Conceptual model for cyber intelligence network security system

Roumen Trifonov, Georgi Tsochev, Radoslav Yoshinov, Slavcho Manolov and Galya Pavlova

Abstract— With the increase in the number of devices that go to work, the need to use the Internet and access to more information resources is encouraged. This, in turn, also raises the number of accidents with network and information security. Cyber crimes and cyber threats are one of the most common in our day. This article presents a conceptual model of a penetration prevention and detection system. The methods of artificial intelligence that are applicable in this system are presented. They are divided into two main groups of traditional and adaptive optimization.

Keywords— adaptive optimization techniques, inteligent network, security, artificial intelligence, cyber security, algorithms.

I. Introduction

Too often, the unified security programs, based on comprehensive analyses of unified information from across the IT infrastructure, are costly, complex, difficult to implement and inefficient. As a result, most organizations lack accurate threat detection and informed risk-management capabilities. Therefore, the response to new information security threats can be a “security intelligence” approach with a reactive new policies or rules [1].

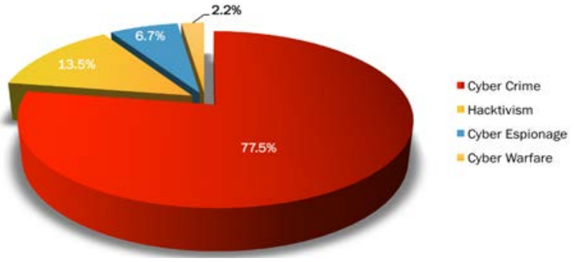

Fig. 1. Motivation the attack to be done

Owing to the distributed nature of modern attacks (e.g. denial-of-service), it is extremely challenging to detect such malicious behavior using traditional intrusion detection systems. Intelligent techniques play a role in automating the intrusion detection process and to reduce human intervention. The process of intelligent detection applies advanced communication protocols based on artificial intelligence (AI) techniques such as fuzzy set, neural networks, and evolutionary computing, that operate as classifiers for anomaly detection to ensure detection accuracy along with stability [2]. Denning (1987) used a rule-based expert system for Intrusion Detection Systems (IDSs) to improve detection performances. Although the rules may cover known patterns, they are unable to adapt in cases where attack patterns modify (e.g. attack polymorphs). In order to provide high accuracy detection in anomaly detection, computational intelligence (CI) can serve in the construction of a model detection system by automatically iterating training and testing data.

The Faculty of Computer Systems and Technology at Technical University of Sofia began research on the application of intelligent methods for increasing the security in computer networks. This article is the result of a research done by the project team and summarizes the results. The article shows the principles of several adaptive optimization techniques for intelligent network security - Fuzzy Logic, Genetic Algorithms, Q-Learning, Reinforcement Learning, Game Theory and one new approach adaptation of multi-agent based fuzzy reinforcement learning. They are the theoretical foundations on which the techniques proposed in our research for increasing the network security is based. Special attention is drawn to Reinforcement Learning and Fuzzy logic, which are the methods selected for our main and future research.

II. Intrusion Detection and Prevention Systems

Intrusion Detection System (abbreviated as IDS) is a security system that detects hostile activity on the network. The key is then to detect and possibly prevent actions that could jeopardize the security of the system, or attempt to break in the work, including the phases of exploration / collection of data that include, for example, a port scan. One of the key features of intrusion detection systems is their ability to provide a view of the unusual activity and issue alarms, notifies administrators and/or block the connection of the suspect.

Security devices that are NOT IDS:

This research is conducted and funded in relation to the execution of a contract № H07/56 “Increasing the level of network and information security using intelligent methods”.

R. Trifonov, Faculty of Computer Systems and Technology, Technical University of Sofia, Sofia, Bulgaria, (corresponding author, e-mail: r_trifonov@tu-sofia.bg)

G. Tsochev, Faculty of Computer Systems and Technology, Technical University of Sofia, Sofia, Bulgaria (e-mail: gtsochev@tu-sofia.bg)

R. Yoshinov, , Laboratory of Telematics, Bulgarian Academy of Sciences, Sofia, Bulgaria (e-mail: yoshinov@cc.bas.bg)

S. Manolov, Laboratory of Telematics, Bulgarian Academy of Sciences, Sofia, Bulgaria (e-mail: slav1943@gmail.com)

G. Pavlova, Faculty of Computer Systems and Technology, Technical University of Sofia, Sofia, Bulgaria (e-mail: raicheva@tu-sofia.bg)

- Logging systems used to detect any type of Denial of Service (DoS) attack across a network, are called network traffic monitoring systems.
- Vulnerability assessment tools that check for bugs and flaws in operating systems and network services (security scanners), for example Cyber Cop Scanner.
- Very similar to intrusion detection systems are anti-virus products designed to detect malicious software such as viruses, trojan horses, worms, logic bombs and etc.
- Security/cryptographic systems, for example VPN, SSL, S/MIME, Kerberos, Radius etc.

