IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 2, 2020, accepted September 17, 2020, date of publication September 28, 2020,
date of current version October 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027463

A Systematic State-of-the-Art Analysis of Multi-Agent Intrusion Detection

IMTITHAL A. SAEED[1,2], ALI SELAMAT[1,3,4,5], (Member, IEEE), MOHD FOAD ROHANI[1], (Member, IEEE), ONDREJ KREJCAR[4], AND JUNAID AHSENALI CHAUDHRY[6], (Senior Member, IEEE)

[1]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia UTM, Skudai 81310, Malaysia
[2]College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 16278, Saudia Arabia
[3]Media and Games Center of Excellence (MagicX), Universiti Teknologi Malaysia UTM, Skudai 81310, Malaysia
[4]Center for Basic and Applied Research, Faculty of Informatics and Management, University of Hradec Kralove, 500 03 Hradec Kralove, Czech Republic
[5]Malaysia-Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia
[6]Duja Inc., Perth, WA 6004, Australia

Corresponding author: Ali Selamat (aselamat@utm.my)

This work was supported in part by the Universiti Teknologi Malaysia (UTM) through Research University under Grant Vot-20H04, in part by the Malaysia Research University Network (MRUN) under Grant Vot 4L876, in part by the Fundamental Research Grant Scheme (FRGS) through Ministry of Education Malaysia for the Completion of the Research under Grant Vot 5F073, in part by the SPEV Project, University of Hradec Kralove, FIM, Czech Republic (2020), and in part by the Ph.D. Student Sebastien Mambou in for Consultations Regarding Application Aspects.

ABSTRACT Multi-agent architectures have been successful in attaining considerable attention among computer security researchers. This is so, because of their demonstrated capabilities such as autonomy, embedded intelligence, learning and self-growing knowledge-base, high scalability, fault tolerance, and automatic parallelism. These characteristics have made this technology a de facto standard for developing ambient security systems to meet the open and dynamic nature of today's online communities. Although multi-agent architectures are increasingly studied in the area of computer security, there is still not enough empirical evidence on their performance in intrusions and attacks detection. The aim of this paper is to report the systematic literature review conducted in the context of specific research questions, to investigate multi-agent IDS architectures to highlight the issues that affect their performance in terms of detection accuracy and response time. We used pertinent keywords and terms to search and retrieve the most recent research studies, on multi-agent IDS architectures, from the major research databases and digital libraries such as SCOPUS, Springer, and IEEE Explore. The search processes resulted in a number of studies; among them, there were journal articles, book chapters, conference papers, dissertations, and theses. The obtained studies were assessed and filtered out, and finally, there were over 71 studies chosen to answer the research questions. The results of this study have shown that multi-agent architectures include several advantages that can help in the development of ambient IDS. However, it has been found that there are several issues in the current multi-agent IDS architectures that may degrade the accuracy and response time of intrusions and attacks detection. Based on our findings, the issues of multi-agent IDS architectures include limitations in the techniques, mechanisms, and schemes used for multi-agent IDS adaptation and learning, load balancing, scalability, fault-tolerance, and high communication overhead. It has also been found that new measurement metrics are required for evaluating multi-agent IDS architectures.

INDEX TERMS Multi-agent, IDS architectures, intrusion detection, attacks, review, malware, cyber-physical system.

I. INTRODUCTION

The openness and dynamic nature of today's online communities have raised a lot of security concerns on our digital valuables. Intrusions and attacks are intentionally designed to corrupt information and network communications. Recently, they have been rapidly increasing due to the extensive use of computer networks. A new report released by Semantic says that they record events from 123 million attack sensors worldwide, block 142 million threats daily, and monitor threat activities in more than 157 countries [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Khandaker.

For protecting computer networks, Intrusion Detection System (IDS) is used to detect intrusions and attacks in real-time by analyzing network activities using statistics [2]–[4], rules [5]–[8] or machine learning [9]–[11]. IDS can be host-based, network-based, or hybrid. Host-based IDS is used for monitoring and analyzing the internal computer system state [12], while network-based IDS is to monitor and analyze the external computer system state(network traffic) [13]. The hybrid IDS combines both host-based and network-based. After the expansion in computer networks usage, the traditional host-based and network-based IDSs that use statistical, rule-based, or conventional machine learning methods had been ineffective in detecting the massive, sophisticated attacks that invade computer networks in extremely high speeds. As a result of this, significant progress has been made in IDS research by moving from the traditional methods towards more intelligent techniques.

Several multi-agent IDS architectures have been proposed using machine learning and other advanced computational intelligence methods [14]–[20], [28]. The main difference between the proposed multi-agent IDSs and the traditional distributed IDSs is that multi-agent IDS deals with distributed problem solving and how agents interacts to detect attack incidents in computer networks, while the traditional IDS is about the distribution of the IDS architecture itself [21]. Besides that, multi-agent systems own unique capacities such as autonomy, portability, mobility, and social capabilities [22]. All these capacities motivated researchers to use this technology as it is suitable for solving the complex tasks of intrusions and attacks detection especially in the open and dynamic online environments [33].

Although the multi-agent architectures are increasingly studied in the area of computer security, but still there is no enough empirical evidence on their performance in intrusions and attacks detection. This lack of evidence limits the utilization of multi-agent technology in IDS research. By conducting this SLR, we will provide an up-to-date comprehensive reference for IDS researchers and developers to start new research and utilize the best techniques in the literature.

The aim of this systematic literature review (SLR) is to investigate the most recent multi-agent IDS architectures to highlight the issues that may affect the performance of intrusion detection in terms of accuracy and response time. To achieve this aim, we focused on four aspects of multi-agent IDS architectures: (1) The classification of multi-agent IDS architectures, (2) The aspects that influence intrusion detection performance in terms of accuracy and response time, (3) The limitations of these aspects (4) The measurement metrics that are used to evaluate multi-agent IDS architectures.

This SLR was carried out in four phases, firstly: a set of research questions were formulated based on the study aim. Secondly, search processes were launched, and materials were collected. Third, the collected materials were assessed and filtered out to choose the most relevant studies for providing answers to the research questions. Finally, data from the selected studies were synthesized and compared according to the formulated research questions. This SLR is limited to answer the formulated research questions. The data collected is related to the selected studies that were chosen using the selection criteria stated in subsection III-A3. This SLR is not concerned with the traditional host based and distributed network based IDS.

The remaining parts of this paper are organized as follows: Section II reports related works on multi-agent IDS literature review. Section III describes the methodology followed in carrying out this research. Section IV presents the discussion of the results based on the formulated research questions. Section V summarizes the publication limitation. Section VI is the conclusion of this study.

II. RELATED WORK

According to our investigation, to date, there is no SLR conducted on multi-agent IDS architectures. However, a limited number of literature reviews have been reported in this domain. The report in [23] discussed and summarized the use of mobile agents in intrusions detection along with their advantages and disadvantages. The advantages include reducing network latency and traffic, asynchronous execution and autonomy, structure and composition, dynamic adaptability, dealing with heterogeneity, fault tolerance, and scalability. The report also raised issues on mobile agent security, performance, code size, lack of a priori knowledge, and coding and deployment difficulties. The study in [24] investigated the immunological essentials in designing a multi-agent IDS. The early five autonomous agent architectures for distributed intrusion detection were evaluated. The literature review in [25] investigated network and agent based IDSs. The first part of the review focused on the network based IDSs while the second part shed the light on IDS based on mobile agents.The review enumerated the architectural characteristics and advantages of using mobile agents, and discussed three of the agent based IDS architectures. The study in [26] discussed multi-agent IDS architectures based on immune system algorithms. The study focused on agents' roles, architectural characteristics, and the security mechanisms used for securing computer network. The survey study in [27] reviewed the existing trends in IDS. Beside highlighting the advantages and disadvantages of the data mining and soft computing techniques used in intrusions detection, the survey also discussed agent based IDS, honeypots and honeynets. The literature review in [28] investigated the multi-agent IDS from an architectural point of view. There were approximately 15 studies related to IDS based on stationary agents and 15 studies related to IDS based on mobile agents. The review focused greatly on the types and distribution of the agents used. Also, the review discussed how a multi-agent IDS is constructed and how data flows.

The review paper in [29] presented a classification for the typical IDS and then conducted a strategical review on the existing mobile agent-based IDSs focusing on their classification, architectures, mode of data collection, data analysis techniques, and their security.

