multi-step attacks is to detect the multi-step attacks in the levels of the stage (before they posed a really harmful step). The DFSM had the following properties [113, 112]:

- The DFSM could be visualized graphically and easily tested.
- The system states changes from the current state to the next state based on the current state-transition.
- The events and conditions caused the state-transitions between the predefined system states.
- The system could not be in more than one state at the same time.

The general form of the DFSM [114] can be described by a 5-tuple expressed in Equation (15).

$$M = (Q, \Sigma, \delta, q_0, F), \tag{15}$$

where Q presents the set of finite system states, Σ shows the alphabet system inputs, δ presents the predefined transitions function, q_0 shows the initial system state and F indicates the final or accepted system state.

Branch et al. [103], propose an approach to detect the DoS multi-step attack based on the DFSM. The time intervals between specific alert correlations were used to enhance the accuracy

rate of the DFSM. The multi-step attack's signature (pattern) was defined as a sequence of events. The proposed detection approach's general structure includes several important procedures such as data filtration, event generators, and rule generators. The final state of the proposed detection approach indicates if the DoS multi-step attack has been completed, or not. The proposed approach was tested and evaluated using the DARPA dataset which is a benchmark dataset for different multi-step attack scenarios. The proposed DFSM approach was able to successfully detect the DoS multi-step attack within the DARPA dataset. The work of Sekar et al. in [115] focuses on detecting the multi-step attack based on the system call parameters. The system call parameters were used as the proposed approach's input parameters. These parameters were extracted using the Program Counter (PC) function. The normal behavior was defined as a sequence pattern of system call parameters. The PC function's system call parameters were adapted as states. The system call parameters were chosen to move the system from the current state to the next state. The proposed approach follows any sequence of system call parameters that did not follow the predefined standard normal behavior. The simulated environment indicates that the proposed detection approach works well to detect the multi-step attack. There are other works that adapted the normal behavior pattern to detect the multi-step attacks.

Treurniet et al. in [116], study the simulated network profile's normal behavior in order to detect the multi-step attack, implementing the DFSM as a detection model. The proposed model was applied to monitor any new transitions or events which did not follow the predefined pattern of normal behavior. It operates the TCP flags as input parameters to move the system from the current state to the next system state based on predefined rules. The proposed detection model was tested and evaluated based on the benchmark DARPA dataset, "week1". Subsequently, the proposed DFSM model successfully detected the abnormality connections that were not following the patterns of normal behavior.

There are other hybrid DFSM solutions too. The work of Han et al. in [117], proposed a hybrid model for detecting multi-step attacks called the Adaptive Time-dependent Finite Automata (ATFA). The general structure of ATFA was implemented based on the time-dependent finite automata. The ATFA model consists of two phases. In the first phase, the time series of the network profile is analyzed in order to define the normal and abnormal patterns (training phase). In the second phase, the Hsiaos sequential approach is applied to determine the causal relation between the series of packets. The Hsiaos sequential approach was used to define the sequence series of packets which appeared as a multi-step attack. The ATFA model was tested and evaluated using the DARPA benchmark dataset and was found to work well for detecting multi-step attacks within the simulated environment.

Branch et al. [103], continued the work of Vigna et al. in [118] which proposed the STAT model as a detection mechanism. Branch et al. then extend their work to include the NeSTAT model which was adapted to be used with the DFSM as a detection mechanism. The aim of this extension is to define the different types of multi-step attacks as state transition scenarios. This extension defines the different types of multi-step attacks as state transition scenarios. The proposed detection model assumes that the initial system state is the normal state. The abnormality

patterns are then defined as a sequence of actions. These actions are responsible for moving the system from a normal state to a compromised state. The authors applied the formal models of the attacks' scenarios as state transition diagrams. Thus, in the early stage of the NeSTAT model (analyzer stage), the attack scenario should be extracted in its precise order. In the analyzer stage, the DoS multi-step attack, UDP/TCP spoofing and remote buffer overflows have been defined and illustrated as state transition scenarios. The NeSTAT was able to detect the previously discussed types of multi-step attacks.

Some other works applied the DFSM against the Transmission Control Protocol (TCP) flooding attack. Gemona et al. [119], focus on detecting the TCP flooding attack which is a type of DoS attack. The TCP flood attack's sequence of events was implemented on the proposed DFSM model. The proposed DFSM model acts as a passive monitoring system for the TCP packets. The model consisted of three parts: monitoring, modeling, and detection. The modeling part was performed by determining the connections and defining the system states (SYN/ACK, SYN/Received, and ACK), meanwhile the detection part determined a large number of SYN/Received states. The results showed that the DFSM model was able to detect the TCP flooding attack within the simulated test-bed environment.

7.3 Fuzzy Automaton Based Detection Model Architecture

The multi-step attack's detection mechanisms in subsection (7.2) provide convincing contributions and show support for the persistent need to detect and predict multi-step attacks at their onset before they pose serious harm. However, these methods share some common disadvantages, summarized as follows:

- Their detection mechanisms applied the binary decision which supports the boundary problem, a constant challenge for implementing an efficient IDS detection mechanism [5, 6]. Herein, there are no clear boundaries between normal and intrusion traffics.
- The studied detection mechanisms did not determine the level of degree of system states; they only applied a binary decision to recognize the system state and to define the normal and intrusion traffic.
- They adopt a large amount of expert knowledge either for defining the complete attack scenarios or for defining the pre and post conditions in a precise order.
- The system could not be in more than one state at the same time while using the DFSM [112, 113]. Therefore, the detection mechanism could only follow a single path of event change state.

In response to these issues, in this dissertation, a novel model for detecting and preventing the multi-step attacks by using the fuzzy automaton and the FRI based reasoning is suggested. The reasons for using the fuzzy automaton and the FRI based reasoning are summarized as follows:

- The integration of a fuzzy system and automaton theory can form the concept of fuzzy automaton. This integration allows a discretely defined state-machine to act on continuous universes.

- The fuzzy system effectively smoothes the boundary between normal and intrusion traffics, effectively avoiding the binary decision.
- The fuzzy automaton detection mechanism presents the system states as a vector of membership values allowing the system to be in more than one state at the same time. As a result, the fuzzy automaton could follow multi-paths of intrusion-state changes.
- The proposed detection mechanism adapts FRI based reasoning instead of using classical inference methods. This simplifies rule definition because the missing state transition rules are interpolated by the reasoning (FRI) mechanism. In other words, the FRI reasoning mechanism can produce results even when some situations are not explicitly defined in the fuzzy rule-based knowledge representation.
- The fuzzy automaton detection mechanism did not involve a large knowledge base. Herein, there is no need to define the pre and post conditions of the attack scenario in a precise order. The fuzzy automaton detection mechanism directly predicts the most plausible intrusion goal by utilizing the available historical data.

Automata theory [120] is defined as the analytical study of abstract systems to solve computational problems. The integration between the fuzzy system and automaton theory results in a fuzzy automaton. This incorporation offers the ability to handle the computational challenges for both discrete and continuous spaces. The fuzzy automaton implemented based on the strengths of two paradigms, the automata, and the fuzzy system. Fuzzy systems are being implemented more frequently in different application areas. Fuzzy systems present comprehensive approximate reasoning results for the system's computational problems. Furthermore, they provide the required extension of the binary decision problem to the continuous truth value [5]. The general definition of the fuzzy automaton [121] is presented as a 6-tuple, illustrated in Equation 16.

$$\tilde{F} = (Q, \Sigma, \delta, R, Z, \omega) \tag{16}$$

Where Q is the finite set of the system states, Q= $\{q_0, q_1, \dots, q_k\}$. Σ is the finite set of the input symbols, $\Sigma = \{x_0, x_1, \dots, x_n\}$. δ is the fuzzy transition function, it is used to map the current system state to the next system state based on the finite set of inputs, $\delta: Q \times \Sigma \times Q \rightarrow (0,1]$. R shows the initial system state $\tilde{F}, R \in Q$. Z presents the finite set of output, $Z = \{Z_0, Z_1, \dots, Z_k\}$. Finally, ω presents the output mapping function which is responsible for mapping the fuzzy states into the output set, $\omega: Q \times \Sigma \rightarrow Z$.

In the fuzzy automaton, the system states, inputs and outputs are all presented as fuzzy sets. The predefined fuzzy states had a degree of membership values. Contrary to other state machines (deterministic, non-deterministic and probabilistic), the transition function was interpreted as a fuzzy transition function. In addition, the transitions between different states occurred based on the predefined fuzzy rules. In [122], the general definition of the fuzzy automaton was extended as shown in Equation 17.

$$\tilde{F} = (S, X, \delta, P, Y, \omega) \tag{17}$$

Where S is the finite set of fuzzy system states, $S = \{m_{s_1}, m_{s_2}, \dots, m_{s_k}\}$. X is the finite set of dimensional input values, $X = \{x_0, x_1, \dots, x_n\}$. δ is the fuzzy transition function, it is used to map the current state to the next state based on the finite set of inputs, $\delta: S \times X \to S$. P shows the initial fuzzy state of $\tilde{F}, P \in S$. Y is the finite set of output dimensional vectors, $Y = \{Y_0, Y_1, \dots, Y_k\}$. Finally, ω presents the output mapping function which is responsible for mapping the fuzzy states based on input values to the output set, $\omega: S \times X \to Y$.

The fuzzy automaton detection approach consists of six major components. These components are listed as follows:

- Setting up the finite fuzzy system states (S).
- Setting up the initial system state (P), assumed to be in the normal state.
- Defining the possible system input values (X). These values depend on which type of multi-step attack could be detected. The input values of the fuzzy automaton detection approach presented as a set of system observations $(i_1, i_2, \dots, i_n)$.
- Defining the fuzzy state-transition function δ which is used to map the current system state to the next system state based on the observations, $\delta: S \times X \to S$.
- Defining the finite set of system outputs, $Y = \{Y_0, Y_1, \dots, Y_k\}$.
- Defining the output mapping function $(\omega: S \times X \to Y)$ which is responsible for mapping the fuzzy states based on input values to the output.