A. Components

The typical components in an IDPS are sensor or agent, management server, database server and console [3].

Sensors and agents monitor and analyze activity. The term sensor is typically used for IDPSs that monitor networks, including network-based, wireless, and network behavior analysis technologies. The term agent is typically used for host-based IDPS technologies [4] [5].

A management server is a centralized device that receives information from the sensors or agents and manages them. Some management servers perform analysis on the event information that the sensors or agents provide and can identify events that the individual sensors or agents cannot. Matching event information from multiple sensors or agents, such as finding events triggered by the same IP address, is known as correlation. Management servers are available as both appliance and software-only products. Some small IDPS deployments do not use any management servers, but most IDPS deployments do. In larger IDPS deployments, there are often multiple management servers, and in some cases there are two tiers of management servers.

A database server is a repository for event information recorded by sensors, agents, management servers. Many IDPSs provide support for database servers.

A console is a program that provides an interface for the IDPS's users and administrators. Console software is typically installed onto standard desktop or laptop computers. Some consoles are used for IDPS administration only, such as configuring sensors or agents and applying software updates, while other consoles are used strictly for monitoring and analysis. Some IDPS consoles provide both administration and monitoring capabilities.

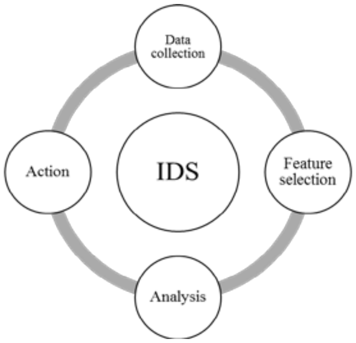

Fig. 2. Functionality of IDS

B. Functions

IDS consist of four major elements (Fig. 2) – data collection, feature selections, analysis and action.

The data collection is a file in which is recorded the data that should be analyzed. In rule based IDS the analysis is done by checking the data of compare it to a signature or pattern. Another method is anomaly based. The action defines the attack and reaction of the system.

III. Methods of Artificial Intelligence in Network and Information Security

As mentioned in the introduction to this article, world practice has already noted a significant number of various "Artificial Intelligence" applications in computer security. Without trying for a comprehensive classification, we could divide these methods into two main directions:

A. Conditionally named "distributed" or "network" methods:

A1. Multi-Agent Systems of Intelligent Agents;

A2. Neural Networks;

A3. Artificial Immune Systems and Genetic Algorithms, etc;

B. Conveniently named "compact" methods:

B1. Machine Learning Systems, including: associative methods, inductive logic programming, Bayes classification, etc.

B2. Pattern recognition algorithms;

B3. Expert Systems;

B4. Fuzzy logic, etc.

Having into account this variety of methods, it is of particular importance that adequate criteria are selected for the assessment and selection of a specific application for each specific solution. In the above mentioned project, the specification was carried out for two of the main sections of CTI.

IV. Traditional methods of AI for Cyber Intelligence – network methods

A. Intelligent agents

Agents can be defined to be autonomous, problem-solving computational entities capable of effective operation in dynamic and open environments [6]. Agents are often deployed in environments in which they interact, and may be cooperate, with other agents (including both people and software) that have possibly conflicting aims. Such environments are known as multi-agent systems. Agents can be distinguished from objects (in the sense of object oriented software) in that they are autonomous entities capable of exercising choice over their actions and interactions. Agents cannot, therefore, be directly invoked like objects. However, they may be constructed using object technology.

Agent architectures are the fundamental engines underlying the autonomous components that support effective behavior in real-world, dynamic and open environments. Agent-based computing has been a source of technologies to a number of research areas, both theoretical and applied. These include distributed planning and decision-making, automated auction mechanisms and learning mechanisms. Moreover, agent technologies have drawn from, and contributed to, a diverse range of academic disciplines, in the humanities, the sciences and the social sciences.

When designing agent systems, it is impossible to foresee all the potential situations an agent may encounter and specify behavior optimally in advance. Agents must therefore learn from, and adapt to, their environment. This task is more complex when the agent is situated in an environment that contains other agents with different (and in many cases unknown) capabilities, goals, and beliefs. Multi-agent learning, (the ability of agents to learn how to communicate, cooperate, and compete) becomes crucial in such domains. Learning is increasingly being seen as a key quality of agents, and research into learning agent technology, such as reinforcement learning and genetic algorithms, is now being carried out across Europe. Applications of learning agent technology have been especially successful in the areas of personalization and information retrieval, and promising results have been achieved in the areas of robotics and telecommunications. More effort will be needed, however, to make learning an inherent part of commercial agent applications.