In our previous paper [30], we studied the evolution of malware detection systems from an architectural perspective and detection techniques used. The study also highlighted the importance of agent-based architectures in the domain of IDS. The review paper in [31] presented a classification for agent types, and the advantages and disadvantages of using agents. The review summarized agents' advantages in: the asynchronous autonomous interactions, reduction of network load, dealing with heterogeneity and ease of configuration. On the other hand, this review summarized the disadvantages of using agents in security issues and absence of common language. The review paper in [32], focused on classifying the existing wireless IDS techniques based on target wireless network, detection techniques, data collection process, trust model and analysis techniques. The pros and cons related to four of the proposed architectures were highlighted. These pros and cons that concern agents' interaction, coordination, management and data analysis.

The literature reviews reported above do not provide comprehensive details on the characteristics of multi-agent architectures that improve the performance of intrusions detection. There are essential characteristics that need to be embedded in multi-agent IDS architectures to improve the performance of attacks detection such as self-learning, adaptation, scalability, load balance, fault tolerance, self-management, self-configuration, and robustness.

This review provides new in-depth analysis for the major properties and characteristics that greatly impact intrusion detection performance. We highlighted the most important limitations of these properties such as multi-agent organizational structure, and computational components (data collection, data synchronization and data analysis). We also outlined the shortcomings of multi-agent IDS properties and characteristics. The finding of this review have not been addressed in any of the other reviews.

III. RESEARCH METHODOLOGY

A systematic literature review (SLR) is a specific research methodology used by researchers to gather and evaluate the evidences related to the topic under investigation. The protocol developed in [34], for conducting a systematic literature review in software engineering was utilized in this SLR to design the research plan described in the following subsections.

A. RESEARCH PLAN

The plan of this SLR consists of four phases: In phase 1, the research questions were formulated In consistency with the main goal of the study. In phase 2: search strategy was determined to specify search elements such as choosing search keywords, defining search strings, and determining electronic resources. This phase also included the tasks of choosing a reference management software tool, executing search processes, and collecting studies. In phase 3, we execute the selection policy that prepared for choosing the relevant studies, and quality assessment. Finally, in phase 4, we analyze data synthesis. These phases are summarized and illustrated in Figure 1. The subsequent sections discuss each phase in more details.

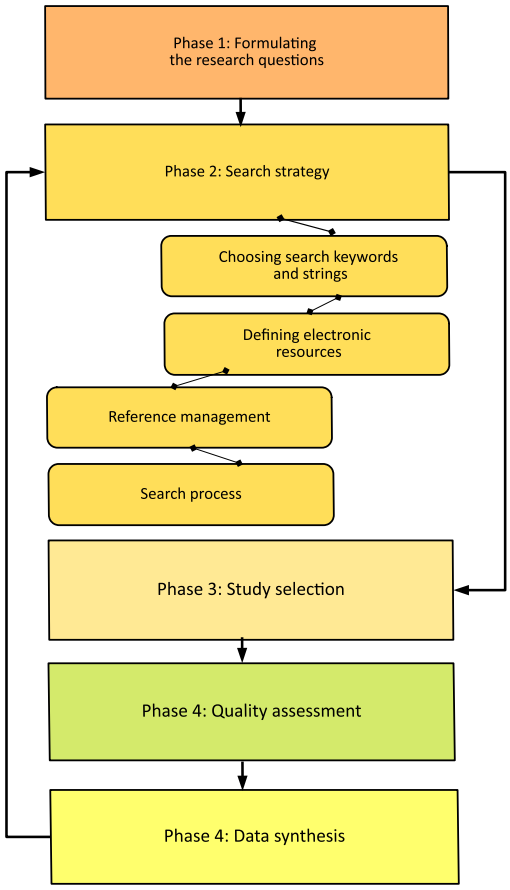


FIGURE 1. Systematic literature review plan.

1) RESEARCH QUESTIONS

This systematic literature review aims to investigate the current multi-agent IDS architectures to identify the most challenging limitations that affect intrusions detection performance. Four research questions were formulated as follows:

- RQ 1: What classifications exist for multi-agent IDS architectures?
- RQ 2: What aspects of multi-agent architectures influence intrusion detection performance in term of speed and accuracy?
- RQ 3: How the characteristics of multi-agent IDS architectures impact the speed and accuracy of intrusion detection?
- RQ 4: To what extent the metrics used to measure and evaluate multi-agent IDS architectures are suitable and sufficient?

To ensure robust and precise research questions, we carried out two step-verification: firstly, the research questions were formulated by referring to the recommendations stated in [36]. Secondly, the formulated research questions were validated and cross-checked by experts in the same field.

2) SEARCH STRATEGY

In this phase, we chose the search keywords and strings, determined the electronic resources, selected a reference management tool, and defined the search execution process. In the following subsections, each process is explained in detail.

a: SEARCH KEYWORDS AND STRINGS

The search keywords and strings were derived from the research questions of this SLR. Synonyms and alternatives of the keywords and terms were also included in the search keywords. The synonyms, keywords, and terms were taken from the relevant research papers in the field of multi-agent IDS. The following examples explain the strings used in the search sentences:

''multi-agent attack detection architecture'', ''multi-agent based malware detection architecture'', ''multi-agent intrusion detection architecture '', ''agent based intrusion detection architecture'', ''cooperative IDS''

b: ELECTRONIC RESOURCES

For retrieving the relevant studies, search electronic resources were determined. We chose to retrieve studies from journals, digital archives, digital libraries, and online bibliographic databases. Examples of these resources include ACM Digital Library, Springer, ScienceDirect, IEEE Xplore, Google Scholar, and Google search engine.

c: REFERENCE MANAGEMENT

A large number of studies were retrieved from the online resources by using the search strings and keywords. The retrieved materials were collated and organized by a reference management software called EndNote.[1] This made it easy, adding and removing the studies whenever it is required.

d: SEARCH PROCESS

The search processes were launched on online electronic resources to retrieve journal articles, conference papers, book chapters, and theses. The references were recorded, and the full pdf files were downloaded and stored. These search operations resulted in more than 1000 studies. The EndNote was used to combine each reference with its' related pdf file to make it easy to read the papers. Afterward, the study selection process was applied to filter out the unrelated studies.

3) STUDY SELECTION

To choose the most relevant studies, two procedures were conducted. Firstly, the studies' titles and abstracts were manually checked. The studies that matched the aim of the research were selected, while the other studies that do not match the aim of this SLR were discarded. This process resulted in more than 220 relevant studies. The relevant studies were collated, and their bibliographic information was checked. Secondly, study selection was performed by applying the inclusion/exclusion criteria described in Table 1. This selection operation resulted in more than 70 relevant materials. The search process and study selection are illustrated in Appendix G.

TABLE 1. Filtering criteria.

Inclusion	Exclusion
1.Papers published during 2010-2019 2.Papers are written in English. 3.Papers with identified source reference.	1. Papers published before the year 2010. 2. Repeated papers. 3. Papers with unidentified reference. 4. Papers with unidentified reference. 5. Paper focusing specifically on agent-based IDS in Bluetooth, MANET, Ad hoc networks, and smartphone.

In addition to the described method for including and excluding studies, manual checking was performed on the selected studies, and more relevant studies were added or removed to the group. Also, studies with unknown reference sources were excluded. This process finally resulted in 71 studies, including 37 journal articles, 21 conference papers, 7 book chapters, and 6 theses. This is illustrated in Appendix H.

4) QUALITY ASSESSMENT

Quality assessment is an important step to show the reliability level of the studies under investigation. The assessment process was conducted based on two elements: first, the studies were assessed based on their reference type and grouped by their year of publication. We chose only the most recent studies with popular and indexed reference sources. Then, we evaluated the contents of the selected studies. In the following subsections, the assessment of the selected studies is explained.

a: QUALITY ASSESSMENT BASED ON REFERENCE TYPES AND PUBLICATION YEAR

In this step, the studies were evaluated using two measurements: material type and year of publication. First, after the selected studies had been chosen, their bibliographic information was checked. A lot of materials, including journal papers, book chapters, and theses, were collected. The sources of the collected materials were identified. The majority of the materials were indexed under the major indexing databases such as SCOPUS, ISI, and IEEE. Figure 2 shows the percentages of the selected studies per their indexing types.

Second, for sound review, the selected studies should reflect the state-of-the-art of multi-agent IDS. All the selected studies were from 2010 to 2019. Figure 3 shows the distribution of the studies per year of publication.

