International Journal on
Advanced Science
Engineering
Information Technology

Vol.9 (2019) No. 2
ISSN: 2088-5334

Fuzzy Automaton as a Detection Mechanism for the Multi-Step Attack

Mohammad Almseidin[#1], Imre Piller[#2], Mouhammd Al-Kasassbeh[*], Szilveszter Kovacs[#3]

[#] *Department of Information Technology, University of Miskolc, H-3515 Miskolc, Hungary*
E-mail: [1]alsaudi@iit.uni-miskolc.hu; [2]piller@iit.uni-miskolc.hu; [3]szkovacs@iit.uni-miskolc.hu

[*] *Computer Science Department, Princess Sumaya University for Technology, Amman, Jordan*
E-mail: m.alkasassbeh@psut.edu.jo

***Abstract*—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 Intrusion Detection Systems (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. The goal of this paper is to introduce a novel detection mechanism for multi-step attacks built upon Fuzzy Rule Interpolation (FRI) based fuzzy automaton. In that respect, the FRI method instruments the fuzzy automaton to be able to act on a not fully defined state transition rule-base, by offering interpolated conclusion even for situations which are not explicitly defined. In the suggested model, the intrusion definition state transition rule-base is defined using an open source fuzzy declarative language. On the multi-step attack benchmark dataset introduced in this paper, the proposed detection mechanism was able to achieve 97.836% detection rate. Furthermore, in the studied examples, the suggested method was 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. According to these results, the IDS built upon the FRI based fuzzy automaton could be a useful device for detecting multi-step attacks, even in cases when the intrusion state transition rule-based is incomplete. The early detection of multi-step attacks also allows the administrator to take the necessary actions in time, to mitigate the potential threats.**

***Keywords*— Intrusion Detection System (IDS); fuzzy automaton; Fuzzy Rule Interpolation (FRI); multi-step attack.**

I. INTRODUCTION

Nowadays, network administrators face stressful environments with an overload of network traffics. These traffics need to be analyzed and investigated to detect abnormalities. The IDS has benefitted 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 challenging to combat.

Computer and network security systems face different types of sophisticated attacks. One sophisticated kind of 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 execution of the final step of the attack. As detailed in the security report of 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 [3] revealed that 91% of enterprise businesses are affected by these types of sophisticated attacks, the largest proportion of which are Denial of Service (DoS) 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 [4].

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 [5]. Often, there is a causal relationship between the attack steps and forecasting the next step of attack [6]. There is an increasing need to design and implement an efficient IDS detection mechanism capable of handling different attack scenarios. The IDS systems are categorized based either on their monitoring techniques or their detection methodologies [7], [8]. In terms of monitoring techniques, the IDS system can be categorized as a Network Intrusion Detection System (NIDS) which operates to monitor and detect intrusions within all network devices. Also, IDS comprises a Host Intrusion Detection System (HIDS) which works to protect a specific device within the network.

From another perspective, the IDS systems could also be categorized based on their detection methodologies either as anomaly-based IDSs or signature-based IDSs [9]. The anomaly-based IDS detects intrusions based on the normal historical behavior of a specific network. It compares the real-time network traffics with the normal historical behavior to detect for abnormalities. The signature-based IDS verifies the current packet pattern (sequence series of packets) by comparing it with the pre-defined, stored intrusions patterns.

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) [10]. 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 [6]. Therefore, the multi-step attacks consist of different preliminary phases which 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 [11]. 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 [10].

This work proposes a novel detection mechanism for the multi-step attack. The proposed detection mechanism was able to detect the multi-step attack even within the early stages of the attack. Furthermore, it could extend the binary decision to continuous space. The proposed detection mechanism was performed using the fuzzy automaton. The fuzzy automaton derives its strength from two paradigms: the theory of automata and the fuzzy system. The reasoning part of the proposed detection mechanism adopts the FRI method instead of classical reasoning methods. This is done 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.