B. *Datasets*

The Cyber Systems and Technology Group (DARPA Intrusion Detection Evaluation Group) of the MIT Lincoln Laboratory, under the sponsorship of the Defense Advanced Research Projects Agency (DARPA ITO) and the Air Force Research Laboratory (AFRL / SNHS), has collected, developed and disseminated the first standard evaluation of computer networks IDSes. This happened in 1998 and 1999, resulting in the first data sets DARPA 1998 and DARPA 1999, with DARPA 1998 being more popular.

KDD Cup is the annual Data Mining and Discovery Knowledge competition organized by the ACM Special Interest Group. The task of the competition is to develop a classifier capable of distinguishing between legitimate and illegal interactions in computer networks. For this purpose, a DARPA 1998 dataset was used during the race and subsequently became a common name under the name KDD'99.

Both DARPA and KDD data sets consist of nearly 5 million learning paths (i.e. events) labeled "penetration" or "no penetration" and a separate set of data - tests consisting of visible and invisible attacks. Each record consists of 41 different attributes that describe the different characteristics of the link, categorized as follows: basic TCP features, content features, time-based traffic characteristics, and host-based functions. All forms of attack fall into one of three categories: Remote-to-Local (R2L), User-to-Root (U2R), Service DOS (DOS) or Drilling [16].

As already mentioned, KDD'99 is based on DARPA 1998, which itself is strongly criticized by McHugh, largely due to the fact that it has the characteristics of synthetic data. As a result, some of the existing problems in DARPA 1998 remain in KDD'99. One of the most important flaws in the two sets of data is the vast number of abbreviated records, so training algorithms are made as frequent recordings, and thus prevent them from learning themselves (they are usually more harmful to networks like U2R and R2L).

Since there are currently only a few public datasets, such as DARPA 1998 and KDD'99, many of the experimental work of scientists is based on non-public or proprietary datasets. According to many scientists, and according to Tsai's report, it can easily be concluded that these two data sets are recognized as a de facto standard in the field of penetration detection.

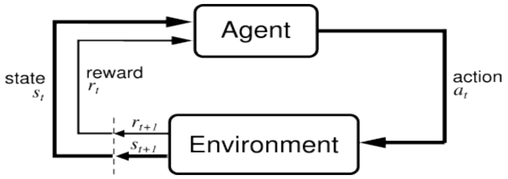

Fig. 3. Reinforcement Learning Algorithm

C. *Genetic algorithms*

Genetic Algorithms incorporate the concept of Darwin’s theory. They were inspired by the biological evolution (development), natural selection, and genetic recombination [5]. Genetic algorithms can be used to evolve simple rules for network traffic. GA generates a set of rules, that later can be used to distinguish the normal and abnormal network traffic. The algorithms that create these data sets use a chromosome-like data structure and evolve the chromosomes using selection, recombination and mutation operators.

GA contributes another type of anomaly-based IDPS, which is adept at utilizing communication energy [10] and applying a grammatical evolution (GE) technique with BNF grammar to identify route disruption attacks in mobile ad-hoc networks [11].

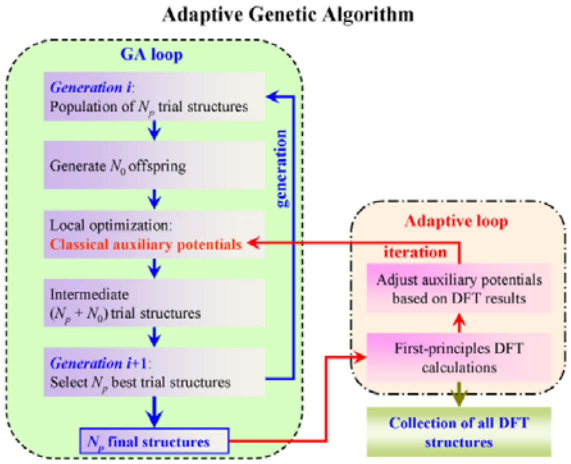
Adaptive Genetic Algorithm
GA loop
Generation i:
Population of N_p trial structures
Generate N_0 offspring
Local optimization:
Classical auxiliary potentials
Intermediate
$(N_p + N_0)$ trial structures
Generation i+1:
Select N_p best trial structures
generation
N_p final structures
Adaptive loop
iteration
Adjust auxiliary potentials based on DFT results
First-principles DFT calculations
Collection of all DFT structures

Fig. 4. Genetic algorithms

D. Artificial Neural networks

An Artificial Neural Network is an information processing system that is inspired by the way biological nervous systems. Neural networks are models built from multiple processing elements (neurons), each of which perform simple numerical operations and share results with their neighbors through weighted connections. Most of the intrusion detection systems based on the ANN are using two kinds of neural networks: multilayered feedforward neural networks and Kohonen's self-organizing maps [17].