The suggested fuzzy automaton detection mechanism adapts FRI (FIVE) method [72, 73, 74]. The FRI (FIVE) method is used to simplify the rule definition and to interpolate the missing state-transition rules. Contrary to the classical reasoning methods, the FRI methods offer the interpolated conclusion even when some situations are not explicitly defined [14]. Fig. 31.shows the general architecture of the fuzzy automaton detection mechanism using the previous six major components.

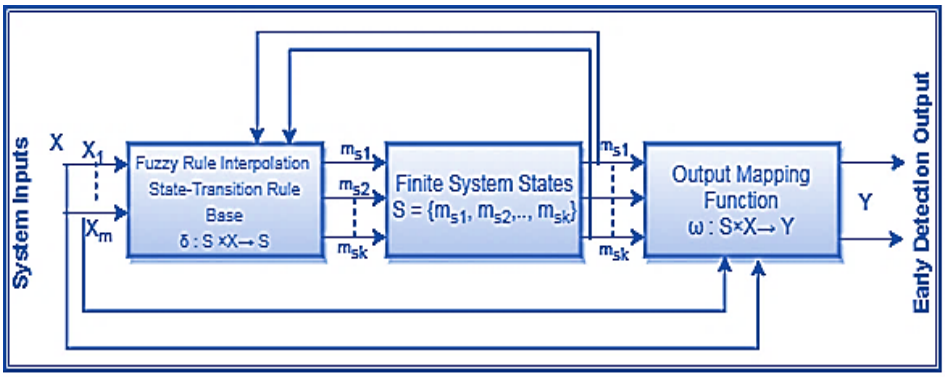

Fig. 31. The Fuzzy Automaton Detection Mechanism Architecture

The suggested fuzzy automaton based detection method consisted of four system states: $S = \{N, A, P, C\}$. These states are similar to those used in [123] which was defined as follows:

- Normal(N): the system behavior is in normal mode and there are no attempts to attack.
- Attempt(A): there are different attempts to gather information about the system in legal ways (different probe tools are launched).
- Prerequisites(P): malicious activity has commenced and the multi-step attack is in the process of launching its final step of the attack.
- Compromise(C): the multi-step attack has been completed successfully. The system is completely infected.

Fig. 32. presents the graph of the system states within the fuzzy automaton detection mechanism. The graph is fully connected to indicate that the transition (between states) may occur from any system to any system state.

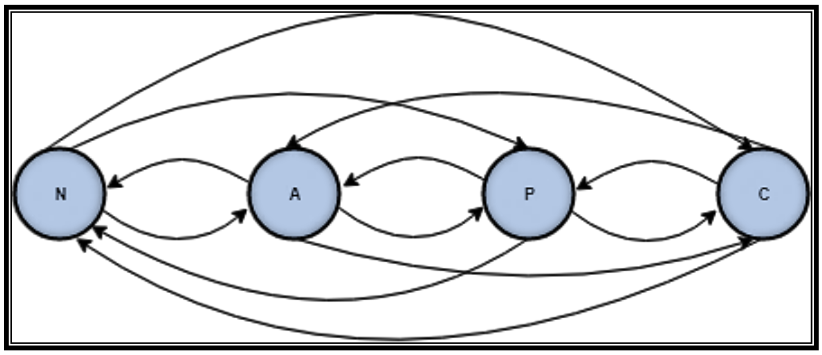

Fig. 32. System States of The Fuzzy Automaton Detection Mechanism

The fuzzy automaton detection mechanism focuses on the initial steps of the multi-step attack to prevent the launch of any further attack steps. Suppose that, there is a multi-step attack with n+m steps to be launched successfully. The fuzzy automaton detection mechanism focuses on predicting the multi-step attack penetrations within the period step (1) and step (n). Fig. 33. presents the concentration intents of the fuzzy automaton detection mechanism. The multi-step attack may be detected early because it built upon different preliminary phases that can be distinguished from one another.

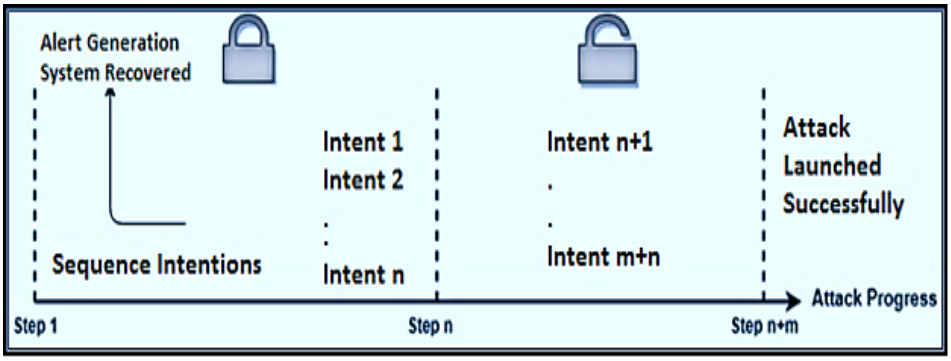

Fig. 33. The Concentration Intents of Fuzzy Automaton Detection Mechanism

7.4 Summary

The integration of a fuzzy system and automaton theory can form the concept of fuzzy automaton. This integration allows a discretely defined state-machine to act on continuous universes and handle uncertainty in applications like IDS. The typical IDS detection mechanisms are targeted to detect and prevent single-stage attacks. These types of attacks can be detected using either a common convincing threshold or by pre-defined rules. However, attack techniques have changed in recent years. Currently, the largest proportion of attacks performed, are multi-step attacks.

This chapter introduces a novel model to detect and prevent the multi-step attack. The suggested model assumes to able not only to detect but also early detect the multi-step attack in stages, where the planned attack is not fully elaborated and hence less harmful. The early detection of multi-step attacks also allows the administrator to take the necessary actions in time, to mitigate the potential threats.

8. Implementation of the Fuzzy Automaton Based Intrusion Detection

In this chapter, the creation and the validation of the suggested fuzzy automaton based multi-step attack detection mechanism is presented and discussed.

8.1 The Validation Methodology for the suggested Fuzzy Automaton based Intrusion Detection Approach

To evaluate the proposed detection model in practice, the DARPA 2000 attack scenarios dataset LLDOS1.0 was used [107]. It seems to be a proper benchmark for the multi-step attack. It consisted of different multi-step attack scenarios. One of the benefits of using the DARPA 2000 dataset is that it contains a detailed truth table that allows for the obtained results to be checked. Moreover, most of the IDS detection approaches have applied this dataset for testing and evaluating processes [103]. This work extracts the first attack scenario which was a DDOS multi-step attack.

According to the extracted DDOS multi-step attack scenario, the attacker aimed to install the DDOS multi-step attack on any computer within the target network. The attack was based on five steps [124]. It lasted three hours and was performed for these subnets 172.16.112.0/24, 172.16.113.0/24, 172.16.114.0/24 and 172.16.115.0/24. Consequently, there were three hosts infected by the DDOS multi-step attack. These hosts were 172.16.115.20, 172.16.112.50 and 172.16.112.10. Table 25 illustrates the five sequence steps of the first DARPA attack scenario.

Table 25. The Sequence Steps of The DARPA Attack Scenario

Step	Name	Time
1	IP Sweep	09:45 - 09:52
2	Sadmind	10:08 - 10:18
3	Break-In	10:33 - 10:34
4	Installation	10:50
5	Launching	11:27

- Step (1): The attacker sends a large number of Internet Control Message Protocol (ICMP) echo requests in this sweep and waits for the echo replay to obtain the live IP addresses (hosts).
- Step (2): The result of the step (1) is the list of live hosts. Every live host in the previous step was probed to define the hosts running the sadmind service. The sadmind investigation was applied using sadmind exploit software and ping command.
- Step (3): The result of step (2) is the list of live hosts running the sadmind service. The break-in script was executed for every live host. Break-in script tries the sadmind remote to root access. During the period (10:33 to 10:34) there were 6 break-in attempts.
- Step (4): The result of step (3) is the list of infected hosts (three hosts were infected). Herein, the break-in script executed the remote to root successfully. Therefore, the attacker

had the required access to install the DDOS multi-step attack for these infected hosts.

- Step (5): The attacker launched the DDOS multi-step attack using the TELNET login.

The simulated DDOS multi-step attack scenario lasted for a total of (11836 seconds). Table 26 presents the DDOS multi-step attack phases according to the simulation time.

Table 26. The Phases During The DDOS Multi-step Attack

Attack States	Description	Time in Seconds
IP Sweep	Step 1 of Attack	1500 - 1920
Sadmind	Step 2 of Attack	2880 - 3480
Break-in	Step 3 of Attack	3650 - 5200
Installation	Step 4 of Attack	5400 - 6500
launching	Step 5 of Attack	7620 - 11836

The DARPA attack scenario dataset LLDOS1.0 was reformulated by extracting the values of the main features and labeling the data according to the existing literature results [124, 123]. Table 27 shows the extracted features values for the entire DARPA attack scenario LLDOS1.0. The fuzzy automaton detection mechanism's fuzzy system states are defined as follows: $S = N, A, P, C$. The initial state of the fuzzy automaton detection mechanism is assumed to be in the normal state (N). The N state indicates there are no attack attempts or privacy violations; the system is in normal mode. The A state indicates that there are some attempts to gather and probe for information using IP Sweep and sadmind. The P state indicates that malicious activity has been initiated by running the break-in and installation scripts. The C state indicates that the system has been completely infected; the multi-step attack has been launched successfully.