More details on reference and sources of the selected studies are shown in Appendix D, and Appendix E.

b: QUALITY ASSESSMENT BASED ON THE STUDIES CONTENTS

To evaluate the contents of the selected studies, we developed quality assessment questions to be answered by either ''yes'',

[1] Software program for reference management

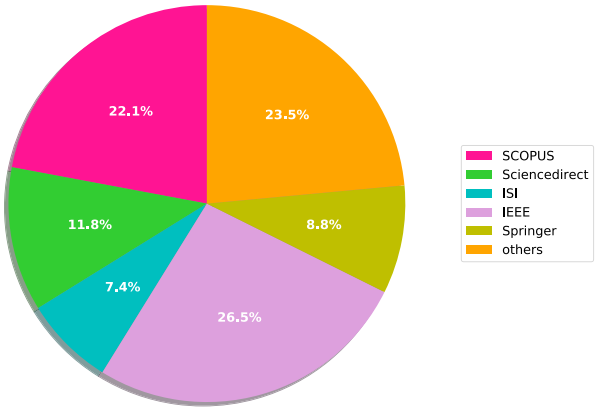


FIGURE 2. Selected studies per indexing type.

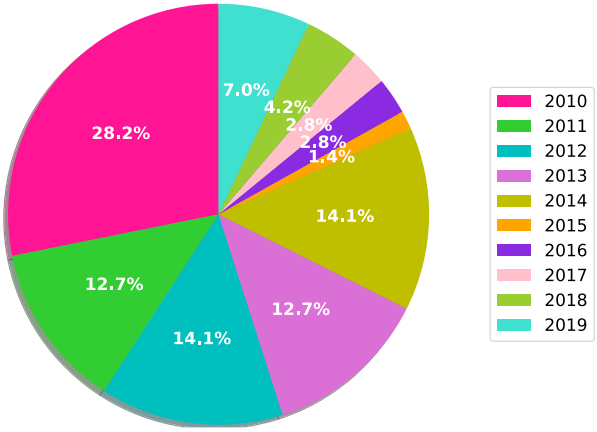


FIGURE 3. Selected studies per years.

TABLE 2. Quality assessment criteria.

Q#	Questions	Score
Q 1	Is the objective of the study clearly stated?	"yes" =1, "no"=0, "somehow="0.5"
Q 2	Is the proposed architecture properly described?	"yes" =1, "no"=0, "somehow="0.5"
Q 3	Are the purposed approaches supported by experiments?	"yes" =1, "no"=0, "somehow="0.5"
Q 4	Are the experiments properly designed to reflect the studies' purposes (in term of measurements and metrics)?	"yes" =1, "no"=0, "somehow="0.5"
Q 5	Are the results clearly discussed?	"yes" =1, "no"=0, "somehow="0.5"

''no'' or ''somehow''. These questions are shown in Table 2. Each question must be answered by an option that has a number associated with it: ''yes'' = 1, ''no'' = 0, and ''somehow'' = 0.5. The total score for each study was computed by averaging all the scores. To ensure that the selected studies are reliable, we only considered the studies with scores above 50% because those rated below 50% are either conceptual papers or include frameworks for other papers that already included in the selected studies.

Using this method for weighting the selected studies is very effective in giving insights about the reliability of the studies' contents. The quality scores for the selected studies are shown in Appendix F.

5) DATA SYNTHESIS

In this step, the scrutinized papers were carefully reviewed, compared, collated, and summarized according to the formulated research questions. The papers' summaries include qualitative data such as characteristics, properties, and performance metrics of multi-agent IDS architectures. We organized these summaries in table formats to help to analyze and interpret the results.

Data related to the research question RQ 1 was extracted and organized in a tabular format. Appendix A illustrates the architectural characteristics and properties of multi-agent IDS, and Appendix B illustrates agent types exhibited by the selected studies. To synthesize the data related to the research question RQ 2, the aspects of multi-agent IDS architectures that influence intrusion detection performance were identified and placed in tables for coherent analysis. This is shown in Appendices B and C. The limitations of multi-agent IDS architectures that related to research question RQ 3 were extracted and organized in textual formats. The metrics used for evaluating multi-agent IDS architectures, question RQ 4, were enumerated in textual forms. Synthesizing the results data into tables would help in making coherent analysis and interpretation for the research findings.

IV. RESULTS DISCUSSION

The aim of this SLR was to investigate the current multi-agent IDS architectures to highlight the issues that affect intrusions and attack detection in terms of accuracy and response time. To achieve this goal, the discussion scope was centred around answering four research questions, explained in section III-A1. As a basis for our analysis, we mainly considered the synthesized results described in subsection III-A5. In the following subsections, we present detailed descriptions, comparisons, analysis, and interpretations of the findings. To maintain adequate focus and flow, the discussions are ordered according to the research questions.

A. RESEARCH QUESTION RQ 1 (WHAT CLASSIFICATIONS EXIST IN MULTI-AGENT IDS ARCHITECTURES?)

The results revealed that multi-agent IDS architectures have three classifications: organizational structures, agent types, and computational components classification. The data related to this research question is shown in the appendices A, B, and C. In the following subsections, we will give detailed explanations of each classification.

1) THE CLASSIFICATION OF ORGANIZATIONAL STRUCTURES

Organizational structures provide frameworks for agents' interactions through the definition of roles, behaviour expectations, and authority relations. They also impose constraints on the ways agents communicate and coordinate [22]. The results uncovered that multi-agent IDS architectures constitute three organizational structures. First, the hierarchical structure that is the most common among the current

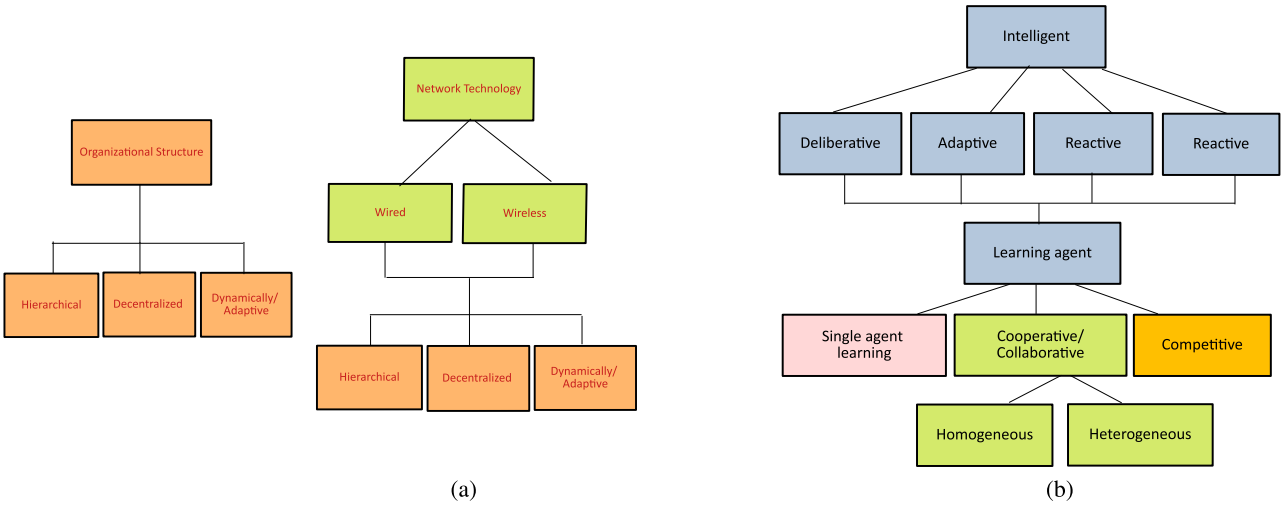


FIGURE 4. (a) Multi-agent IDS architectures based on organizational structure. (b) Multi-agent IDS architectures based on agents types.

multi-agent IDS architectures, it was adopted by 27 studies, see Appendix A. In this structure, agents are assigned specific tasks and distributed over the network in a hierarchical manner, and data flows from multiple sensors, located at the bottom of the hierarchy, to one control agent at the top. The main issue of this structure is the central point of failure.

Second, a decentralized (or distributed) organizational structure is also found. In this structure, the architecture agents organized in small groups belonging to the same subnet or network segment. The data flow from multiple sensors to one control agent residing in the same network segment. In some studies, the analysis agents are organized in a layered style, which resembles the hierarchical model, and the results of lower analysis agents fed to the analysis agents of the next layer until it reaches to the security centre on the top of the hierarchy [19]. This structure was adopted by 24 studies, see Appendix A. The main issue of this structure is its complexity Third, a dynamically adaptive structure was also found. In this structure, the architecture agents change their behaviour dynamically to adapt to network changes [5], [35], [37]. The dynamic behaviour of this structure is usually implemented using mobile agents or by removing and instantiating agents immediately when changes happen in a network. The agents of this structure incorporate adaptation techniques to enable them to respond to environmental changes. This structure was adopted by 20 studies. The dynamic adaptive structure is the most appropriate structure for open and dynamic environments. The disadvantage of such a structure is its complexity.