The rest of the paper is organized as follows: section (II) describes the different types of multi-step attacks and classifies them, in detail, based on their implementation scenarios. It also explains how the intrusion detection mechanisms (finite state machine and hidden Markov models) can be used to detect for multi-step attacks. These studies are then evaluated and followed by a discussion section to present some recent gaps in these methods. Subsection (II-H) details the design and structure of the proposed FRI based fuzzy automaton model. The results and discussion presented in section (III). Finally, section (IV) concludes the paper.

II. Material And Method

This section 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.

A. Denial of Service (DoS) Mstream

DoS attacks are considered one of the most harmful types of attack. They directly affect the confidentiality, integrity, and availability of network services. This attack aims to prevent several network and computer services [10]. 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 [8]. 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 [12] 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 can generate a list of intended victims. The attacker collects the victims to implement the root access login using Remote SHell (RSH) access script. This step aims 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.

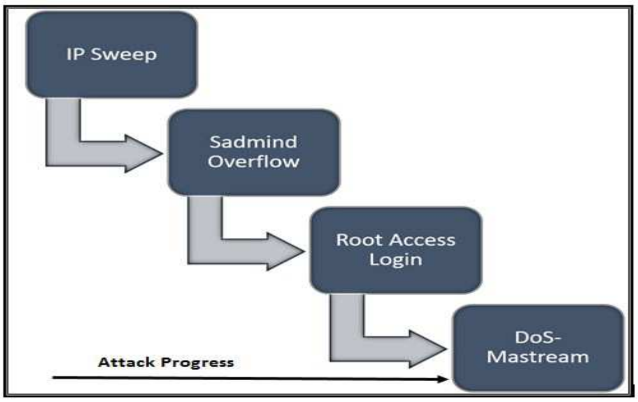

Fig. 1 The Sequence Events of The DoS-Mstream Attack

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.1 presents the event sequence of the DoS-Mstream multi-step attack.

B. File Transfer Protocol (FTP) Bounce

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 [13]. 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 [14].

According to [13], [14], 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.2 presents the sequence of events for the FTP bounce multi-step attack.

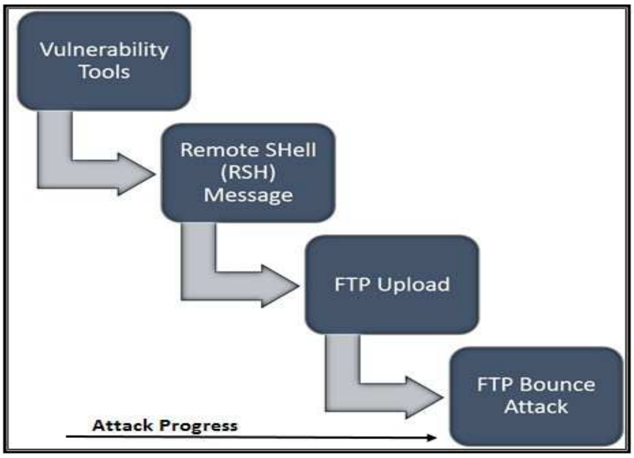

Fig. 2 The Sequence of Events of the FTP-Bounce Attack

C. DoS on Domain Name Server (DNS)

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 [4]. 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 [15] 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 [15] or HyenaeFE [16].

Fig.3 presents the sequence of events of the DoS-DNS multi-step attack.