The fuzzy automaton based intrusion detection mechanism's input parameters (the set of observations) are the reformulated DARPA attack scenario. Due to a large number of extracted features and for the sake of simplification, one-eighth of the total number of features was selected as an input parameter. Some of the literature works suggested applying the intersection operation between multi-features selection algorithms for more accurately as in [125]. For this reason in this work, the relevant features were selected as in [125] based on the intersection operation between the Gain Ratio (GR) algorithm, the Information Gain (IG) algorithm, and the ReliefF (RF) algorithm. Those features that fulfill the intersection criteria, as shown in Equation 18, were selected as the proposed detection mechanism's input parameters. Table 28 shows the relevant input parameters for the fuzzy automaton detection mechanism.

$$GR \cap IG \cap RF \tag{18}$$

Table 27. The Extracted Features of DARPA LLDOS 1.0

Index	Feature Name	Description
1	host a	The source machine
2	host b	The destination machine
3	port a	Source port
4	port b	Destination port
5	total packets a2b	The total number of packets exchanged between the host a and host b.
6	total packets b2a	The total number of packets exchanged between the, host b and host a.
7	resets sent a2b	The count of reset (RST) packets sent from host a and host b
8	ack pkts sent a2b	The total number of ACK packets seen between host a ad host b
9	ack pkts sent b2a	The total number of ACK packets seen between host b ad host a
10	pure acks sent a2b	The total number of ACK packets without payload and any SYNFIN/RST flags bits set in the connection from hots a and host b.
11	pure acks sent b2a	The total number of ACK packets, without payload and any SYNFIN/RST flags bits set in the connection from hots b and host a.
12	outoforder pkts a2b	The total count of all the packets that were appeared to arrive out of order between host a and host b
13	outoforder pkts b2a	The total count of all the packets that were appeared to arrive out of order between host b and host a
14	pushed data pkts a2b	The total number of packets with push flag bits between host a and host b
15	pushed data pkts b2a	The total number of packets with push flag bits between host b and host a
16	adv wind scale a2b	It is an indicator if the, window scale was used in the connection.
17	adv wind scale b2a	It is an indicator if the window scale was used in the connection
18	sack pkts sent	In connection between host a and host b, the host a sent SACK in the SYN packet opening the connection, Y is printed, else N is printed
19	sack pkts received	In connection between host b and host a, the host a sent SACK in the SYN packet opening the connection, Y is printed, else N is printed
20	mss requested a2b	In connection between host a and host b, Maximum Segment Size (MSS) requested as a TCP option in the SYN packet opening the connection.
21	mss requested b2a	In connection between host b and host a, Maximum Segment Size (MSS) requested as a TCP option in the SYN packet opening the connection.
22	max segm size a2b	In connection between host a and host b, the maximum segment size observed during the lifetime of the connection.
23	max segm size b2a	In connection between host band host a, the maximum segment size observed during the lifetime of the connection.
24	min segm size a2b	In connection between host a and host b, the minimum segment size observed during the lifetime of the connection.
25	min segm size b2a	In connection between host b and host a, the minimum segment size observed during the lifetime of the connection.
26	min win adv a2b	The minimum window advertisement sent from host a to host b
27	min win adv b2a	The minimum window advertisement sent from host b to host a
28	initial window bytes a2b	The total number of bytes sent in the initial window from host a to host b
29	initial window pkts b2a	The total number of packets sent in the initial window from host b to host a
30	ttl stream length a2b	The theoretical stream Length (TTL). This is calculated as the difference between the sequence numbers of the SYN and FIN packets, giving the length
31	ttl stream length b2a	The theoretical stream Length (TTL). This is calculated as the difference between the sequence numbers of the SYN and FIN packets.
32	missed data a2b	It is an indicator of the missed data between host a and host b using calculated the difference between the ttl stream length and the unique bytes sent.

Table 28. The Relevant Input Parameters

Parameter	Description
MSS Request	In the connection between host a and host b, maximum Segment Size (MSS) requested as a TCP option in the SYN packet opening the connection.
Pure A2B	The total number of ACK packets without payload and any SYNFIN/RST flags bits set in the connection from hots a and host b.
Pure B2A	The total number of ACK packets without payload and any SYNFIN/RST flags bits set in the connection from hots b and host a.
Total bytes between A2B	The total number of packets exchanged between the host a and host b.

8.2 The State-transition Rules

The state transition rule base definition is based on expert heuristic. An efficient tool for easily creating an expert fuzzy rule-base is the Fuzzy Behavior Description Language (FBDL) [126]. It is a declarative language providing a simple structure for defining the state-transition rule-base size in a humanly readable form, closely resembling the original verbal form. Regarding the fuzzy declarative language, there are two conditions used to define the state transition rule-base:

- Each rule-base should have a unique name.
- The name of the rule-base must be the same as the name it's consequent.

The aim behind using the FBDL is to present a simple form for defining the state transition rule base which could be more readable and understandable by a human. The connections between the rule-bases can be defined by these unique names of the antecedents and the consequent. The FBDL considered a structured language that includes different blocks. Each block can be opened simply using the valid keyword which defined the block type. Furthermore, each block should be closed using the end keyword. FBDL allows the expert to define the description for each block optionally. The main block in the FBDL is the rule-base block which is illustrated in Fig. 34. At the beginning, the method block relies on which of the consequent calculation methods should be used based on its corresponding parameters. The current version of FBDL supports two methods which are the FIVE and direct Shepard interpolation.

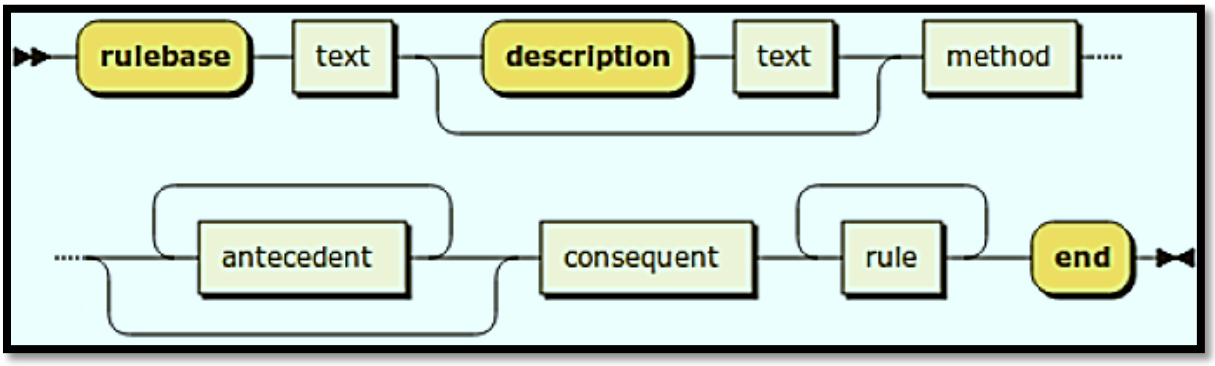

Fig. 34. Rule base Definition [126]

The state transition rule base definition within the FBDL presented by the rule-base class. The stored class had the method name, and also its required parameters. The connections between all state-transition rule base automatically achieved using the unique names of rule- bases and input values. The antecedent names considered as inputs of the system.

It is worth mentioning that, in the classical reasoning methods, the size of the state-transition rule-base grows exponentially with the number of the inputs (observations). For this reason, the proposed detection mechanism adapts the FRI, as it can effectively reduce the size of the state-transition rule-base. The fuzzy automaton detection mechanism has continuous states which are presented as a vector of membership values. These states were defined in the fuzzy declarative language as follow:

```
Universe "Normal State"
Description "The Degree of Normal State"
        "Low"   0 0
        "High" 1 1
End
Universe "Attempt State"
Description "The Degree of Attempt State "
        "Low"   0 0
        "High" 1 1
End
```

```
Universe "Prerequisite State"
Description "The Degree of Prerequisite State "
        "Low"   0 0
        "High" 1 1
End
Universe "Compromise State"
Description "The Degree of Compromise State "
        "Low"   0 0
        "High" 1 1
End
```

The applications of the FRI methods are beneficial in the IDS application area [5]. Using FRI methods, expert knowledge can be used as the basis of fuzzy rules. In the suggested FRI fuzzy automaton detection mechanism, the rules are not strict; the expert can sort some of the known cases only. Most important cases and scenarios can be sufficiently defined by using the proposed fuzzy declarative language. The description contains the definition of ranges (as universes) and the rules (in the form of rule-bases). The definition of the universes describes non-linear scaling on the considered input and output dimensions. Experts must define language symbols which may be similar to the domain-specific terms. Therefore, the FRI method formalizes the expert knowledge to the form, which can be interpreted and evaluated automatically by the inference engine. Using the language symbols allows the results to more closely resemble the natural language equivalent.