Additionally, the considerable development of computer network technologies resulted in very different network types such as LAN, WAN, MAN, VPN, Adhoc, MANET, and others. In this study, we consider only the main two categories of computer networks, wired and wireless. Generally, the organizational structures of multi-agent IDS Depend on the computing environment they deal with, for example, the wireless network is dynamic and requires a dynamically adaptive structure to face the dynamism of such networks. Figure 4a shows the classification of the multi-agent IDS architectures based on their organizational structure.

2) THE CLASSIFICATION OF GENTS' TYPES

Agent types represent the features that describe the internal capabilities and functionalities agents have [39]. The results showed that the existing multi-agent IDS architectures encompass several agent types, including autonomous, intelligent, adaptive, reactive, proactive, cooperative, collaborative, and deliberative agents. Almost all the studies under investigation adopted autonomous agents in their approaches. The autonomous agents have the ability to work and maintained by their own [40]. All the proposed architectures adopted intelligent agents by incorporating Artificial Intelligence(AI) techniques like machine learning, soft computing, and immune system mechanisms. Several of the proposed approaches adopted adaptive agents that can change their behaviour according to the environment changes. Also, they can adjust their abilities depending on the parameters they received from the other interacting agents [20]. Also, the results show that the current architecture incorporates reactive and proactive agents. The agents that do misuse detection are considered proactive, while the agents that do anomaly detection are considered as reactive agents. The reactive agents monitor their environment and react to the changes that occur in a timely fashion. Reactivity of an agent also means the ability to immediately adjust its behaviour when the environment situations change. A proactive agent must show opportunistic behaviour and take the initiative at the right time. Most of the studies adopted cooperative and collaborative agents in their architecture. Cooperative agents have different beliefs and reasoning methods, and they share a common goal, while collaborative agents share a common objective but keep their individual goals [41]. Few of the selected studies adopted deliberative agents in their architectures [43]–[45]. This type of agent can reason using

built-in knowledge. There is also competitive agents used by [49]. Figure 4b illustrates the classification of multi-agent IDS architectures based on agents' types. For more details on the types of agents used, see Appendix B.

3) THE COMPUTATIONAL COMPONENTS CLASSIFICATION

The computational components refer to the algorithmic techniques that the agents include to achieve attack detection tasks. In this SLR, we found four computational components in the proposed architectures, namely data collection and synchronization, data analysis, management and coordination, and knowledge sharing. First, data collection and synchronization in the current multi-agent IDS architectures is a distributed process that involves multiple sensor agents located in multiple places on the network to collect, aggregate, and prepare data for the analysis process. In the hierarchical and decentralized organizational structures, the data produced by sensor agents at lower layers travel upwards through the hierarchy to upper layers to provide a broader view about current incidents. On the other hand, in the dynamically adaptive structure, this process is quite challenging as agents change their locations and behaviours. Multi-agent data collection has not sufficiently addressed in the literature yet. However, there are few studies proposed techniques for data merging and synchronization using time interval [10], [45]–[47], IP address [48] and attack type [42], [49], [50].

Second, data analysis is the process of manipulating the collected attacks data by analyzer agents to detect incidents. The analyzer agents, can be misused [47], [51], [55], anomaly [46], [52], [53] or mixture of both (hybrid) [9], [48], [54]. The pre-process and analyze the data using technologies such as statistical methods and AI methods such as machine learning, soft computing, and biologically inspired methods. The analysis process can be done by an individual agent or collectively by a team of agents. The location of the analyzer agent can be either centralized at a specific location in the network, such as the security centre [19], [20], [55], or decentralized at several points on the network [45], [56], [57]. Fig. 5 illustrates the classification of the analysis techniques used with multi-agent IDS architectures.

The third component is the management and coordination component that used to configure, organize, and maintain the multi-agent IDS architecture. In some cases, this component is manually managed by an administrator [6], [11], [53], while, in other cases, it is self-managed [37]. In the dynamically adaptive structure, this component is responsible for all self-management tasks. Finally, the knowledge sharing component is used for communicating data and results among the agents. All the architectures' agents use knowledge sharing to inform the other agents by their actions. There are three methods of knowledge sharing: a shared knowledge-base or ontology [4], [58], a distributed knowledge-base [47], [59], [60] and message exchange scheme [14], [19], [61]. In some of the selected studies, the proposed architectures exploited mobile agents to exchange knowledge among agents [62].

From the discussion of this research question, three classifications of the existing multi-agent IDS have been found. The first classification categorized the multi-agent IDS architectures based on their organizational structure, as illustrated in figure 4a. The second classification categorized the multi-agent IDS architectures according to the agents' types, as illustrated in Figure 4b. The third classification is based on the techniques used for data analysis, as illustrated in 5. Also, detailed information is shown in Appendix A. We identify that these properties are vital features that influence all other aspects of the architectures. More specifically, these key features have impacts on task distribution, data collection and synchronization, management and coordination, and knowledge sharing. Furthermore, these key features also influence the characteristics of multi-agent IDS architectures such as learning and adaptation, communication, scalability, and reliability. In the following subsections, we will discuss in detail how aspects such as task distribution, data collection and synchronization, management and coordination, and knowledge sharing can affect the performance of multi-agent IDS architectures and their characteristics.

B. RESEARCH QUESTION RQ 2 (WHAT ASPECTS OF MULTI-AGENT IDS ARCHITECTURES INFLUENCE INTRUSION DETECTION PERFORMANCE?)

For answering this research question, we consider the previous discussion on question RQ 1. In the light of what has been discussed there, it can be concluded that there exist interrelationships among the organizational structures, the agents' types, and the computational components of multi-agent IDS architectures.

1) AGENTS AND TASKS DISTRIBUTION

Tasks distribution is the process of decomposition and distribution of problem-solving tasks among multiple agents [41]. In multi-agent IDS architectures, agents are allocated subtasks and distributed according to the chosen organizational structure, which determines how sensor agents will be placed, whether there will be individual analyzer agent or multiple analyzers, and how the architectural management processes will be dealt with. With respect to sensor agents, it was observed all the organizational structures had adopted distributed sensor agents to capture attacks (e.g., distributed and coordinated attacks) traffic from the network [63]. The distribution of multiple sensors can increase the system scalability by increasing the number of agents to collect and pre-process large volumes of data concurrently.

Second, as for analyzer agents, the multi-agent IDS architectures exhibited two ways, individual and multiple analyzers. Using multiple analyzers will help in load balancing by dividing the workload among multiple analyzer agents for parallel execution [9]. There is also another benefit of using multiple analyzer agents, which is the creation of multiple agents with different analysis techniques to analyze sophisticated types of attacks that cannot be detected by a single analysis technique. That is because, in some cases, specific