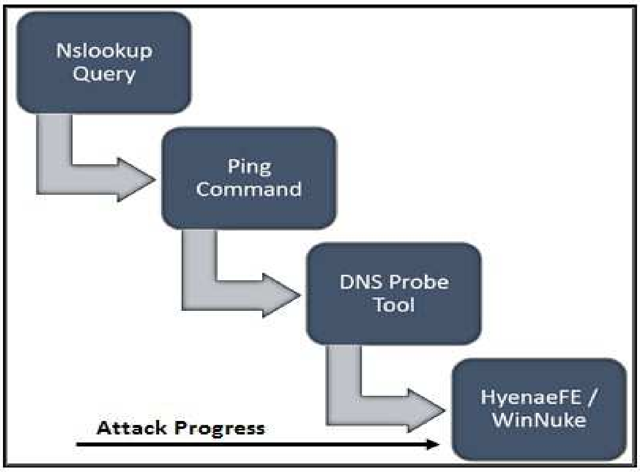

Fig.3 The Sequence of Events of the DOS-DNS Attack

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 [17]. 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 the multi-step attacks [12], [18], [19].

D. Hidden Markov Models (HMMs) Against Multi-Step Attacks

HMMs [20] 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 [21]:

- 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 [22], 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 [23]. 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 HMM solutions too. Devarakonda et al., in [24], propose a model to detect multi-step attacks based on HMMs and Bayesian network. 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) [25]. The second phase was performed based on the extracted states' 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 [26], 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 detected 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 [27] 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 Bayesian network algorithm generated the state transition tables for both the normal and multi-step attacks. 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 [22] and [26]-[28], the system states could be determined using various methods including the Bayesian network, clustering algorithms, and Non-Nested Generalized Exemplars (NNGE).

E. Deterministic Finite State Machine (DFSM) Against MultiStep Attacks

At present, research has been conducted to implement the DFSM as a mechanism for detecting multi-step attacks. The DFSM [29] 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 the DFSM against multi-step attacks is to detect the multi-step attacks in the levels of the stage (before they posed a harmful step). The DFSM had the following properties [29], [30]:

- 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 [31] can be described by a 5-tuple expressed in Equation 1.

$$M = (Q, \Sigma, \delta, q_0, F) \qquad (1)$$

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., in [19], 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 detect the DoS multi-step attack within the DARPA dataset successfully.

The work of Sekar et al., in [32], 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 which 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 adapted to the normal behavior pattern to detect the multi-step attacks.

Treurniet et al., in [33], study the simulated network profile's normal behavior 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 [34], 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 are analyzed to define the normal and abnormal patterns (training phase). In the second phase, the Hsiaos sequential approach is applied to determine the causal relationship between the series of packets. The Hsiaos sequential approach was used to define the sequence series of packages 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., in [19], continue the work of Vigna et al., in [35], 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. This extension aims 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 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., in [36], 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.

F. Discussions

The works mentioned above provide convincing contributions and show support for the persistent need to detect and predict multi-step attacks at their onset before they pose a serious harm. However, the previous works 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 [10], [11]. 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 adapt 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 [29], [30]. Therefore, the detection mechanism could only follow a single path of event change state.

In response to the previous issues, this paper introduces a novel mechanism for detecting multi-step attacks by the application of the FRI based fuzzy automaton. 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 smooth 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 the 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 history data.

G. Fuzzy Automaton Detection Mechanism

This subsection presents the full architecture of the fuzzy automaton detection mechanism and discusses its main functions and interactions.

Automata theory [37] 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 automat, 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 [10]. The general definition of the fuzzy automaton [38] is presented as a 6-tuple, illustrated in Equation 2.

$$F = (Q, \Sigma, \delta, R, Z, \omega) \qquad (2)$$

Where Q is the finite set of the system states, $Q=\{q_0,q_1,...,q_k\}$. Σ is the finite set of the input symbols, $\Sigma = \{x_0,x_1,...,x_n\}$. 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, $\delta\colon Q \times \Sigma \times Q \rightarrow (0,1]$. R shows the initial system state F, Re ∈ Q. Z presents the finite set of output, $Z = \{Z_0, Z_1,..., Z_k\}$. Finally, ω presents the output mapping function which is responsible for mapping the fuzzy states into the output set, $\omega\colon Q \times \Sigma \rightarrow Z$.