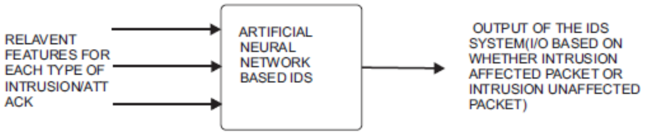
RELAVENT FEATURES FOR EACH TYPE OF INTRUSION/ATT ACK
ARTIFICIAL NEURAL NETWORK BASED IDS
OUTPUT OF THE IDS SYSTEM(I/O BASED ON WHETHER INTRUSION AFFECTED PACKET OR INTRUSION UNAFFECTED PACKET)

Fig. 5. Application of ANN for detection of intrusion for a class of intrusion [18]

E. Artificial Immune Systems

The Artificial Immune Systems (AIS) were inspired by the Human Immune System that is robust, decentralized, error tolerant, and adaptive [19]. The HIS is made of molecules, cells, and tissues that establish human body's resistance to infections caused by viruses and etc. The AIS can distinguish and eliminate the different pathogens from self-cells. This provides a great source of inspiration for the security of computer systems, especially IDS.

The first who began researches in this field are Farmer, Packard, and Perelson. Their algorithm describes a method for change detection that is based on the generation of T-cells in the immune system. In 1994 Forrest and her group at the University of New Mexico began research to build an IDS based on AIS. They proposed a negative selection algorithm to utilize the process of the HIS for a sophisticated anomaly-detection process [20].

V. Adaptive Optimization Techniques – Compact Methods

In recent years, intrusion detection and prevention systems have become very important. To cope with security attacks on infrastructure over the last decade, security organizations have paid special attention to cost savings, with the concept of IDPS being of relevant interest [7]. From this perspective, self-optimization typically comprises network parameter tuning. Nonetheless, the set of network parameters that can be optimized in a network is extremely large, as there are countless IDPS algorithms running on it and their parameters need to be optimized. In addition, even if the optimization process is only done on a few relevant parameters, the connection among parameter settings and network performance is not clear-cut. For this reason, IDPS parameter optimization should be performed intelligently. As a result, the IDPS would be able to amend its parameters in terms of intelligence-based IDPS (IIDPS) in order to achieve optimum performance with no human work. However, countless parameters can be changed remotely in the IIDPS system. From the operator's standpoint, adjusting IIDPS parameters that do not require time scheduling is the preferred alternative.
Types of Optimization Algorithms

Optimization Algorithm falls in 2 major categories -

1. **First Order Optimization Algorithms**—These algorithms minimize or maximize a Loss function **E(x)** using its ***Gradient*** values with respect to the parameters. Most widely used First order optimization algorithm is **Gradient Descent.**The First order derivative tells us whether the function is decreasing or increasing at a particular point. First order Derivative basically give us a **line** which is ***Tangential*** *to a point on its Error Surface.*
2. **Second Order Optimization Algorithms**—Second-order methods use the **second order derivative** which is also called **Hessian** to minimize or maximize the **Loss** function.The **Hessian** is a Matrix of ***Second Order Partial Derivatives***. ***Since the second derivative is costly to compute, the second order is not used much*** .The second order derivative tells us whether the ***first derivative*** is increasing or decreasing which hints at the function's curvature.Second Order Derivative provide us with a **quadratic** surface which touches the curvature of the **Error Surface**.

A. Fuzzy logic

This section presents the theoretical basis of the computational intelligence methodology known as Fuzzy Logic. This discipline was initiated by Lotfi A. Zadeh (1965) [8], professor at the University of California, Berkeley.

Fuzzy Logic emerged as an important tool for system control and complex industrial processes, as well as for home and entertainment electronics, diagnostic systems and other expert systems. Currently, a multitude of applications based on Fuzzy Logic are applied in many different areas, for instance control systems, robotics, medicine, pattern recognition, computer vision, information and knowledge management systems, earthquake prediction, scheduling optimization, etc. As an alternative to Classical Logic, Fuzzy Logic introduces a degree of imprecision when items are evaluated [9]. In real life, there is an abundance of knowledge that is ambiguous and imprecise, and human reasoning usually handles this kind of information. In this sense, Fuzzy Logic was designed specifically to imitate human behaviour. Additional benefits of Fuzzy Logic include simplicity and flexibility. In particular, this methodology can deal with problems with imprecise and incomplete data, and it can easily model non-linear functions of arbitrary complexity.