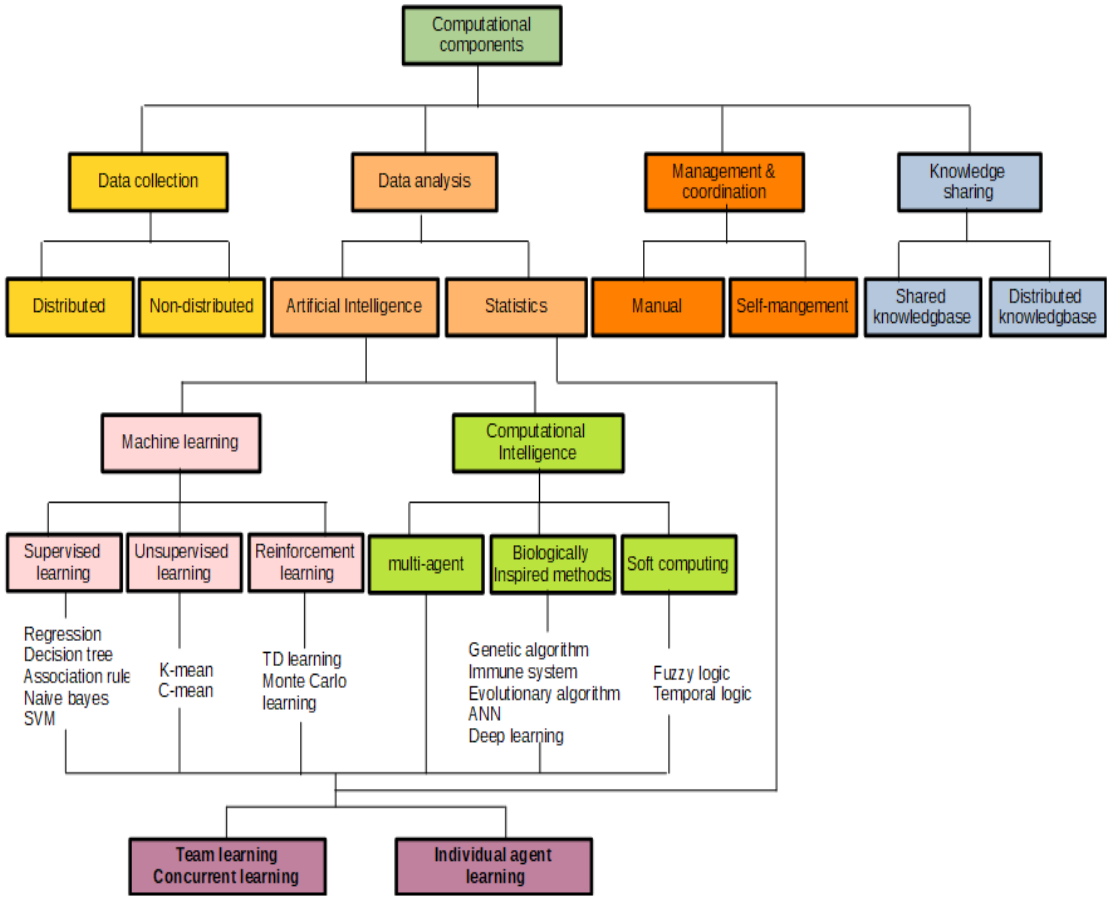


FIGURE 5. The Classification of the intelligent AI techniques used with multi-agent IDS architectures.

techniques fail in analyzing specific types of attack data, while others can analyze them effectively [4].

Third, in most of the proposed approaches, the management task of the multi-agent IDS architecture is centralized and manually achieved by an administrator who interacts with the system through a user interface (console) to accomplish the management tasks such as creating, adding, or deleting agents [11], [19].

2) DATA COLLECTION AND SYNCHRONIZATION

Based upon our studies, the existing multi-agent IDS architectures disclosed that data collection and synchronization had not been discussed sufficiently in the literature. However, there are a few studies that also included tasks such as data collection, aggregation, synchronization, and preparation for analysis by analyzer agent(s). These tasks, in some cases, were embedded in sensor agents [58], [64], while in other cases, they were added to the tasks of analysis agents [57], [65]. As apart of data collection and synchronization, sensor and analysis agents also encompassed methods to generate and derive new features, from the accumulated data, these features assumed to be effective in classifying the attack incidents [4], [5], [54].

One of the issues that challenge efficient data aggregation is to merge data from different sources. This problem has not been discussed sufficiently in the literature, though some studies contained schemes for merging and synchronizing data by source and destination IP addresses [46], [48], timestamps [10], [45]–[47], and protocol type [42], [49], [50].

The data collection and synchronization processes affect detection performance in different ways. One way is that the complexity of aggregation methods may degrade the detection performance by increasing the processing cost. Additionally, if there is a separate agent for data aggregation, this can also add additional communication overhead on the system as the agents need to communicate. Consequently, the throughput of the architecture will be reduced, and the response time would be increased in contrast. Furthermore, data aggregation methods also affect the detection accuracy by the quality level of the generated features.

3) DATA ANALYSIS

The data analysis component of multi-agent IDS architectures is the most crucial component because it carries out data processing and analysis. The agents learn while they

are analyzing data using techniques considered the core of multi-agent intelligence. The analysis techniques used with the current architectures can be divided into four categories: statistical methods, AI, soft computing, and immune system techniques, Appendix C shows the analysis methods used with the current architectures.

Our investigation disclosed that most multi-agent IDS architectures actually use single-agent learning, and there is no clear definition of team learning in multi-agent IDS architectures except in few studies that used some sort of multi-agent learning but not exactly team learning, such as [16], [66]. Multi-agent team learning is a very complicated task, but it has great benefits on the agent's rationality [22] and concurrent learning [41].

The techniques used with the data analysis component may affect intrusions and attacks detection performance in two ways. First, the processing cost of some techniques, such as ANN, is very high, and this could cause a delay in response time. The computational cost of data analysis techniques of multi-agent IDS architectures is liable to the analysis technique complexity, and also the data amount needs to be analyzed. Complex AI techniques used to consume too much CPU time and RAM space rather than simple methods such as statistics.

Second, the analysis component is also affected by the selected features and the effectiveness of the analysis technique chosen in data classification. An example of this, some features are useful in detecting some attack types; while they are not in detecting other attack types. Also, there is some classification technique that is effective in classifying some attack types; but they are less effective in other cases.

In regards to network performance, team learning agents can use distributed data analysis and provide a mechanism for load balance. In the case when massive attacks such as coordinated DDoS and worms strike a network, the data volumes can suddenly become very big for IDS to process in real-time. To solve this situation, multiple analysis agents can divide the workload among them and process the data concurrently. There are several studies that use multiple analyzer agents with multiple analysis techniques to balance the load and benefit from multiple analyzers [11], [67], [68]. The grouping of the data analysis methods and their advantages and disadvantages are explained in Appendix C.

4) MANAGEMENT AND COORDINATION

The results showed that the existing multi-agent IDS architectures achieve coordination and management tasks by using a separate agent called manager, coordinator, or moderator, such that the proposed architectures in [14], [20], [61]. The manager agent performs management tasks either manually or automatically by using a self-management mechanism. In the case of manual management, an administrator is in charge of performing all the management operations such as adding, removing, and configuring architecture agents [11], [19]. The disadvantage of this method is that the IDS architecture is completely un-configurable and un-scalable without an administrator. Therefore, the system cannot change, adapt, or extend by itself to face the environmental changes. In the automatic or self-management architecture, there is no administrator, and the IDS architecture can automatically adapt to environmental changes. This needs intelligent mechanisms to check out environmental changes and react to them by adding or removing agents. Quite a few architectures use the automatic management scheme [37].

5) KNOWLEDGE SHARING

Knowledge sharing is an essential part of multi-agent intelligence. This SLR found out that there are three different knowledge sharing schemes used with the existing multi-agent IDS architectures. First, some architectures adopted shared knowledgebase or, in some cases, shared ontologies. A shared knowledge base or ontology represents a central hub for all agents to exchange their desires and beliefs; this scheme adopted by [4], [9], [15], [58], [59]. The problem of this scheme is that the multi-agent IDS architectures become susceptible to the risk of central point of failure, central the hub may face errors and crashes. Also, this scheme adds extra communication overhead because the architectures agents contend at the shared point to communicate their pieces of knowledge. Moreover, the processing time of the communication messages will increase the overall time of data analysis.

Second, some of the existing architectures adopted distributed knowledgebase schemes. In these schemes, every agent has its own knowledge base, and it needs to synchronize it with other architecture agents [47], [59], [70]. This also adds additional communication overhead to the system architecture because agents need to communicate with each other to synchronize their knowledgebase.

Third, all the existing architectures used message exchange schemes. Such schemes are considered vital for architectures' agents to cooperate and collaborate to achieve their goals. The main disadvantage of this scheme is that there could be extreme communication overhead among the agents if they use an inefficient cooperation protocol (e.g., unconstrained interactions among agents).

From the discussion of this research question, we conclude that there are very strong interrelationships among the properties and characteristics of multi-agent IDS architectures and the performance of attack detection. The discussion focused mainly on the effects of agent distribution, data collection and aggregation, data analysis, management and coordination, and knowledge sharing. We found that the mechanisms and techniques used have direct effects on the performance of attack detection. Based on what has been discussed, we can say that the more intelligent the mechanisms and techniques used, with multi-agent IDS architectures, the faster and accurate it is attack detection. Figure 6 summarizes the interrelationships among multi-agent IDS architectures properties, characteristics, and intelligence.