The universe definitions of the observations of the proposed detection mechanism are defined based on the expert knowledge and presented in the fuzzy declarative language as follows:

```
Universe "Pure_A2B "
"VSmall"    0 31
"Small"      31 69
"Medium"   161 69
"Large"      2430 616
"VLarge"    8845 2430
End
Universe "Pure_B2A "
"Low"    0 380
"High"   380 780
End
```

```
Universe "Total_A2B "
"Small"    1 8400
"Large"    8400 17693
End
Universe "Mss_Request"
"VSmall"    1 1987
"Small"      2200 2700
"Medium"   3200 5350
"Large"      6500 8000
End
```

The state-transition rule-bases were defined based on expert knowledge. Fourteen state transition rules were constructed. For example, the attempt state rule definitions and the prerequisite state rule definitions presented as follows:

```
Rulebase "Attempt_State"
Rule
"High" when
"Mss_Requested" is "Medium" and
"Pure_A2B" is "Meduim"
end
Rule
"High"   when
"Pure_A2B" is "Small" and
"Mss_Requested" is "Medium"
end
Rule
"High" when
"Pure_B2A" is "High" and
"Mss_Requested" is "Medium"
end
Rule
"Low" when
"Mss_Requested" is "Small"
end
```

```
Rulebase "Prerequisite_State"
Rule
"High" when
"Mss_Requested" is "Large" and
"Pure_A2B" is "VSmall"
end
Rule
"High" when
"Mss_Requested" is "Medium" and
"Pure_A2B" is "Small" and
"Pure_B2A" is "Low"
End
Rule
"Low" when
"Mss_Requested" is "VSmall"
end
Rule
"Low" when
"Mss_Requested" is "Large"
end
End
```

8.3 Experiments and Results

According to the way, as the FRI (FIVE) method calculates the conclusion, the evaluation process of rule bases can be described in a bottom-up manner. In the first step, the inference engine calculates the observations' distances from the defined symbols on the given universes. Subsequently, the rules' distances are evaluated. In the considered configuration, the rule's distance is the normalized Euclidean norm of the included symbol distances. The measure of the rule-base was obtained by the Shepard interpolation (inverse distance weighting) of the rule distances and their consequent values. Fig. 35. presents the simulation environment and indicates, that the system's initial state is normal. The simulation environment can be accessed through [127].

Fig. 35. Fuzzy Automaton Detection Mechanism Simulation Environment

The proposed detection mechanism generated intelligible results due to its fuzzy nature, subsequently allowing the degree of the system state to be determined and for the system to be in more than one state at the same time. Table 29 presents the proposed detection mechanism's output response in case of intrusion instances. Unlike DFSM and HMMs, the system states within the proposed detection mechanism are presented as a vector of membership values. This could benefit administrators because it helps them to understand the current security status and to mitigate future risks by forecasting the upcoming system state.

Table 29. The Output Response of The Suggested FRI Fuzzy Automaton Based IDS

Input Parameters			
	Instance 1	**Instance 2**	**Instance 3**
Mss request	4200	6300	3869
Pure A2B	110	96	141
Pure B2A	614	750	688
Total A2B	10536	9365	12369
The Proposed Detection Method Output			
	Output 1	**Output 2**	**Output 3**
Normal	0.270791	0.085362	0.126221
Attempt	0.919518	0.212365	0.932641
Prerequisite	0.446831	0.926831	0.482133
Compromise	0.157381	0.357381	0.198752

The fuzzy automaton detection mechanism was tested and evaluated in the following durations of the DDOS multistep attack: (15-1062 seconds), (1800-2786 seconds), (37505191 seconds) and (8210-10342 seconds). These durations were chosen to verify the performance of the fuzzy automaton detection mechanism in order to detect the DDOS multi-step attack in its early stages before it posed a severe risk.

The fuzzy automaton detection mechanism was evaluated using 5639 observations. The first detection was obtained by the fuzzy automaton detection mechanism at 1800 seconds, 2 minutes before the attacker completes the works in step 1. The second detection was obtained at 3750 seconds, 24 minutes before the attacker completes the works in step 3. The third detection was at 8210 seconds, 60 minutes before the attacker completes the works in step 5. Thus, early detection of the multi-step attack gives administrators time to take the necessary actions to mitigate any future risk from this type of attack. The IDS detection mechanism's standard performance measure is typically performed using both the Receiver Operating Characteristic (ROC) and the confusion matrix [103], where the ROC shows the trade-off between sensitivity and specificity [128]. In keeping with the standard measure of most other IDS detection mechanisms, Fig. 36. shows the evaluation performance, with the ROC curve, for the fuzzy automaton detection mechanism states.

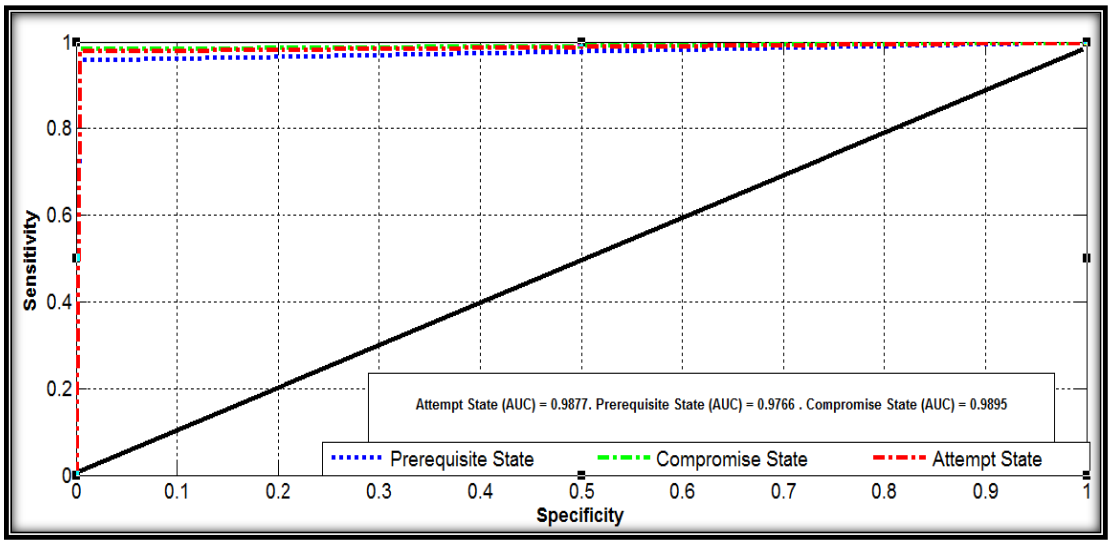

Fig. 36. The ROC Curve for The Fuzzy Automaton Detection States

Table 30 illustrates the confusion matrix obtained during the evaluation process. The results reflected that the fuzzy automaton detection mechanism obtained a 97.836% overall accuracy rate. Furthermore, the implemented experiments demonstrated that the fuzzy automaton detection mechanism was able to detect the DDOS multi-step attack within its early stages, using the DARPA dataset. Therefore, the early detection of the multi-step attack could be beneficial for the administrator to perform the required mitigation actions.

Table 30. The Confusion Matrix of The Evaluation process

	Normal	**Attempt**	**Prerequisite**	**Compromise**	**Overall Observations**	**Precision**
Normal	1045	2	2	0	1049	**99.619%**
Attempt	13	985	8	0	1006	**97.913%**
Prerequisite	0	58	1312	1	1371	**95.697%**
Compromise	0	0	38	2175	2213	**98.283%**
Truth Overall	1058	1045	1360	2176	5639	
Overall Accuracy	**97.836%**					

For summarizing the results of the benchmark based tests, it can be stated, that the suggested FRI fuzzy automaton based IDS could be a promising mechanism for detecting multi-step attacks. The FRI fuzzy automaton based detection mechanism can be characterized by the following key points:

- The fuzzy automaton detection mechanism offers the system states as a vector of membership values.
- Unlike the DFSM, the system can be in more than one state at the same time, thereby allowing the fuzzy automaton detection mechanism to follow more than one path of system states changes.
- Adapting the FRI (FIVE) method offers interpolated results even when lacking knowledge-based representation. In other words, The FRI (FIVE) method interpolates the results even when some of the state transition rules are missing.
- The fuzzy automaton detection mechanism produces verbal detection results which can be more easily understood by administrators.
- The fuzzy system extends the binary decision to the continuous space, smoothing the boundaries and offering a solution to the boundary problem in addition to generating more comprehensible results.
- The fuzzy automaton detection mechanism can detect the DDOS multi-step attack within its early stages, using the DARPA dataset. Thus, early detection could help the administrator mitigate this type of attack.
- The proposed detection mechanism's strength is based on combining fuzzy automaton and FRI based reasoning. Thus, the fuzzy system effectively smooths the decision boundary between normal and intrusion traffics, avoiding the binary decision. And the FRI based implementation is eliminating the need for the complete state-transition rule-base definition.

The main characteristics of the proposed FRI fuzzy automaton IDS and the other state machine detection mechanisms are compared in Table 31.

Table 31. The Main Characteristics of Some Widely Used Detection Methods

	HMM Detection Mechanism	DFSM Detection Mechanism	Fuzzy Automaton Detection Mechanism
Binary Decision	Yes	Yes	Approximated
System State	Discrete	Discrete	Continuous
Uncertainty	Not Applicable	Not Applicable	Applicable
Rules	Statistical	Knowledge Base	Knowledge Base

The proposed FRI fuzzy automaton IDS eliminates the boundary decision problem, which is considered as a constant challenge because in real situation there are no clear boundaries between the normal and intrusion traffics. In addition, implementing the FRI (FIVE) method instead of the classical reasoning methods for the reasoning part helps to reduce the total number of state-transition rules (simplification) and offers interpolated results even if the knowledge representation is incomplete.

8.4 Summary

This chapter has implemented the novel method for detecting multi-step attacks by combining the (FIVE) FRI reasoning, with the fuzzy automaton. The strength of fuzzy automaton is derived from two paradigms, the theory of automata and the fuzzy system. The reasoning part of the proposed detection mechanism adopts the FRI (FIVE) method instead of the classical reasoning methods. This decreases the total number of the intrusion state transition fuzzy rules needed to be defined (simplification) and also offers interpolated results even when the knowledge representation is incomplete. The state-transition rule-base was defined using an open-source fuzzy declarative language. This provides a simple way for defining the state-transition rule-base in a humanly readable form, which is closely resembling the original verbal form.