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. Also, the transitions between different states occurred based on the predefined fuzzy rules. In [39], the general definition of the fuzzy automaton was extended as shown in Equation 3.

$$F = (S, X, \delta, P, Y, \omega) \qquad (3)$$

Where S is the finite set of fuzzy system states, $S = \{m_{s1},m_{s2},...,m_{sk}\}$, x is the finite set of dimensional input values, $x = \{x_0,x_1,...,x_n\}$. 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, $\delta : S \times X \rightarrow S$. P shows the initial fuzzy state of F, $P \in S$. Y is the finite set of output dimensional vectors, $Y = \{Y_0, Y_1,..., Y_k\}$. Finally, ω represents the output mapping function which is responsible for mapping the fuzzy states based on input values to the output set, $\omega\colon S\times X \rightarrow Y$.

H. Fuzzy Automaton Detection Mechanism Architecture

The fuzzy automaton detection mechanism 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 mechanism are presented as a set of system observations (i1, i2, in).
- 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, δ: S × X → S.
- Defining the finite set of system outputs, Y =
- {Y0,Y1,...,Yk}.
- Defining the output mapping function (ω: S×X → Y) which is responsible for mapping the fuzzy states based on input values to the output.

The fuzzy automaton detection mechanism adapts the Fuzzy Interpolation based on the Vague Environment FRI method (FIVE) which was introduced by Kovacs in [40]-[42]. 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 [43]. Fig.4 shows the general architecture of the fuzzy automaton detection mechanism using the previous six major components.

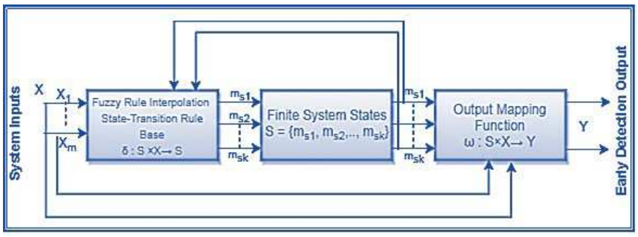

Fig. 4 The Fuzzy Automaton Detection Mechanism Architecture

The suggested fuzzy automaton based detection mechanism consisted of four system states S = {N, A, P, C}. These states are similar to those used in [44] 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.5 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.

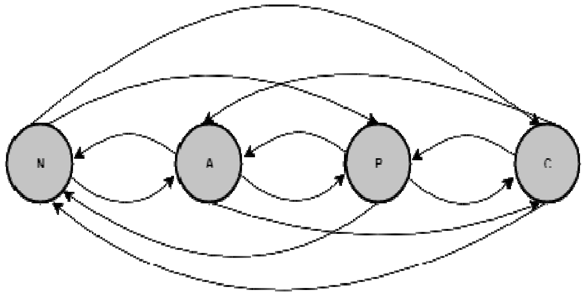

Fig. 5 System States of the Fuzzy Automaton Detection Mechanism

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.6 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.

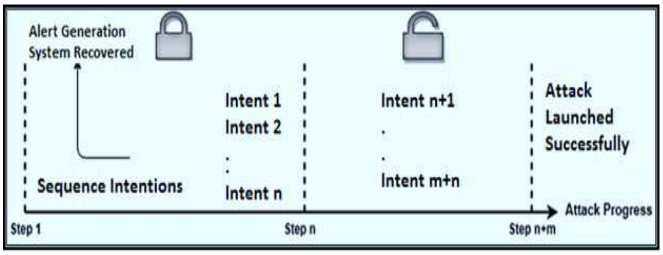

Fig. 6 The Concentration Intents of Fuzzy Automaton Detection Mechanism

I. The Validation Methodology for The Fuzzy Automaton Detection Mechanism

In this subsection, the fuzzy automaton detection mechanism's validation methodology is presented and discussed. The DARPA 2000 attack scenarios dataset LLDOS1.0 (inside) was used [23] to evaluate the fuzzy automaton detection mechanism in practice. 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 DARPA 2000 dataset is that it contains a detailed truth table which allows for the obtained results to be checked. Moreover, most of the IDS detection approaches have applied this dataset for testing and evaluating processes [19]. 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 [45]. 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 I illustrates the five sequence steps of the first DARPA attack scenario.