B. Reinforcement learning

In a particular environment, an agent can be encouraged to engage in a specific action that will lead to maximizing a cumulative reward. This type of machine learning is known as Reinforcement Learning. Its two defining characteristics are a trial and error search and actions with consequences that can affect immediate and future rewards [12].

A key concept in Reinforcement Learning is the trade-off between exploration and exploitation. When an agent is required to act, it will select an action that has yielded rewards in the past. Continuous decision making from an agent can be strengthened by reinforcement learning. In Reinforcement Learning the goal of the agent is formalized in terms of the reward signal. Reinforcement Learning agents try to maximize the cumulative reward they receive from the environment. In this context, important environmental properties and state signals, otherwise known as the Markov Property, can be found. In this context, important environmental properties and state signals, otherwise known as the Markov Property, can be found. The Markov property is fulfilled when a state signal retains all relevant information. In this situation, at time step t+1, the response of the environment is only dependent on time t. As such, the environmental dynamics can be defined as:

$$P_r\{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t\} \quad (1)$$

where $Pr\{.\}$ denotes the probability of its argument; s is the state of the environment; s' is any state in the system; r is the received reward; and a denotes the action taken by the agent. When the environment contains the Markov property, it is possible to predict the next state and rewards based on current states and actions.

The model of the environment is represented with a Markov Decision Process. A Markov decision process (MDP) is a Reinforcement Learning task with the Markov property. A finite MDP has finite states and actions and is further defined by a set of actions and states and the dynamics of the environment. The latter is specified by transition prospects and how valuable the next reward is expected to be. The transition probability for each probable next state (s’) for any current state (s) and any action (a) can be calculated using the equation below:

$$P_{ss'}^{a} = P_r\{s_{t+1} = s' \mid s_t = s, a_t = a\} \quad (2)$$

Likewise, the expected value of the next rewards for any current action a, state s, and next state s', is calculated as follows:

$$R_{ss'}^{a} = E\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\} \quad (3)$$

where $E\{\cdot\}$ is the expected value of its argument. The most important factors of a dynamic finite MDP are its transition probabilities and expected value of the next reward.

C. Q-learning adaptation

The Q-Learning algorithm was proposed as a way to optimize solutions in Markov decision process problems. The distinctive feature of Q-Learning is in its capacity to choose between immediate rewards and delayed rewards. At each step of time, an agent observes the vector of state xt, then chooses and applies an action ut. As the process moves to state xt+1, the agent receives a reinforcement r(xt, ut). The goal of the training is to find the sequential order of actions which maximizes the sum of the future reinforcements, thus leading to the shortest path from start to finish.

The transition rule of Q learning is a very simple formula:

$$\text{Q(state, action)} = \text{R(state, action)} + \text{gamma}* \\ *\text{Max[Q(next state, all actions)]} \quad (4)$$

The gamma parameter has a range of 0 to 1 (0<=gamma>1), and ensures the convergence of the sum. If gamma is closer to zero, the agent will tend to consider only immediate rewards. If gamma is closer to one, the agent will consider future rewards with greater weight, willing to delay the reward.

The Q-Learning algorithm goes as follows:

1. Set the gamma parameter, and environment rewards in matrix R.
2. Initialize matrix Q to zero.
3. For each episode:

Select a random initial state.

Do While the goal state hasn't been reached.

Select one among all possible actions for the current state.

Using this possible action, consider going to the next state.

Get maximum Q value for this next state based on all possible actions.

Compute: Q(state, action) = R(state, action) + gamma * Max[Q(next state, all actions)]

D. Adaptation of multi-agent based fuzzy reinforcement learning

Assumptions are made in multi-agent environments to assure that convergence will occur. However, these assumptions are frequently violated. Complexities are created even in simple situations where agents share a common, stationary setting and are required to only learn a strategy for a single state. In situations where the agents have opposing goals there may be no optimal solutions and establishing equilibrium between agents becomes the primary goal; essentially, agents are unable to improve their payoffs because other agents keep their actions fixed.

Dynamic environments not only have multiple agents, but they also have multiple, sequential decisions that increase their complexity. In these settings, agents must coordinate and consider the current state of their dynamic environment with very limited information. Typically, agents in dynamic environments cannot observe the actions of other agents or see what rewards they obtain as a consequence although the actions of the other agents influence their immediate environment along with the rewards they can obtain. In very complex environments agents may be unaware that other agents are present and may interpret their environment as non-stationary. Similar, equally complex environments allow agents to access information, but the state action spaces are not conducive to learning because of their complexity and the amount of coordination required between agents. Before an effective multi-agent approach can be developed, all these challenges must be addressed. A standard multi-agent reinforcement learning model is presented in Fig.6.