The experiments applied on a multi-step attack benchmark dataset are demonstrated, that the proposed detection mechanism can achieve an acceptable overall detection rate. It was able to successfully detect the multi-step attack within the test-bed environment at an early stage of the attack. One of the main benefits of the proposed detection method is its ability to present the system states, as a vector of membership values. It could also extend the binary decision to the continuous space which smooths the boundaries and offers a solution to the boundary problem. Moreover, the proposed detection method allows the system to be in more than one state at the same time. Consequently, the fuzzy automaton detection mechanism could be a suitable detection mechanism for detecting multi-step attacks at their early stage, before they cause a serious risk and harm.

Thesis III.: The FRI (FIVE) based fuzzy automaton could be a suitable model for detecting and preventing the multi-step attacks. By offering interpolated conclusion even for situations that are not explicitly defined, the FRI method instruments the fuzzy automaton to be able to act on partly defined state transition rule-base. The integration of fuzzy state machine and fuzzy rule

interpolation allows a discretely defined state-machine to act on continuous universes and handle uncertainty in applications like intrusion detection systems.

Thesis IV.: The FRI (FIVE) based fuzzy automaton, with an expert-defined state-transition rule-base given in Fuzzy Behavior Description Language (FBDL), was suitable for detecting and preventing the multi-step attacks in stages. The state-transition fuzzy rule-base is required for the FRI (FIVE) based fuzzy automaton intrusion detection model, and can be easily defined using the FBDL.

The results introduced in this chapter are supporting the statements of Thesis III and Thesis IV and published in [130].

9. Contribution and Future Research Direction

This dissertation contributes to the field of fuzzy systems (especially fuzzy rule interpolation), intrusion detection system and also fuzzy state machine.

A novel method, IDS model-based fuzzy rule interpolation along with a novel method to detect abnormalities by combining the Fuzzy Interpolation based on the Vague Environment (FIVE) FRI reasoning with the Management Information Base (MIB) parameters have been constructed. This method not only allows the intrusion detection system to be used in continuous spaces but makes it possible to use a sparse fuzzy rule base. This way, the overall rule base size significantly smaller. Because of its fuzzy rule base knowledge representation nature, it can be easily adapt expert knowledge, and also be suitable for predicting the level of degree for threat possibility. Furthermore, the combining of the (FIVE) FRI reasoning with the Management Information Base (MIB) parameters is a promise detection method to mitigate the network intrusions. See Thesis I. and Thesis II. below, and also the fifth and sixth chapters for a detailed description of these methods. The incorporated fuzzy rule interpolation method, FIVE, has been successfully constructed specifically for the intrusion detection systems, taking the performance of these methods to a higher level.

Furthermore, this dissertation proposes a novel model for detecting the multi-step attacks. The proposed model was built upon the fuzzy rule interpolation-based fuzzy state machine. Based on the proposed model which has a simple rule-based knowledge representation format and where the completeness of the rule-base is not required. This model does not only allows the intrusion detection system to be used in continuous spaces, but also makes it possible to be in more than one state at the same time. Additionally, the proposed model interpolates the results even when some of the state transition rules are missing. Details on the structure and implementation of the model can be found in chapters seven and eight, and also see thesis III and thesis IV. below.

Future research and investigation into the possibilities of adapting the IDS based fuzzy rule interpolation in the Internet of Things (IoT) based smart environments seems promising. The IoT paradigm has recently evolved to incorporate different application areas. In the IoT environment, several heterogeneous devices are connected via different types of sensors. These wireless sensors beside the IPv6 added advantage to extend the IoT environment to serve many application areas. The heterogeneous devices within the IoT environment may have a different level of security. Some of the IoT devices have little or no security embedded into them. This deficiency could affect the availability of the IoT connected network and made several security flaws. Moreover, attackers continuously targeted the modern aspects of technology, and trying abusing these technologies using complex attack scenarios such as Botnet attacks. Due to the limited computing and storage capabilities of IoT devices and the specific protocols used, typical IDS may not be suitable for IoT environments. Therefore, the IDS based fuzzy rule interpolation could be a suitable alternative to mitigate IoT-related security attacks. This is due to its fuzzy nature, and its ability to render results even when faced with only partially completed fuzzy rules. Moreover, the results are rendered in a human-readable form.

Another direction for future research lies in investigating the potential for adapting the fuzzy rule interpolation for the Cyber Forensics system. The field of digital forensic is growing dramatically in parallel with the rise of computer crimes. Network forensics is a digital forensic in networked environments. It consists of a large amount of network traffic. All this traffic needs to be analyzed and investigated to define the digital evidence; however, not all of this information is useful as evidence. Therefore, the irrelevant information needs to be removed by an expert. The expert also is responsible to define some rules to decide which information could be used as digital evidence. Thus, there is an emerged need to design an automated system for analyzing the network forensics, and at the same time had the ability to deal with the issues associated with the deficiencies of the knowledge-based representation. Furthermore, generating the digital evidence with a level of severity could be beneficial for clarifying computer crimes. Also, it helps the expert to understand the current digital evidences in a more readable form. Therefore, the fuzzy rule interpolation reasoning methods could be suitable for use as an expert system capable of providing the forensics experts with the necessary information to successfully reduce the time, and cost of analyzing the network forensics, and generating the digital evidence in case of lacking knowledge-based representation.
The scientific results of the research presented in this work summarized as theses can be read in the followings:

Thesis I.: [5]

The FIVE based fuzzy rule interpolation model can be used in the IDS as a suitable inference method. Furthermore, the FRI inference system has yielded promising results when implemented as an IDS detection mechanism. Additionally, during the studies test application, the FRI inference system effectively decreased the rate of false positive values. Moreover, because of it's tendency for fuzzy rule based knowledge representation, it can easily adapt to expert knowledge, and be suitable for predicting the potential threat level.

Thesis II.: [129]

The FIVE based fuzzy rule interpolation for SNMP-MIB data based intrusion detection achieving acceptable results in an IDS detection mechanism. I concluded that, using this method there is no need to deal with raw traffic processing, which is time-consuming, and difficult to compute. The MIB parameters reflect the normal and abnormal nature of the network traffics. Furthermore, expert knowledge can be easily adapted by eliminating the need for creating a complete fuzzy rule base.

Thesis III.: [130]

The FRI (FIVE) based fuzzy automaton could be a suitable model for detecting and preventing the multi-step attacks. By offering interpolated conclusion even for situations that are not explicitly defined, the FRI method instruments the fuzzy automaton to be able to act on partly

defined state transition rule-base. The integration of fuzzy state machine and fuzzy rule interpolation allows a discretely defined state-machine to act on continuous universes and handle uncertainty in applications like intrusion detection systems.

Thesis IV.: [130]

The FRI (FIVE) based fuzzy automaton, with an expert-defined state-transition rule-base given in Fuzzy Behavior Description Language (FBDL), was suitable for detecting and preventing the multi-step attacks in stages. The state-transition fuzzy rule-base is required for the FRI (FIVE) based fuzzy automaton intrusion detection model, and can be easily defined using the FBDL.

Author's Publication

[5] Mohammad Almseidin and Szilveszter Kovacs. Intrusion detection mechanism using fuzzy rule interpolation. *Journal of Theoretical and Applied Information Technology*, 96(16):5473–5488, 2018. **Scopus Indexed [Q3].**

[129] Mohammad Almseidin, Mouhammd Al-kasassbeh and Szilveszter Kovacs. Fuzzy Rule Interpolation and SNMP-MIB for Emerging Network Abnormality. *International Journal on Advanced Science, Engineering and Information Technology*, vol. 9, no. 3, pp. 1-10, 2019. **Scopus Indexed [Q2].**

[130] Mohammad Almseidin, Imre Piller , Mouhammd Al-kasassbeh and Szilveszter Kovacs. Fuzzy Automaton as a Detection Mechanism for the Multi-Step Attack. *International Journal on Advanced Science, Engineering and Information Technology*, vol. 9, no. 2, pp. 17-31, 2019. **Scopus Indexed [Q2].**

[56] Ibrahim M Obeidat, Nabhan Hamadneh, Mouhammd Alkasassbeh, Mohammad Almseidin, and Mazen Ibrahim AlZubi. Intensive pre-processing of kdd cup 99 for network intrusion classification using machine learning techniques. *International Journal of Interactive Mobile Technologies*, 13(1), 2019. **Scopus Indexed [Q3].**

[57] Mohammad Almseidin, Maen Alzubi, Szilveszter Kovacs, and Mouhammd Alkasassbeh. Evaluation of machine learning algorithms for intrusion detection system. *In Intelligent Systems and Informatics (SISY), 2017 IEEE 15th International Symposium on*, pages 000277– 000282. IEEE, 2017. [Online]. Available: https://doi.org/ 10.1109/SISY.2017.8080566.

[131] Mouhammad Alkasassbeh and Mohammad Almseidin. Machine learning methods for network intrusion detection. *International Journal of Computer and Information Engineering* 12 : 8 p. 1 (2018).

[132] Mohammad Almseidin, Mouhammd Al-kasassbeh and Szilveszter Kovacs, Detecting Slow Port Scan Using Fuzzy Rule Interpolation. *In IEEE International Conference on New Trends in Computing Sciences (2019).*

References

[1] C Yuan. Research on multi-step attack detection method based on gct. Master's Thesis, Jilin University, Jilin, 2010.

[2] Yanxue Zhang, Dongmei Zhao, and Jinxing Liu. The application of baum-welch algorithm in multistep attack. The Scientific World Journal, 2014.

[3] Salem Benferhat, Tayeb Kenaza, and Aicha Mokhtari. A naive bayes approach for detecting coordinated attacks. In Annual IEEE International Computer Software and Applications Conference, pages 704–709. IEEE, 2008.