TABLE I
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 TELNET login.

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

TABLE II
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 (inside) was reformulated by extracting the values of the main feature and labeling the data according to the existing literature results [44], [45]. 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 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 were selected as an input parameter. The relevant features were selected based on the intersection operation between the Gain Ratio (GR) algorithm, the Information Gain (IG) algorithm and the ReliefF (RF) algorithm [46]. Those features that fulfill the intersection criteria, as shown in Equation 4, were selected as the proposed detection mechanism’s input parameters. Table III shows the relevant input parameters for the fuzzy automaton detection mechanism.

$$GR \cap IG \cap RF \tag{4}$$

TABLE III
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.

J. The State-transition Rules

The state-transition rule-base, which is extracted from the expert heuristic, is necessary to implement the fuzzy automaton detection mechanism. One efficient tool for shaping the expert heuristic to fuzzy rules is the fuzzy declarative language [47]. It provides 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.

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 mechanisms have 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 application of the FRI methods is beneficial in the IDS application area [10]. 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-based). 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 "Mss_Request"
   "VSmall"   1 1987
   "Small"    2200 2700
   "Medium"  3200 5350
   "Large"    6500 8000
End

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
```

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

```
Rulebase "Attempt_State"
Rule
"High" when
 "Mss_Requested" is "Medium" and "Pure_A2B" is "Medium"
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
Rule
"Low" when
                   "Mss_Requested" is "Medium"
 end
 End
```

The prerequisite state has four rule definitions as follows:

```
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
```

The compromised state has five rule definitions as follows:

```
Rulebase "Compromise_State"
Rule
"High" when
      "Mss_Requested" is " VLarge " and "Pure_A2B" is "Medium"
end
Rule
"High"  when
         "Mss_Requested" is  "Large" and "TotalA2B" is "Large"
end
Rule
"High" when
        "Mss_Requested" is "VLarge" and "Pure_A2B" is "Large"
 end