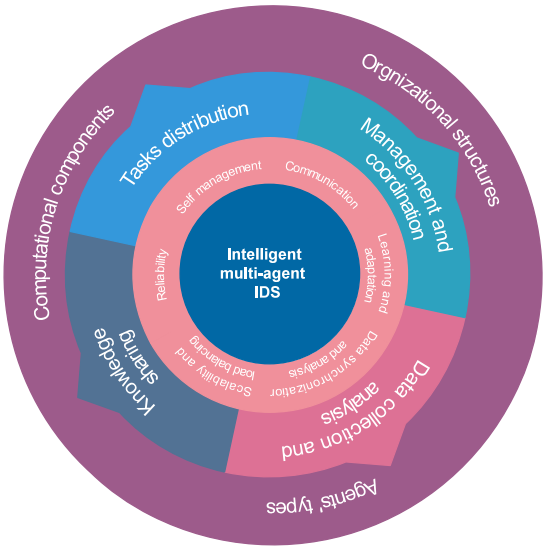


FIGURE 6. Interrelationships of multi-agent IDS architectures properties, characteristics, and intelligence.

C. RESEARCH QUESTION RQ 3 (WHAT LIMITATIONS EXIST IN MULTI-AGENT IDS THAT INFLUENCE INTRUSIONS AND ATTACKS DETECTION?)

The discussion of question RQ 2 emphasizes that the properties and characteristics of multi-agent IDS architectures influence the overall performance of intrusions and attacks detection. The discussion also confirms that although multi-agent IDS architectures have advantageous characteristics, but they also suffer limitations that can eventually degrade the overall performance of intrusions and attacks detection. In the following subsections, we will make detailed discussions on the limitations found in the existing multi-agent IDS architectures. The discussion will focus mainly on the limitations that may impact the response time and accuracy of intrusions and attacks detection.

1) SCALABILITY

Scalability means the ability of an application to grow up to meet the increasing performance demands [71]. In other words, it means the ability to incrementally add agents to a system for processing volumes of data [41]. In this SLR, the scalability of multi-agent IDS architectures will be discussed from an architectural perspective where the system automatically scales up by instantiating agents (e.g., sensors and analyzers) as needed. On the other side, as the number of agents grows, the state space will grow as well.

Multi-agent systems are inherently scalable; this is so because of the innate modularity they are supposed to have. So, the techniques used for handling this modularity are of importance [39]. Based on the obtained results, the scalability of multi-agent IDS architectures was not sufficiently addressed. However, there are some architectures that provide untested mechanisms and techniques for scalability. All three organizational structures (hierarchical, decentralized, and dynamically adaptive) suffer from limitations in scalability.

In the hierarchical structure, there are few architectures that have mechanisms to provide scalable IDS. In [20], the proposed architecture included a method that creates agents with specific tasks and distributes them across the layers of the hierarchy to compose the IDS. In this study, the degree of scalability is still not sufficient as the new agents are located in the same host and share the same system resources. The proposed architectures in the studies [9], [11], [72], [74], provide mechanisms for scalability by instantiating new agents when needed. These architectures didn't enforce any constraint on agents' creation. Thus, creating agents without restrictions will exhaust computer system resources and degrades the performance of intrusions detection.

The decentralized structure includes only two study that deals with scalability. In [48], the analysis agent used to analyze data and replicate itself as needed. The drawback of this algorithm is that this mechanism dealt with scalability in the same host and didn't deal with the network factor. Additionally, it does not consider the limitation of host resources. The proposed architecture in [14] manage scalability by dividing the network traffic to subparts and process these subparts simultaneously using a group of agents distributed over the network. In [66],task decomposition and team learning were used, that instead of having one big DDoS attack problem at the victim machine, there would be multiple smaller DDoS at the team leaders.

In the dynamically adaptive structure, there are also few studies dealt with scalability. The proposed architectures in [35], [61] used simple methods for adding and removing agents. These methods didn't enforce any constraint on agents' creation and deletion. In [62], a technique to transfer data to the neighbouring nodes using mobile agents was used. This technique adds additional communication overhead to the system because the mobile agent repeatedly travels among hosts carrying data. In addition to that, mobile agents have security issues reported in [75]. In [59], the proposed architecture was designed to scale up dynamically by creating a random number of sensor and analyzer agents to gather and analyze data related to attacks. The random creation of agents exhausts system resources. For this reason, there should be limits and constraints for creating agents. In [60], [76], [102], the proposed architectures use mobile agents to search for the most effective nodes to analyze data, but again, the use of a mobile agent will add communication overhead.

The proposed architecture in [54] used a mechanism that selects the best analysis agents, in terms of their clustering capabilities, to replicate them for analyzing data. The disadvantage of this mechanism is that it does not enforce strict constraints on agents' replication. The architecture should be expanded only when exposed to massive attacks, such as worms and DDoS; otherwise, it will be a consumption of system resources. Agents' creation algorithms should take into consideration system and network resources constraints.

The architectures proposed in [60], [76], scale up by moving agents to the nodes with useful classification capabilities.

The disadvantages of this approach are the use of mobile agents that suffer from security issues reported in [75].

From this discussion, we conclude that the scalability of the multi-agent IDS architectures suffers limitations in the methods that have been proposed so far. Therefore, it is necessary to develop and evaluate models, frameworks, and approaches to provide scalability as an essential characteristic in the architectures.

2) LOAD BALANCING

Load balancing is the distribution of workload among multiple agents to optimize resource usage, maximize throughput, minimize response time, or to avoid overload on a single resource [77]. This characteristic is interrelated to scalability; a scalable system is a load balancing necessarily. That is because when tasks are divided and distributed among architecture agents, the workload also has to be divided. The obtained results reveal that the load balancing of multi-agent IDS architectures was not sufficiently addressed. But, there are some architectures that provide untested mechanisms and techniques to handle situations when IDSs congest and need for loads balancing. Based on the selected studies, all three organizational structures suffer from load balancing limitations.

In the hierarchical structure, there are some architectures used mechanisms for load balancing. In [20], the proposed architecture encompassed an algorithm for dividing the workload among various agents and avoid the centralization of traffic. The algorithm also can change agents' roles depending on the current network status. There are no details on how the architecture scales up to accommodate data when attacks are launched. For instance, when massive attacks, such as DDoS and worms, occur in a network, the algorithm doesn't show how the workload is divided and distributed among agents to process the huge attack data. The method should take into account the amount of data expected to be processed so that the IDS estimate the required number of analysis agents. In [63], the proposed architecture included a cooperation mechanism that distributes the workloads among agents in the neighbouring hosts. The architecture included a global intrusion detection agent to communicate with its neighbouring agents to detect attacks or to use storage. In this architecture, the total number of agents remains the same, but the load is distributed among the neighbouring agents. The issue in this mechanism is that it does not handle the situations when all agents are loaded. In [7], the proposed architecture used an algorithm for distributing the detectors (attack data) across hosts to improve the processing time. This method distributes the workloads arbitrarily across the network without constraints to choose which host is the best choice for processing the data. Because if a host is already loaded, adding more jobs will decrease throughput. In [78], the proposed architecture provides minimum network load and better CPU utilization by shadowing log file. Shadowing log file is considered as work duplicated, and this also increases the processing time. The proposed architectures in the studies [9], [11], [72], [74] included mechanisms for distributing the classification tasks throughout the various layers of the architectures. The mechanisms dealt with distributing the tasks but didn't handle how a workload is divided among the architecture agents.

In the decentralized structure, there are a few studies that used mechanisms for load balancing. In [101], A method to control the CPU usage ratio of the IDS server with both normal traffic and flooding attacks was used. The method works by managing registration requests of the clients to keep low CPU load on the IDS server. The proposed architecture in [79] used a mechanism for applying a filtering operation that matches between the captured traffic and the intrusion database to ensure that only the unclassified traffic will be processed.

In the dynamically adaptive structure, also there are a few studies used mechanisms for load balancing. In [14], [61], the proposed architectures balance the load by creating multiple analysis agents, based on the network traffic and the availability of logical processors. Then, the architecture divides the dataset into sub-datasets. This method does not enforce any rules on the agents' creation. An algorithm based on a dynamic election idea, to search and choose agents with fewer loads for analyzing attacks' data, was proposed in [48]. The election algorithm limits search operations (for agents) to be in the same network segment and didn't take into account the other segments. The issue is that if the same network segment is loaded, there should be a chance to move the analysis processes to other segments with less traffic. Another election based mechanism was proposed by [5]. The mechanism was used for searching and choosing the agent with the lightest load to replace the current agent, which is having a higher load due to an attack. The algorithm searches the whole agents' list, using a binary search or group search until it finds a management agent with the lowest load. Although this algorithm provides load balancing, when attacks strike, such as DDoS or Worm attack, the whole network is loaded, not only the analysis agents. For this reason, the constraints should include other factors such as host and network states.