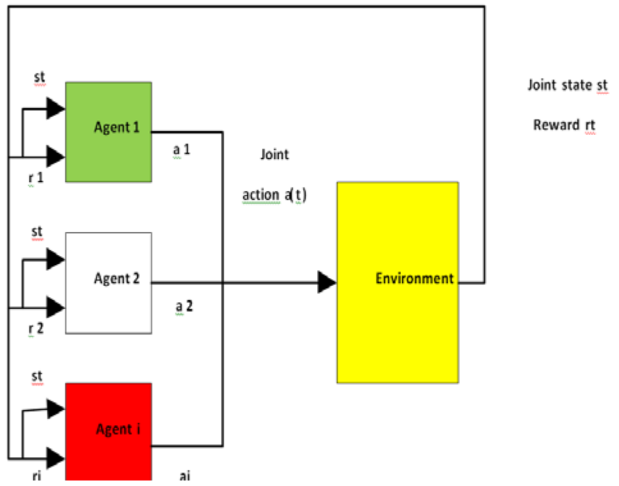

Fig. 6 Multiple-agents acting in the same environment

Regardless of learning complexity, the demand for multi-agent systems continues to increase. In cases of systems that are decentralized and single, central learning methods are impractical. Such systems can be found when data was subjected to disruptions caused by multiple, conflicting objectives or if a single centralized controller requires too many resources. Multi-robot setups, distributed load balancing, decentralized network routing, electronic auctions, and systems designed to control traffic are all examples of such system types.

E. Game theory

The game theory provides a model of strategic interactions based on individuals competing against each other in a game. A mathematical object is used to represent the game as it outlines the consequences of the interactions between players in terms of the rewards to be obtained. AI researchers often rely on extensive game forms where players take turns performing an action to model classic minimax algorithms [13].

F. k-means

k-Means is a method of partitioning in data mining. This method divides the data into a number of clusters using the Euclidean similarity in distances. Objects are classified or grouped on the basis of cluster characteristics.

The Euclidean equation for finding the distance between two objects is:

$$D(a,b)= D(b,a)= |a-b|= \sqrt{\sum_{i=1}^{n}(bi\ -\ ai)^2} \quad (5)$$

where a and b are both objects, and bi and ai are their coordinates in the n-dimensional space.

Basic steps for clustering data using k-means:

1. Select any number of cluster centers;
2. Each object is assigned to the nearest center using the Euclidean equation;
3. Move each center in the middle of the objects assigned to it.

Steps 2 and 3 are repeated while the difference in the change is less than a predetermined threshold. An advantage is the relative efficiency of grouping undisclosed data, but as a disadvantage - it cannot cope with noised data.

G. Swarm Intelligence

Swarm Intelligence (SI) is a paradigm of distributed intelligence and innovative in optimizing troubleshooting inspired by collective behavior of many living beings. Typically comprise a population of agents (able to perform various tasks) interacting among themselves and with the surrounding environment. The absence of a single control structure, local interactions among these agents lead to the emergence of selforganizing global behaviors [14].

Many optimization algorithms such as Ant colony optimization, Particle Swarm Optimization, Bacterial foraging optimization, Bee Colony Optimization, Artificial Immune Systems, Firefly algorithm, Gravitational search, Biogeography-Based Optimization, Bat algorithm and Krill herd are inspired by the metaphors of this behavior [15]. The following subsections examine in general some of these new algorithms paradigms

VI. Conceptual Model of a Computer Network Security System

Fig. 7 shows the combination of Network and Host-based IDPS (NIDPS, HIDPS) in a fully distributed framework structure in a networking environment with Collaborative Intelligent IDPS This formation is readily applicable to any network and its requirements. It demonstrates the impact of a Multi-agent system-based computational intelligence technique on enhancing detection efficiency and false alarm rates. In the context of Collaborative Intelligent IDPS, adaptive game theoretic techniques are adequate for network parameter optimization due to network complexity and dynamism. The main benefits of applying such techniques are cost savings and improved network performance.

The Collaborative Intelligent IDPS complexity incorporates three basic concepts of detection management: Fuzzy Reinforcement Learning, Knowledge Management (KM), and Multi-agent Management (MA) into the core architectural design. The collaborations between the optimization techniques portrays the clear notion of cooperative learning-based detection to satisfy the requirements of Intelligent IDPS.

The conceptual model of a computer network security system is based on three layers shown in Fig. 8. The first layer shows the traditional system components that monitor and collect the audit data through the sensors, analyses the data and detect intrusions, generate alarms and herald the proper response through the actuators. The next two layers are advanced layers based on the proposed collaborative intelligent IDPS.