Rule
"Low" when
                        "Mss_Requested" is "VSmall"
end
Rule
"Low" when
                         "Mss_Requested" is "Small"
 end
 End
```

III. Results and Discussion

According to the way, as the FRI (FIVE) method calculates the conclusion, the evaluation process of rule bases can be described in 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. The simulation environment can be accessed through [48].

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 IV 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 IV
THE OUTPUT RESPONSE OF THE PROPOSED MECHANISM

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 [19], where the ROC shows the trade-off between sensitivity and specificity [49]. In keeping with the standard measure of most other IDS detection mechanisms, Fig. 7 shows the evaluation performance, with the ROC curve, for the fuzzy automaton detection mechanism states.

Table V 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.

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 the 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.

TABLE V
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%**					

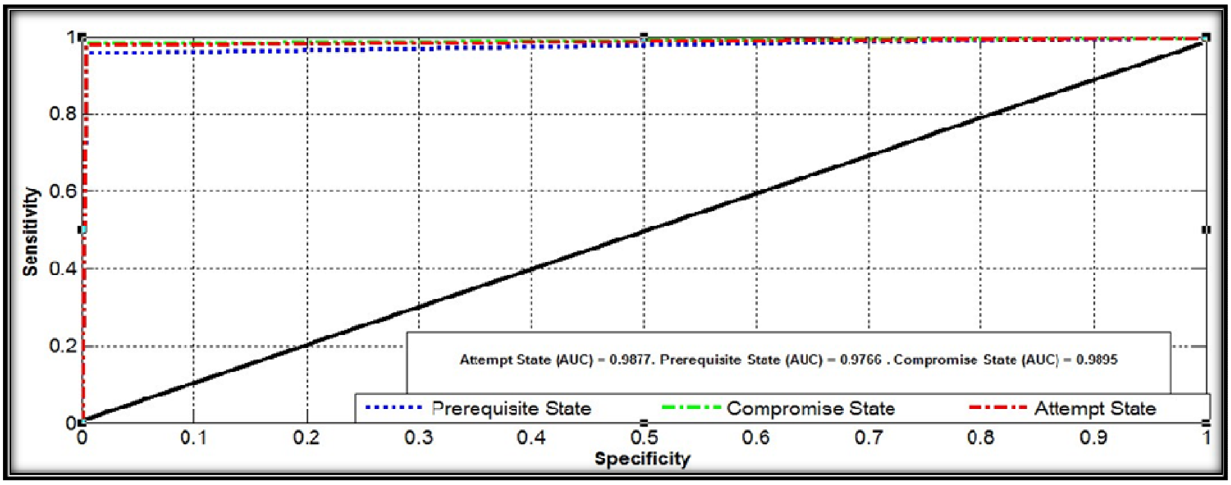

Fig. 7 The ROC Curve for The Fuzzy Automaton Detection States.

The main characteristics of the proposed FRI fuzzy automaton IDS and the other state machine detection mechanisms are compared on Table VI. 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. Also, 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.

TABLE VI
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

IV. CONCLUSIONS

This paper has introduced a novel method for detecting multi-step attacks by combining the Fuzzy Interpolation, based on the Vague Environment (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.

ACKNOWLEDGMENT

The described article was carried out as part of the EFOP-3.6.1-16-00011 “Younger and Renewing University – Innovative Knowledge City – institutional development of the University of Miskolc aiming at intelligent specialization” project implemented in the framework of the Szechenyi 2020 program. The European Union supports the realization of this project, co-financed by the European Social Fund.

REFERENCES

[1] C Yuan. Research on multi-step attack detection method based on GCT. *Jilin University, Jilin*, China, 2010.

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

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

[4] Samaneh Rastegari, M Iqbal Saripan, and Mohd Fadlee A Rasid. Detection of denial of service attacks against domain name system using neural networks. *In Proceedings of the World Congress on Engineering, Vol I WCE* 2010, June 30 - July 2, 2010, London, U.K.

[5] 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.

[6] Can Chen and BQ Yan. Network attack forecast algorithm for multistep attack. *Computer Engineering*, 5(37):172–174, 2011.

[7] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):11994–12000, 2009.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

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

[24] 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.

[25] 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. A*CM SIGKDD explorations newsletter*, 2(2):81–85, 2000.

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

[27] 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. [Online]. Available: https://doi.org/10.1016/j.protcy.2012.05.081.

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

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

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

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

[32] 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.

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

[34] 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.

[35] 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.

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

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

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

[39] Szilveszter Kovacs, David Vincze, Marta Gacsi, Adam Miklosi, and Peter Korondi. Ethologically inspired robot behavior implementation. *In Human System Interactions (HSI), 4th International Conference on, IEEE*, pages 64–69, 2011.

[40] Szilveszter Kovacs. 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.

[41] Szilveszter Kovacs and László 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.

[42] Szilveszter Kovacs and László Kóczy. Approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rule base. *In Intelligent Engineering Systems, Proceedings, IEEE International Conference on, pages 63–68. IEEE,* 1997.

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

[44] 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.

[45] Andre Arnes, 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. [Online]. Available: https://doi.org/10.1007/11856214_8

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

[47] Imre Piller, David Vincze, and Szilveszter Kovacs. Declarative language for behaviour description. *In Emergent Trends in Robotics and Intelligent Systems*, pages 103–112. Springer, 2015. [Online]. Available: https://doi.org/10.1007/978-3-319-10783-7_11.

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

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