[4] Can Chen and BQ Yan. Network attack forecast algorithm for multi-step attack. Computer Engineering, 5(37):172–174, 2011.

[5] M. Almseidin and S. Kovacs. Intrusion detection mechanism using fuzzy rule interpolation. Journal of Theoretical and Applied Information Technology, 96(16):5473–5488, 2018.

[6] R Shanmugavadivu and N Nagarajan. Network intrusion detection system using fuzzy logic. Indian Journal of Computer Science and Engineering (IJCSE), 2(1):101–111, 2011.

[7] Swati Dhopte and NZ Tarapore. Design of intrusion detection system using fuzzy class-association rule mining based on genetic algorithm. International Journal of Computer Applications, 53(14), 2012.

[8] Lotfi A Zadeh. Fuzzy sets. Information and control, 8(3):338–353, 1965.

[9] Lotfi A Zadeh. Fuzzy algorithms. information and control, 12(2):94–102, 1968.

[10] SN Sivanandam, Sai Sumathi, SN Deepa, et al. Introduction to fuzzy logic using MATLAB, volume 1. Springer, 2007.

[11] Ebrahim H Mamdani and Sedrak Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. International journal of man-machine studies, 7(1):1–13, 1975.

[12] Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. In Readings in Fuzzy Sets for Intelligent Systems, pages 387–403. Elsevier, 1993.

[13] Zsolt Csaba Johanyák and András Szabó. Tool life modelling using rbe-dss method and lesfri inference mechanism. A GAMF Közleményei, Kecskemét, 22:17–28, 2008.

[14] Szilveszter Kovács. Fuzzy rule interpolation. In Encyclopedia of Artificial Intelligence, pages 728–733. IGI Global, 2009.

[15] Zsolt Csaba Johanyák and Szilveszter Kovács. A brief survey and comparison on various interpolation based fuzzy reasoning methods. Acta Polytechnica Hungarica, 3(1):91–105, 2006.

[16] P. Kazienko and P. Dorosz. Intrusion detection, 2003.

[17] Karen Scarfone and Peter Mell. Guide to intrusion detection and prevention systems (idps). Technical report, National Institute of Standards and Technology, 2012.

[18] O Adetunmbi Adebayo, Zhiwei Shi, Zhongzhi Shi, and Olumide S Adewale. Network anomalous intrusion detection using fuzzy-bayes. In International Conference on Intelligent Information Processing, pages 525–530. Springer, 2006.

[19] Mouhammd Alkasassbeh, Ghazi Al-Naymat, Ahmad BA Hassanat, and Mohammad Almseidin. Detecting distributed denial of service attacks using data mining techniques. International Journal of Advanced Computer Science and Applications, 7(1), 2016.

[20] Mohammed Alenezi and Martin J Reed. Methodologies for detecting dos/ddos attacks against network servers. In The Seventh International Conference on Systems and Networks Communications ICSNC, pages 92–98, 2012.

[21] Jaehak Yu, Hansung Lee, Myung-Sup Kim, and Daihee Park. Traffic flooding attack detection with snmp mib using svm. Computer Communications, 31(17):4212 – 4219, 2008.

[22] Gang Wang, Jinxing Hao, Jian Ma, and Lihua Huang. A new approach to intrusion detection using artificial neural networks and fuzzy clustering. Expert systems with applications, 37(9):6225–6232, 2010.

[23] Abdelaziz Araar and Rami Bouslama. A comparative study of classification models for detection in ip

networks intrusions. Journal of Theoretical & Applied Information Technology, 64(1), 2014.

[24] Md Al Mehedi Hasan, Mohammed Nasser, Biprodip Pal, and Shamim Ahmad. Support vector machine and random forest modeling for intrusion detection system (ids). Journal of Intelligent Learning Systems and Applications, 6(01):45, 2014.

[25] Dipali Patalau Gaikwad, Sandeep Jagtap, Kunal Thakare, and Vaishali Budhawant. Anomaly based intrusion detection system using artificial neural network and fuzzy clustering, International Journal of Engineering Research & Technology, 5 (02) 2012.

[26] Kevin Gurney. An introduction to neural networks. CRC press, 2014.

[27] Laurene V Fausett et al. Fundamentals of neural networks: architectures, algorithms, and applications, volume 3. prentice-Hall Englewood Cliffs, 1994.

[28] Leo Breiman. Random forests. Machine learning, 45(1):5–32, 2001.

[29] Sujay Apale, Rupesh Kamble, Manoj Ghodekar, Hitesh Nemade, and Rina Waghmode. Defense mechanism for ddos attack through machine learning. International Journal of Research in Engineering and Technology, 3(10):291–294, 2014.

[30] Ayman I Madbouly, Amr M Gody, and Tamer M Barakat. Relevant feature selection model using data mining for intrusion detection system. arXiv preprint arXiv:1403.7726, 2014.

[31] Zheng Zhang, Jun Li, CN Manikopoulos, Jay Jorgenson, and Jose Ucles. Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In Proc. IEEE Workshop on Information Assurance and Security, pages 85–90, 2001.

[32] Gary C Kessler. Defenses against distributed denial of service attacks. SANS Institute, 2002, 2000.

[33] Huy Anh Nguyen and Deokjai Choi. Application of data mining to network intrusion detection: classifier selection model. In Asia-Pacific Network Operations and Management Symposium, pages 399–408. Springer, 2008.

[34] Swati Paliwal and Ravindra Gupta. Denial-of-service, probing & remote to user (r2l) attack detection using genetic algorithm. International Journal of Computer Applications, 60(19):57–62, 2012.

[35] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on, pages 1–6. IEEE, 2009.

[36] P Amudha, S Karthik, and S Sivakumari. Classification techniques for intrusion detection-an overview. International Journal of Computer Applications, 76(16), 2013.

[37] Kdd cup 1999 data. 1999.

[38] A Ozgur and H Erdem. A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ Preprints, 4(e1954v1), 2016.

[39] Mr Kamlesh Lahre, Mr Tarun Dhar, Diwan Suresh, Kumar Kashyap, and Pooja Agrawal. Analyze different approaches for ids using kdd 99 data set. International Journal on Recent and Innovation Trends in Computing and Communication, 1(8):645–651, 2013.

[40] Fariba Haddadi, Sara Khanchi, Mehran Shetabi, and Vali Derhami. Intrusion detection and attack classification using feed-forward neural network. In Computer and Network Technology (ICCNT), 2010 Second International Conference on, pages 262–266. IEEE, 2010.

[41] Wafa' Alsharafat. Applying artificial neural network and extended classifier system for network intrusion detection. International Arab Journal of Information Technology (IAJIT), 10(3), 2013.

[42] Neeraj Bhargava, Girja Sharma, Ritu Bhargava, and Manish Mathuria. Decision tree analysis on j48 algorithm for data mining. Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering, 3(6), 2013.

[43] Chris Fleizach and Satoru Fukushima. A naive bayes classifier on 1998 kdd cup, 1998.

[44] Mouhammd Alkasassbeh, Ghazi Al-Naymat, Ahmad BA Hassanat, and Mohammad Almseidin. Detecting distributed denial of service attacks using data mining techniques. International Journal of Advanced Computer Science & Applications, 1(7):436–445.

[45] Safaa O Al-mamory and Firas S Jassim. Evaluation of different data mining algorithms with kdd cup 99 data set. Journal of Babylon University/Pure and Applied Sciences, 21(8):2663–2681, 2013.

[46] Stephen D Bay. The uci kdd archive [http://kdd. ics. uci. edu]. irvine, ca: University of california. Department of Information and Computer Science, 404:405, 1999.

[47] Mouhammd Al-Kasassbeh. Network intrusion detection with wiener filter-based agent. World Appl. Sci. J, 13(11):2372–2384, 2011.

[48] J Ross Quinlan. C4. 5: programs for machine learning. Elsevier, 2014.

[49] Manpreet Singh Bhullar and Amritpal Kaur. Use of data mining in education sector. In Proceedings of the World Congress on Engineering and Computer Science, volume 1, pages 24–26, 2012.

[50] Ron Kohavi and Dan Sommerfield. Targeting business users with decision table classifiers. In KDD, pages 249–253, 1998.

[51] Purohit Aditi and Gupta Hitesh. A new approach of intrusion detection system using clustering, classification and decision table. 2013.

[52] Sankar K Pal and Sushmita Mitra. Multilayer perceptron, fuzzy sets, and classification. IEEE Transactions on neural networks, 3(5):683–697, 1992.

[53] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. Machine learning, 29(2-3):131–163, 1997.

[54] Adele Cutler and Guohua Zhao. Pert-perfect random tree ensembles. Computing Science and Statistics, 33:490–497, 2001.

[55] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. ACM SIGKDD explorations newsletter, 11(1):10–18, 2009.

[56] Ibrahim M Obeidat, Nabhan Hamadneh, Mouhammd Alkasassbeh, Mohammad Almseidin, and Mazen Ibrahim AlZubi. Intensive pre-processing of kdd cup 99 for network intrusion classification using machine learning techniques. International Journal of Interactive Mobile Technologies, 13(1), 2019.

[57] Mohammad Almseidin, Maen Alzubi, Szilveszter Kovacs, and Mouhammd Alkasassbeh. Evaluation of machine learning algorithms for intrusion detection system. In Intelligent Systems and Informatics (SISY), 2017 IEEE 15th International Symposium on, pages 000277–000282. IEEE, 2017.

[58] Mouhammad Alkasassbeh and Mohammad Almseidin. Machine learning methods for network intrusion detection. 12(8):2840, 2018.