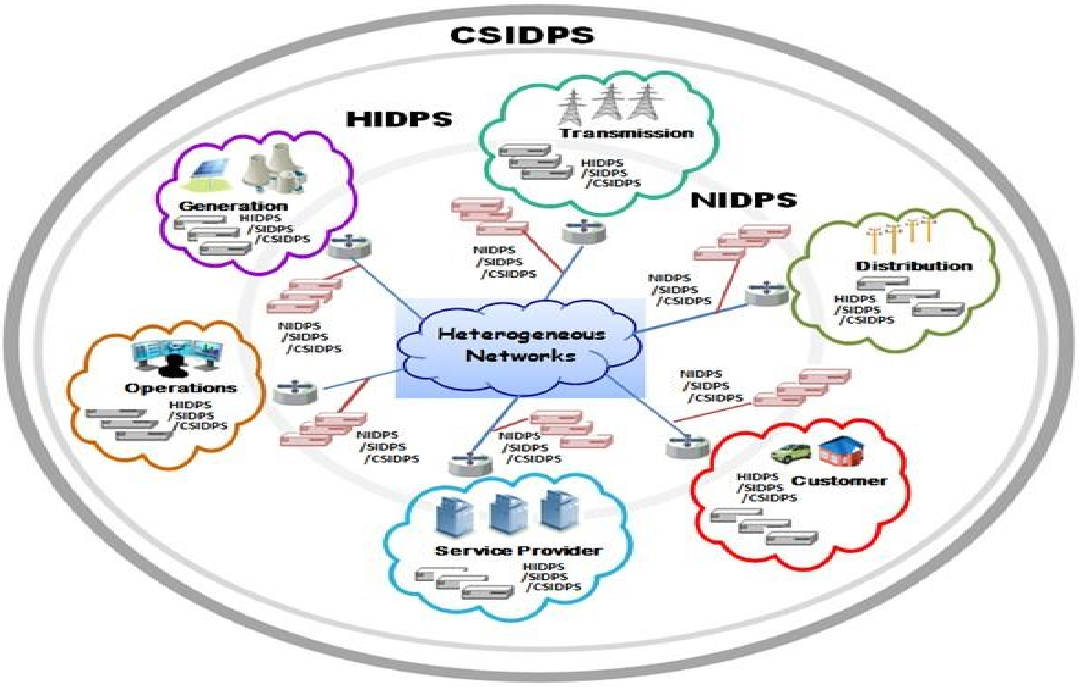

Fig. 7. Collaborative Intelligent IDPS

The second layer shows the traditional artificial intelligence methods that are communicating to each other for more accurate results. Knowledge management enables the characterization of anomaly profile knowledge as a set of related concepts within an anomaly calculator domain. The policy aspect of a multi-agent manager is thus utilized to predict anomaly behavior. They combined with the adaptive optimization techniques such as neuron networks with fuzzy logic and reinforcement leaning to detect intrusions and feed the obtained results into the autonomic solution mode components comprising a self-optimizer, self-learning, and self-configuration (third layer). The last layer components are defined in autonomic computing principles in real-time without human intervention. This removes the human factor as the cause of errors. The arrows point out the information flow between components while the dash arrows indicate the logical communications between the components.

VII. Conclusion

Finally, this article was devoted to Reinforcement Learning, which is the method selected amongst several ones. The main benefit of this approach is that Reinforcement Learning algorithms learn from interaction, which becomes essential in complex systems such as networks. The reinforcement learning emerged as a result of applying fuzzy techniques to Collaborative Intelligent IDPS, leading to robust, fault-tolerant and easy to manage and operate networks. Knowledge management enables the characterization of anomaly profile knowledge as a set of related concepts within an anomaly calculator domain. The policy aspect of a multi-agent manager is thus utilized to predict anomaly behavior. In summary, the scalable, fully distributed structure of our system exposes the risks of low accuracy detection and difficulty in synchronizing information between autonomous agents.