The previous discussion shows that the load balancing mechanisms used with multi-agent IDS architectures suffer limitations. So, it is necessary to conduct experiments and propose models, frameworks, and metrics to create and evaluate this characteristic on multi-agent IDS architectures.

3) LIMITATIONS OF RELIABILITY, FAULT TOLERANCE AND AVAILABILITY

Reliability, fault tolerance, and availability are interleaved characteristics defined, in an article published by Microsoft [71], as follows: 1) the reliability is the probability of failure for a single solution component. The system is reliable if it is fault-tolerant; otherwise, it might face unavailability. 2) fault tolerance is the possibility of a system architecture to continue functioning when parts of it face failure. 3) availability means the percentage of time that a system can achieve its intended functions.

Based on the obtained results, most of the hierarchical architectures do not have these essential characteristics because there is always the issue of a single point of failure. However, some of the existing hierarchical architectures used some mechanisms to recover from errors and failures. For making the system reliable, an auto fault tolerance characteristic was developed by [78] using mobile agents. In [9], the proposed architecture comprised a mechanism to create new agents to recover from errors and problems. This study doesn't provide any details on the performance of the mechanism. In [63], the proposed architecture used a reliability technique to make each of the architecture's components monitored by a different component. This method is not a useful solution for providing a reliable operation on the architecture. The communication overhead among the components can cause high traffic on the network, which can lead to undesirable results such as delay in response. In both [7], [73], the robustness of the proposed architectures comes from integrating several detecter agents with several classification algorithms and manage them to detect attack incidents.

There are few studies related to the decentralized and distributed structure that provides these characteristics. In [44], [45], the architectures used an algorithm based on Byzantine Agreement Protocol (BAP) to detect and isolate the compromised node. This is still not enough for the architecture reliability as the mechanism doesn't include a recovery technique to restore the system architectures. In [79], the proposed system used an algorithm that stores each piece of data in three locations to promote fault tolerance and recovery.

The dynamically adaptive structure has a few studies showing these characteristics. The architecture in [5] used a dynamic selection algorithm based on an improved Bully algorithm for timely system recovery when an error occurs due to the higher load at the management agent. The algorithm focused on the failures due to the higher load and didn't address the problems when the agent is dead due to network and resource congestion. In [59], the architectures are fault-tolerant by using self-diagnosis, self-healing, and self-testing mechanisms to detect abnormal behaviour, recover and repair the damage. A robust communications protocol for multi-agent IDS was proposed to handle transmission losses [60], [76]. This protocol didn't handle agents' failures or system crashes.

From the discussion, the existing multi-agent IDS architectures suffer limitations in reliability, fault tolerance, and availability. Therefore, it is necessary to propose and evaluate robust and fault tolerance methods on multi-agent IDS architectures.

4) ADAPTATION AND LEARNING

Adaptation means the ability of an interactive system to change its behaviour according to environment changes and information acquired from the surrounding agents [80], [81]. The term learning refers to the process that includes all computations such as knowledge sharing, knowledge acquisition, classification, prediction, inference, and decision making that are executed to achieve a particular learning goal [82]. Adaptation, learning, self-organization, and self-configuration are interconnected characteristics. These characteristics are significant for what is called self-management [83]. Self-organization is the ability of a system to automatically arrange its components and elements in a suitable way without any external help [69]. Self-configuration is the ability of a system to automatically control changes in its parameters to produce the desired output [84].

The results reveal that the adaptation of multi-agent IDS architectures is twofold, adjusting agents' states to respond to new attacks or changing system parameters(e.g., traffic, agents number) to respond to environmental changes. Adjusting agents' internal states is the connection point between adaptation and learning. With respect to this, the results uncovered that there are several adaptation mechanisms used for changing agents' internal states to enable them to detect newly seen attacks. Generally, these mechanisms rely on continuously updating agents' knowledgebase with the help of machine learning, soft computing, and other AI techniques. The suggested architecture in [6], used a simple mechanism, based on association rules, that changes the profiles of the typical behaviours to enable the analysis agents to detect the new abnormal behaviour. The main issue of this mechanism is that creating rules needs prior knowledge, which will not be available most of the time. Another issue of interest is the use of single-agent learning and not using cooperative learning; when a network experiences a DDoS attack, for example, agents' cooperation is required to collect and jointly process the attack data for fast and effective detection.

The proposed architecture in [46], a rule-based algorithm using information theory, was suggested to detect when abnormal behaviour is detected; the system enters a self-diagnosis mode to categorize the fault and get detailed information about the incident such as source IP address and symptoms on the system. Once these details are identified, the system adapts by generating features to be tested to find the best ones for building classification rules. This mechanism uses labelled data, which will not be available in all cases.

The adaptive architecture proposed in [45] uses a Bayesian-based learning algorithm to enable the agents to detect new attack types. In addition to the prior knowledge required by the Bayesian algorithm, also the manual confirmation by the system administrator is needed, which is impractical.

The proposed architecture in [54], uses an evolutionary algorithm in association with a Naïve Bayesian classifier to estimates the probability of intrusions' activities. The algorithm used genetic algorithm to let agents continuously learn from the environment. Also, in [85], an evolutionary algorithm was used in combination with game theory. The computational cost of evolutionary algorithms is very high esspecialy in processing huge data. The same problem exists in [86], [94] where a bunch of classification algorithms used

for evaluating the performance of multi-agent architecture. In [87], a knowledge-base with a reasoning algorithm is used, but still, the problem of learning needs to be fixed, because the prior knowledge is not available in the case of new attacks.

The proposed architecture in [67], uses a technique composed of a series of machine learning techniques include Online Random Tree, Online Random Forest, Online LaRank, Online Multi-Class Linear Programming Boost, and Multi-Class Gradient Boost. These techniques fused with multi-agent to detect malicious behaviours. The main issue in this approach is the blending of human expertise with modern artificial intelligence. Human expertise is difficult to extract and also takes time to be translated into rules.

The architectures in [7], [64], use mechanisms inspired by the immune system and genetic algorithm to generate a knowledge base of attacks to immediately respond to the attacks in the future. The algorithms have two issues: limitations in learning and high computational cost. The learning limitation is that the algorithms need to learn from previously labelled data, which will not be available in the shadow of the very rapid growth of Cybersecurity attacks. The high computational cost of these mechanisms is caused when matching the suspicious network connections with the whole self and non-self-detectors lists. This causes computational overhead that consumes system resources (CPU time and RAM) and eventually will cause a delay in response time. Also, Genetic algorithms do not scale well with complexity [88].

In [52], the proposed architecture contains an immune system based algorithm that selects the best analyzer agents to randomly replicate them to classify attacks, based on a fitness function with parameters such as number of agents, number of classification records etc. The issue of this algorithm is that the fitness function does not consider any of the environmental factors such as system resources or network resources. The mechanism might fail in case the specified agent resides in a very busy host or network segment. On the other hand, this mechanism clone agent randomly, which is considered as resource consumption.

The architectures in [59], [70] include learning and adaptation mechanisms based on the immune system to enable agents to detect new abnormal activities. The problem of this mechanism is that the agents need labelled data for training, which is not always available. Another issue in this mechanism is that the adaptation mechanisms did not take into consideration how the architecture agents will adapt to the environmental changes. For example, when a network is exposed to high traffic because of an attack, the architecture agents could immediately adapt by increasing the number of analyzer agents, choosing analysis techniques, or isolating the suspicious nodes to reduce the attack invasion. The suggested mechanism doesn't show how the agents get involved in cooperative learning.

The architectures proposed in [42], [89], [90] use an adaptation mechanism inspired by the immune system to adjust the architectures' agents according to environmental changes. The mechanisms adapt in three stages: the first stage is called diversity generation, which is the generation of different agents with distinct specificity by using mutations. The second stage is called self-maintenance, where the agents are adjusted to be insensitive to known attacks during the development phase. Third, is the phase called the memory of non-self, where agents are adjusted to be more sensitive to unknown attacks. The issue in this mechanism is the excessive creation of the agents that will consume system resources. On the other hand, making the agents sensitive and non-sensitive is a recurrent process based on calculating the danger degree using mathematics, and this will also cause high computational cost. The danger theory algorithm of the immune system also utilzed by [99].

The architecture proposed in [48] includes an adaptation mechanism based on an improved Artificial Neural Network(ANN) algorithm that adjusts the learning rates adaptively. The performance of ANN is exceptional when a huge number of training data is used. However, one of the ANN problems is called over-fitting, which occurs when the network stores all training examples, but it is unable to generalize to new cases. From another side, the huge data volumes cause high processing cost.