[59] Rufai Kazeem Idowu, Zulaiha Ali Othman, et al. Denial of service attack detection using trapezoidal fuzzy reasoning spiking neural p system. Journal of Theoretical & Applied Information Technology, 75(3), 2015.

[60] Stephen D Bay, Dennis Kibler, Michael J Pazzani, and Padhraic Smyth. The uci kdd archive of large data sets for data mining research and experimentation. ACM SIGKDD explorations newsletter, 2(2):81–85, 2000.

[61] Shailesh P Thakare and MS Ali. Introducing fuzzy logic in network intrusion detection system. International Journal of Advanced Research in Computer Science, 3(3), 2012.

[62] Nenekazi Nokuthala Penelope Mkuzangwe and Fulufhelo Vincent Nelwamondo. A fuzzy logic based network intrusion detection system for predicting the tcp syn flooding attack. In Asian conference on intelligent information and database systems, pages 14–22. Springer, 2017.

[63] Mehdi Naseriparsa, Amir-Masoud Bidgoli, and Touraj Varaee. A hybrid feature selection method to improve performance of a group of classification algorithms. arXiv preprint arXiv:1403.2372, 2014.

[64] Yogita Danane and Thaksen Parvat. Intrusion detection system using fuzzy genetic algorithm. In 2015 International Conference on Pervasive Computing (ICPC), pages 1–5. IEEE, 2015.

[65] Ali Feizollah, Shahaboddin Shamshirband, Nor Badrul Anuar, Rosli Salleh, and Miss Laiha Mat Kiah. Anomaly detection using cooperative fuzzy logic controller. In FIRA RoboWorld Congress, pages 220–231. Springer, 2013.

[66] Amin Einipour. Intelligent intrusion detection in computer networks using fuzzy systems. Global Journal of Computer Science and Technology, 2012.

[67] Alireza Shameli-Sendi, Rouzbeh Aghababaei-Barzegar, and Mohamed Cheriet. Taxonomy of information security risk assessment (isra). Computers & security, 57:14–30, 2016.

[68] Andrea Saracino, Daniele Sgandurra, Gianluca Dini, and Fabio Martinelli. Madam: Effective and efficient behavior-based android malware detection and prevention. IEEE Transactions on Dependable and Secure Computing, 15(1):83–97, 2018.

[69] Irfan Sofi, Amit Mahajan, and Vibhakar Mansotra. Machine learning techniques used for the detection and analysis of modern types of ddos attacks. learning, 4(6), 2017.

[70] Martin Roesch et al. Snort: Lightweight intrusion detection for networks. In Lisa, volume 99, pages 229–238, 1999.

[71] Fangyi Li, Ying Li, Changjing Shang, and Qiang Shen. Improving fuzzy rule interpolation performance with information gain-guided antecedent weighting. Soft Computing, 22(10):3125–3139, 2018.

[72] Szilveszter Kovács. New aspects of interpolative reasoning. In Proceedings of the 6th. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain, pages 477–482, 1996.

[73] Szilveszter Kovács and László T Kóczy. The use of the concept of vague environment in approximate fuzzy reasoning. Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, Bratislava, Slovak Republic, 12:169–181, 1997.

[74] Szilveszter Kovacs and Laszlo T Koczy. Approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rulebase. In Intelligent Engineering Systems, 1997. INES'97. Proceedings., 1997 IEEE International Conference on, pages 63–68. IEEE, 1997.

[75] Zs Cs Johanyák. Sparse fuzzy model identification matlab toolox-rulemaker toolbox. In Computational Cybernetics, 2008. ICCC 2008. IEEE International Conference on, pages 69–74. IEEE, 2008.

[76] Zsolt Csaba Johanyák and Szilveszter Kovács. Sparse fuzzy system generation by rule base extension. In Intelligent Engineering Systems, 2007. INES 2007. 11th International Conference on, pages 99–104. IEEE, 2007.

[77] Zsolt Csaba Johanyák and Szilveszter Kovács. Polar-cut based fuzzy model for petrophysical properties prediction. Scientific Bulletin of Politehnica University of Timisoara, Romania, Transactions on Automatic Control and Computer Science, 57(67):195–200, 2008.

[78] Zsolt Csaba Johanyák, Domonkos Tikk, Szilveszter Kovács, and Kok Wai Wong. Fuzzy rule interpolation matlab toolbox-fri toolbox. 2006.

[79] Jaehak Yu, Hansung Lee, Myung-Sup Kim, and Daihee Park. Traffic flooding attack detection with snmp mib using svm. Computer Communications, 31(17):4212–4219, 2008.

[80] L. Garber. Denial-of-service attacks rip the internet. Computer, 33(4):12–17, April 2000.

[81] Mouhammd Al-Kasassbeh and Mo Adda. Network fault detection with wiener filter-based agent. Journal of Network and Computer Applications, 32(4):824–833, 2009.

[82] Joao BD Cabrera, Lundy Lewis, Xinzhou Qin, Wenke Lee, and Raman K Mehra. Proactive intrusion detection and distributed denial of service attacksâ€"a case study in security management. Journal of Network and Systems Management, 10(2):225–254, 2002.

[83] Han-Wei Hsiao, Cathy S Lin, and Ssu-Yang Chang. Constructing an arp attack detection system with snmp traffic data mining. In Proceedings of the 11th international conference on electronic commerce, pages 341–345. ACM, 2009.

[84] Walter Cerroni, Gianluca Moro, Roberto Pasolini, and Marco Ramilli. Decentralized detection of network attacks through p2p data clustering of snmp data. Computers & Security, 52:1–16, 2015.

[85] Walter Cerroni, Gianluca Moro, Tommaso Pirini, and Marco Ramilli. Peer-to-peer data mining classifiers for decentralized detection of network attacks. In Proceedings of the Twenty-Fourth Australasian Database Conference-Volume 137, pages 101–107. Australian Computer Society, Inc., 2013.

[86] Sahar Namvarasl and Marzieh Ahmadzadeh. A dynamic flooding attack detection system based on different

classification techniques and using snmp mib data. International Journal of Computer Networks and Communications Security, 2(9):279–284, 2014.

[87] Mouhammd Al-Kasassbeh, Ghazi Al-Naymat, and Eshraq Al-Hawari. Towards generating realistic snmp-mib dataset for network anomaly detection. International Journal of Computer Science and Information Security, 14(9):1162, 2016.

[88] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60:19–31, 2016.

[89] Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. ACM SIGCOMM Computer Communication Review, 34(2):39–53, 2004.

[90] Mangesh Salunke, Ruhi Kabra, and Ashish Kumar. Layered architecture for dos attack detection system by combined approach of naive bayes and improved k-means clustering algorithm. International Research Journal of Engineering And Technology, 2(3):372–377, 2015.

[91] Saman Taghavi Zargar, James Joshi, and David Tipper. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. IEEE communications surveys & tutorials, 15(4):2046–2069, 2013.

[92] Frank Klawonn. Fuzzy sets and vague environments. Fuzzy Sets and Systems, 66(2):207–221, 1994.

[93] Mouhammd Alkasassbeh. An empirical evaluation for the intrusion detection features based on machine learning and feature selection methods. Journal of Theoretical and Applied Information Technology, 95(22), 2017.

[94] Kaspersky. The cost of ddos attacks. , Kaspersky, B2B International, 2017.

[95] Samaneh Rastegari, M Iqbal Saripan, and Mohd Fadlee A Rasid. Detection of denial of service attacks against domain name system using neural networks. arXiv preprint arXiv:0912.1815, 2009.

[96] Xuejiao Liu, Debao Xiao, Ting Gu, Hui Xu, et al. Scenario recognition based on collaborative attack modeling in intrusion detection. In Proceedings of the International MultiConference of Engineers and Computer Scientists, volume 1, 2008.

[97] Guy Helmer, Johnny Wong, Mark Slagell, Vasant Honavar, Les Miller, and Robyn Lutz. A software fault tree approach to requirements analysis of an intrusion detection system. Requirements Engineering, 7(4):207–220, 2002.

[98] Yanxin Wang, Smruti Ranjan Behera, Johnny Wong, Guy Helmer, Vasant Honavar, Les Miller, Robyn Lutz, and Mark Slagell. Towards the automatic generation of mobile agents for distributed intrusion detection system. Journal of Systems and Software, 79(1):1–14, 2006.

[99] Frédéric Cuppens, Fabien Autrel, Alexandre Miege, Salem Benferhat, et al. Recognizing malicious intention in an intrusion detection process. In HIS, pages 806–817, 2002.

[100] Ashvin Alagiya, Hiren Joshi, and Ashish Jani. Performance analysis and enhancement of utm device in local area network. International Journal of Modern Education and Computer Science, 5(10):43, 2013.

[101] Do-hyeon Lee, Doo-young Kim, and Jae-il Jung. Multi-stage intrusion detection system using hidden markov model algorithm. In Information Science and Security, 2008. ICISS. International Conference on, pages 72–77. IEEE, 2008.

[102] Dirk Ourston, Sara Matzner, William Stump, and Bryan Hopkins. Applications of hidden markov models to detecting multi-stage network attacks. In System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on, pages 10–pp. IEEE, 2003.

[103] Joel Branch, Alan Bivens, Chi Yu Chan, Taek Kyeun Lee, and Boleslaw K Szymanski. Denial of service intrusion detection using time dependent deterministic finite automata. In Proc. Graduate Research Conference, pages 45–51, 2002.

[104] Juan J Flores, Anastacio Antolino, Juan M Garcia, and Felix Calderon Solorio. Hybrid network anomaly detection–learning hmms through evolutionary computation. iConcept Press Ltd., 2012.

[105] Mrs Manisha Bharati and Santosh Lomte. A survey on hidden markov model (hmm) based intention prediction techniques. International Journal of Engineering Research and Applications, 6(1):167–172, 2016.