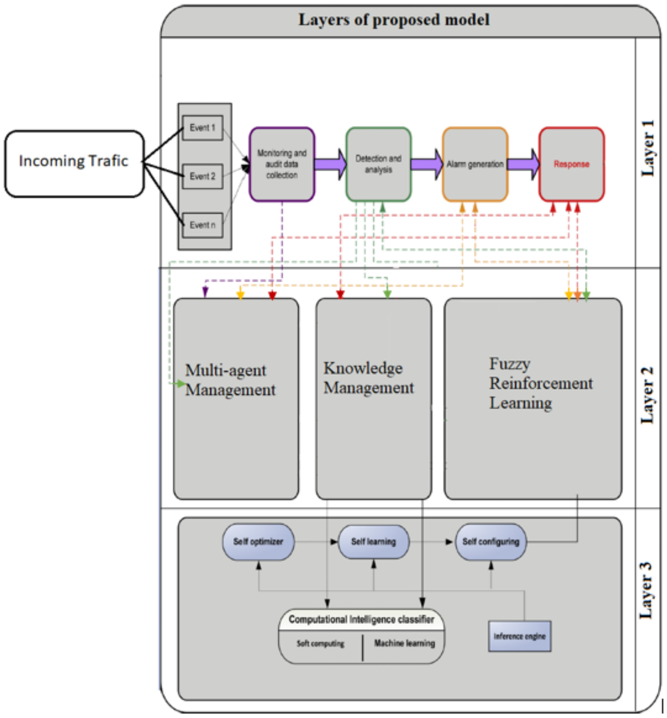

Fig. 8.Conceptual model for IDP system

VIII. Future work

In addition, a collaborative Network and Host-based IDPS (NIDPS, HIDPS) in a fully distributed framework structure in a networking environment with AI is readily applicable to any network and its requirements and the results are encouraging. The goal is to port the abstract network model to a realistic network simulation.

References

[1] IBM, "IT xecutive Guide to Security IntelligenceE," IBM, 2013.

[2] N. B. e. a. Idris, "Artificial Intelligence Techniques Applied to Intrusion Detection," *Annual IEEE in INDICON*, pp. 52-55, 2005.

[3] K. Scarfone and P. Mell , Guide to Intrusion Detection and Prevention Systems (IDPS), National Institute of Standards and Technology, 2007.

[4] M.-J. Kang and J.-W. Kang, "Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security," PLOS one, 7 Jun 2016. https://www.ncbi.nlm.nih.gov/pmc/ articles/PMC4896428/.

[5] A. S. A. Aziz, M. Salama and A. e. Hassanien, "Detectors Generation using Genetic Algorithm for a Negative Selection Inspired Anomaly Network Intrusion Detection System," in *Federated Conference on Computer Science and Information Systems*, WROCŁAW, 2012.

[6] "Host- vs. Network-Based Intrusion Detection Systems," SANS Institute 2000 - 2005, 2016.

[7] A.-S. K. Pathan, "The State of the Art in Intrusion Prevention and Detection," CRC Press, Florida, 2014.

[8] L. A. Zadeh, "Fuzzy sets. Information and Control," vol. 8, no. 3, pp. 338-353, 1965.

[9] R.-E. e. a. Precup, "A survey on industrial applications of fuzzy control," *Computers in Industry*, vol. 62, no. 3, pp. 213-226, 2011.

[10] R. e. a. Khanna, "Reduced Complexity Intrusion Detection in Sensor Networks Using Genetic Algorithm," in *IEEE International Conference on Communications*, Dresden, 2009.

[11] M. e. a. Phillips, "Breath biomarkers of active pulmonary tuberculosis," *Tuberculosis*, vol. 90, no. 2, pp. 145-151, 2010.

[12] R. S. e. a. Sutton, "Reinforcement learning: An introduction," Cambridge Univ Press, 1998.

[13] S. J. e. a. Russell, "Artificial intelligence: a modern approach," Prentice hall Englewood Cliffs, New Jersey, 1995.

[14] Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. Information Sciences, 237, 82–117. doi:10.1016/j.ins.2013.02.041.

[15] A. Engelbrecht, Fundamentals of Computational Swarm Intelligence, Wiley, 2006.

[16] H. G. Kayacık, and N. Z. Heywood, "Analysis of three intrusion detection system benchmark datasets using machine learning algorithms," in Proc. Intelligence and Security Informatics, pp. 362-367, Springer Berlin Heidelberg, 2005

[17] A. Vesely and D. Brechlerova, "Neural networks in intrusion detection systems," Agric. econ. - Czech, vol. 50, pp. 35-39, 2004.

[18] A. BIVENS, C. PALAGIRI, R. SMITH, B. SZYMANSKI and M. EMBRECHTS, "NETWORK-BASED INTRUSION DETECTION USING NEURAL," in Intelligent Engineering Systems through Artificial Neural Networks ANNIE-2002, St. Louis, 2002.

[19] A. S. A. Aziz, M. Salama and A. e. Hassanien, "Detectors Generation using Genetic Algorithm for a Negative Selection Inspired Anomaly Network Intrusion Detection System," in Federated Conference on Computer Science and Information Systems, WROCŁAW, 2012.

[20] S. Forrest, S. A. Hofmeyr and A. Somayaji, "Computer Immunology," Commun. ACM, vol. 40, no. 10, p. 88–96, 1997.