[106] Shrijit S Joshi and Vir V Phoha. Investigating hidden markov models capabilities in anomaly detection. In Proceedings of the 43rd annual Southeast regional conference-Volume 1, pages 98–103. ACM, 2005.

[107] DARPA Datasets. Mit lincoln laboratory, darpa intrusion detection evaluation data sets, 2000.

[108] Nagaraju Devarakonda, Srinivasulu Pamidi, V Valli Kumari, and A Govardhan. Integrated bayes network and hidden markov model for host based ids. International Journal of Computer Applications, 41(20), 2012.

[109] AS Aneetha and S Bose. Probabilistic approach for intrusion detection system-fomc technique. In Advanced Computing (ICoAC), 2014 Sixth International Conference on, pages 178–183. IEEE, 2014.

[110] Nagaraju Devarakonda, Srinivasulu Pamidi, V Valli Kumari, and A Govardhan. Intrusion detection system using bayesian network and hidden markov model. Procedia Technology, 4:506–514, 2012.

[111] Uttam Adhikari, Thomas H Morris, and Shengyi Pan. Applying non-nested generalized exemplars classification for cyber-power event and intrusion detection. IEEE Transactions on Smart Grid, 2016.

[112] IBM. Behaviour modeling with state machine and activity diagrams. KTH Royal Institute of Technology in Stockholm, 2008.

[113] Mor Vered. Finite state machines. Benson Idahosa University, 2008.

[114] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. Automata theory, languages, and computation. International Edition, 24, 2006.

[115] R Sekar, Mugdha Bendre, Dinakar Dhurjati, and Pradeep Bollineni. A fast automaton-based method for detecting anomalous program behaviors. In sp, page 0144. IEEE, 2001.

[116] R&D Defence. A finite state machine algorithm for detecting TCP anomalies: An examination of the 1999 DARPA intrusion detection evaluation data set. Defence R&D Canada-Ottawa, 2005.

[117] Zong-Fen Han, Jian-Ping Zou, Hai Jin, Yan-Ping Yang, and Jian-Hua Sun. Intrusion detection using adaptive time-dependent finite automata. In Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on, volume 5, pages 3040–3045. IEEE, 2004.

[118] Giovanni Vigna and Richard A Kemmerer. Netstat: A network-based intrusion detection approach. In Computer Security Applications Conference, 1998. Proceedings. 14th Annual, pages 25–34. IEEE, 1998.

[119] A Gemona, I Duncan, and A Miller. Nemesi: Using a tcp finite state machine against tcp syn flooding attacks. University of St Andrews, 2006.

[120] Rahul Kumar Singh and Ajay Guide Kumar. Conversion of Fuzzy Regular Expressions to Fuzzy Automata using the Follow Automata. PhD thesis, Thapar University, 2014.

[121] Mansoor Doostfatemeh and Stefan C Kremer. New directions in fuzzy automata. International Journal of Approximate Reasoning, 38(2):175–214, 2005.

[122] Szilveszter Kovács, Dávid Vincze, Márta Gácsi, Ádám Miklósi, and Péter Korondi. Ethologically inspired robot behavior implementation. In Human System Interactions (HSI), 2011 4th International Conference on, pages 64–69. IEEE, 2011.

[123] Alireza Shameli Sendi, Michel Dagenais, Masoume Jabbarifar, and Mario Couture. Real time intrusion prediction based on optimized alerts with hidden markov model. JNW, 7(2):311–321, 2012.

[124] André Årnes, Fredrik Valeur, Giovanni Vigna, and Richard A Kemmerer. Using hidden markov models to evaluate the risks of intrusions. In International Workshop on Recent Advances in Intrusion Detection, pages 145–164. Springer, 2006.

[125] Opeyemi Osanaiye, Haibin Cai, Kim-Kwang Raymond Choo, Ali Dehghantanha, Zheng Xu, and Mqhele Dlodlo. Ensemble-based multi-filter feature selection method for ddos detection in cloud computing. EURASIP Journal on Wireless Communications and Networking, 2016(1):130, 2016.

[126] Imre Piller, Dávid Vincze, and Szilveszter Kovács. Declarative language for behaviour description. In Emergent Trends in Robotics and Intelligent Systems, pages 103–112. Springer, 2015.

[127] Imre Piller. The simulation environment. http://mat76.mat.uni-miskolc.hu/ imre/fbdl/simulator.html.

[128] Aleksandar Lazarevic, Vipin Kumar, and Jaideep Srivastava. Intrusion detection: A survey. In Managing Cyber Threats, pages 19–78. Springer, 2005.

[129] Mohammad Almseidin, Mouhammd Al-kasassbeh and Szilveszter Kovacs. Fuzzy Rule Interpolation and SNMP-MIB for Emerging Network Abnormality. International Journal on Advanced Science, Engineering and Information Technology, vol. 9, no. 2, pp. 1-7, 2019.

[130] Mohammad Almseidin, Imre Piller , Mouhammd Al-kasassbeh and Szilveszter Kovacs. Fuzzy Automaton as a Detection Mechanism for the Multi-Step Attack. International Journal on Advanced Science, Engineering and Information Technology, vol. 9, no. 2, pp. 1-7, 2019.

[131] Mouhammad Alkasassbeh and Mohammad Almseidin. Machine learning methods for network intrusion detection. International Journal of Computer and Information Engineering 12 : 8 p. 1 (2018).

[132] Breiman, Leo, Friedman, J. H, Olshen, R. A, Stone and C. J, Classification and regression trees. Wadsworth, Belmont, CA, 1984

[133] L. Fausett and L. Fausett, Fundamentals of neural networks: architectures, algorithms, and applications, Prentice-Hall, 1994.

[134] J. Jorgenson, C. a. Manikopoulos, J. Li and Z. Zhang, "A hierarchical Anomaly Network Intrusion Detection System Using Neural Network Classification," in Workshop on Information, 2001.

[135] K. Hornik, M. Stinchcombe and H. White. Multilayer feedforward networks are universal approximators. Neural Networks, 2, 359-366, 1989.

[136] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, p. 20, Jul 2019.

[137] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy and survey of intrusion detection system design techniques, network threats and datasets," arXiv preprint arXiv:1806.03517, 2018.

[138] Ye N, Emran SM, Chen Q, Vilbert S (2002) Multivariate statistical analysis of audit trails for host-based intrusion detection. IEEE Trans Comput 51(7):810–820

[139] J. Viinikka, H. Debar, L. Mé, A. Lehikoinen, and M. Tarvainen, "Processing intrusion detection alert aggregates with time series modeling," Information Fusion, vol. 10, no. 4, pp. 312–324, 2009/10/01/ 2009

[140] N. Walkinshaw, R. Taylor, and J. Derrick, "Inferring extended finite state machine models from software executions," Empirical Software Engineering, journal article vol. 21, no. 3, pp. 811–853, June 01 2016

[141] Studnia I, Alata E, Nicomette V, Kaâniche M, Laarouchi Y (2018) A language-based intrusion detection approach for automotive embedded networks. Int JEmbed Syst 10(1):1–12

[142] Kenkre PS, Pai A, Colaco L Real Time Intrusion Detection and Prevention System. Springer International Publishing, Cham, pp 405–411 Khraisat A, Gondal I, Vamplew P (2018) An anomaly intrusion detection system using C5 decision tree classifier. In: Trends and applications in knowledge discovery and data mining. Springer International Publishing, Cham, pp 149–155

[143] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning," arXiv preprint arXiv:1801.06275, 2018

[144] Rutkowski L, Jaworski M, Pietruczuk L, Duda P (2014) Decision trees for mining data streams based on the Gaussian approximation. IEEE Trans Knowl Data Eng 26(1):108–119

[145] S. N. Murray, B. P. Walsh, D. Kelliher, and D. T. J. O'Sullivan, "Multi-variable optimization of thermal energy efficiency retrofitting of buildings using static modelling and genetic algorithms – a case study," Build Environ, vol. 75, no. Supplement C, pp. 98–107, 2014/05/01/ 2014

[146] S. Elhag, A. Fernández, A. Bawakid, S. Alshomrani, and F. Herrera, "On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems," Expert Syst Appl, vol. 42, no. 1, pp. 193–202, 1// 2015

[147] C. Annachhatre, T. H. Austin, and M. Stamp, "Hidden Markov models for malware classification," Journal of Computer Virology and Hacking Techniques, vol. 11, no. 2, pp. 59–73, 2015/05/01 2015

[148] Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," Expert Syst Appl, vol. 39, no. 1, pp. 424–430, 2012.

[149] A. H. Sung and S. Mukkamala, "Identifying important features for intrusion detection using support vector

machines and neural networks," in Symposium on Applications and the Internet, 2003, pp. 209–216
[150] Loukas, G., & Oke, G. Protection Against Denial of Service Attacks: A survey. The Computer Journal, 2009.
[151] Lu, K., Wu, D., Fan, J., Todorovic, S., & Nucci, A. Robust And Efficient Detection of DDoS Attacks For Large-Scale Internet. Computer Networks, 51, 5036-5056, 2007.
[152] Nitin Naik, Ren Diao, and Qiang Shen, Dynamic Fuzzy Rule Interpolation and Its Application to Intrusion Detection, IEEE Transactions On Fuzzy Systems, Vol. 26, No. 4, August 2018
[153] Longzhi Yang ; Jie Li ; Gerhard Fehringer, Intrusion detection system by fuzzy interpolation, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE),2017
[154] Naik, Nitin, Dynamic Fuzzy Rule Interpolation, Aberystwyth University, 2015
[155] Allison, David B and Paultre, General AUC calculated based on the trapezoidal rule, PharmaSug, 1995.