The proposed architecture in [91], uses an adaptive adjustment sub-system, based on ANN, to automatically adjust the system to detect new attacks, by using the information provided by the environment, or manually by an administrator. The problem of this mechanism is that it requires manual intervention from an administrator.

The proposed architecture in [5], uses a mixture of the anomaly and misuse techniques. The anomaly technique implemented to improve the misuse detection applying correlation analysis, sequence analysis, cluster analysis, classification analysis, rough set analysis, and outlier analysis to update rules library and eliminate the old rules set.

In [38], the proposed architecture uses a Qualia based principle that takes the analysis results to modify world models to use in the future. The architecture also uses prior knowledge and agents' own experience in making decisions.

The architectures proposed in [16]–[18], [92], [105] encompass reinforcement learning algorithms with fuzzy logic (as a function approximation) to adapt by selecting the best strategy for detecting attacks and responding to it. The main issue in these architectures, as stated by the researchers, is that convergence may not occur, and that means the optimal solution is not guaranteed. The same problem persists in the reinforcement learning mechanisims proposed in [66], [103], [104].

A trust based adaptation technique integrated with a distributed agent-based architecture for detecting DDoS attacks in WLAN was proposed in [93]. The technique was used for detecting and isolating the attacks. A trust mechanism is a rating process between two peers based on their historical performance. The problem is that if no historical, the agents will not be able to detect the attack.

In the proposed architecture in [38], the agents use a learning model that extracts new information from the surrounding

entities and environment to adapt itself to new threats with little or no human intervention.

In [15], a self-learning ontology was proposed using Intuitionistic Fuzzy Logic (IFL) to generate new attack rules. The problem in this mechanism is the need for labeled data, which will not be available in the case of the swift evolution of cyberattacks.

The suggested architecture in [62], comprises an algorithm based on Support Vector Machine (SVM) for learning and prediction of new attack types. The algorithm trains the analyzer agent directly by using a dataset to create a model and then uses that model for classifying the new attack types. The problem of this algorithm is that it doesn't deal with how the architecture agents could learn from each other or from the environment. SVM was combined with extreme machine learning (ELM) technique and used by the adaptive architecture proposed in [35].

With regards to the adaptation mechanisms that were used with multi-agent IDS architectures to change agents' behaviours to respond to environmental changes, these mechanisms involve tasks such as adding, removing, or changing agents' goals. The proposed system architecture in [62], adapts to the environment by using a mechanism based on mobile agents. There are critical issues for mobile agents mentioned in [75]. Thus, mobile agents are considered a drawback for this mechanism.

The adaptive architecture in [53] comprises a mechanism that dynamically adapts to environmental changes and attacks. Based on a condition, intrusion detection will be achieved by the basic agent, local coordination agent, or global coordination agent. The disadvantage of this process is the long steps to follow in case an event could be detected neither by the basic agent nor by the local coordinator agent.

In [54], the proposed architecture comprises a mechanism based on a genetic algorithm for adding and removing clustering agents according to their fitness. The fitness is a value calculated to determine an agent's ability to cluster data. The clustering agent that produces clusters with high dispersion is considered less effective than the clustering agent that produces clusters with low dispersion of elements. Therefore, the later is replicated, and the former is removed. For replicating the clustering agents, the algorithm considers only the internal agent's state (clustering ability) and ignores the other factors such as system and network status. Suppose the best agents reside in a very busy area in the network, using this adaptation algorithm will increase the response time due to the increased processing cost.

In [76], [102], the proposed architectures include a reputation based algorithm to dynamically find nodes with a high ability to classify network activities, and a multi-objective evolutionary algorithm is enforced to help agents search for useful operational parameter values for classification. The reputation algorithm is used for instructing agents to migrate to other nodes or to share information with other agents. One of the disadvantages of this approach is the security issue related to the mobile agent mentioned in [75]. On the other hand, the central agent controller is liable to the risk of the central point of failure.

The suggested architecture in [67] includes an adaptation mechanism that has two detection engines, misuse, and machine learning. When a malicious multi-agent system changes its behaviour to evade detection, the misuse engine gradually stops warning of the current malicious traffic, and the machine learning agents continue to investigate and alert of the new behaviour. The architecture also uses an election algorithm to choose between multiple machine learning techniques to analyze network traffic. The problem in this architecture it that the supervised machine learning techniques need labelled data which will not be available most of the time in the shadow of the very fast evolution of attacks.

In [5], when a network experiences heavy load because of an attack, one of the management agents initiates an election process by communicating with the other management agents in the network to examine their loads. For each management agent, if its load is less than the load of the management agent that has initiated the election process, the agent will reply by a positive election result. Then, the initiator agent selects the agent with the smallest load to start the analysis. If the initiator agent receives no result, then it will replicate itself and start the analysis process. Although the empirical evaluation of this study was not presented, it can be noticed that the proposed mechanism considered only the load of the agents, and didn't consider the other factors that might affect the detection performance, such as the availability of system resources. In the architecture proposed by [95], an adaptation scheme depends on attack severity was used. This scheme utilized a paramater called relationship metric that characterizes the distribution of clients. A too high value of this metric is considered as abnormality.

The proposed architecture in [19] includes an adaptation mechanism inspired by the biological immune system to automatically create two types of agents: an intermediary agent with a memory of detectors (attack data records) and a superior agent, which is a mobile agent. The intermediary agent is used for identifying attacks, and the superior agent is used for moving across the network to perform autodestructive processes. The use of mobile agent adds additional load on the network because it travels from host to host to destruct attacks and their consequences. Furthermore, the sequential movement of the mobile agent, from host to host to undo the damage caused by attacks, can cause a delay in response time.

In [9], [11], [43], [72]–[74], the architectures use mechanisms that consist of two types of intelligent Case-Based Reasoning(CBR) agents used to learn and adapt to changes in attack patterns and user behaviour. The adaptation mechanisms divide the classification task into two phases. In the first phase, a process called initial filter is implemented to detect simple attacks without using a large number of system resources. In the second phase, more complex computations are performed, and that requires a large amount of computer system resources. The mechanisms can adapt by enforcing a

load balance mechanism to save system resources. A mechanism based on CBR agents was also proposed by [20] to reuse past knowledge to solve new problems.

With respect to the relation between adaptation and self-management, the suggested self-managed architecture in [37] includes a mechanism to dynamically adapt to environmental changes, tune resources, discover, diagnose, react to disruptions, and anticipate detection, identification, and protection against threats. This mechanism uses an autonomous central agent that can diagnose failures, and manage the situations when higher loads and communication overhead arise in the system. There are no enough details on the self-management mechanism; however, the autonomous central management agent is liable to failures.

From the previous discussion, the existing multi-agent IDS architectures have limitations in the adaptation mechanisms, and it is required to conduct experiments on more enhanced adaptation models and frameworks. Also, there is a need for proposing standard measurements and metrics for evaluating the adaptation of multi-agent IDS architectures.

5) MULTI-AGENT LEARNING

Learning is an essential part of multi-agent intelligence. It comprises two types: Single Agent Learning(SAL), which means how an individual agent improves its learning abilities, and Multi-Agent Learning (MAL), which means how a group of agents cooperate in analyzing data and learning effectively in a multi-agent environment [82]. Learning techniques are always embedded in the data analysis components of multi-agent IDS architectures.

Our investigations on the existing multi-agent IDS architectures manifest that true MAL has not been achieved yet. There are several important aspects of multi-agent learning that have not been covered until now in the literature of multi-agent IDS, for instance, distributed AI, parallelism, interactions, and learning methods. So, in this subsection, we will limit the discussion on the few multi-agent IDS architectures that exhibiting some characteristics of MAL.

The multi-agent IDS architectures that adopt the approach of multi-agent learning; they use different mechanisms to let the agents communicate with each other to improve their knowledge. The proposed hierarchical architectures in [16], [18], adjust their learning parameters through fuzzy Q-learning to detect future attacks. The architectures' agents cooperatively learn to adjust their parameters a mechanism based on game theory. The proposed game theory approach was limited to only two agents, so the true MAL doesn't exist in this architecture.

In the proposed architecture by [17], a cooperative fuzzy artificial immune system mechanism was proposed to improve the agents' self-learning capacities and provide the agents with an incentive function to protect the most vulnerable sensor nodes. There are two issues in the mentioned architectures: first, using Fuzzy logic need human effort for designing fuzzy rules, which is not practical while there exists a huge number of new attack every day. Second, the cooperation using the hierarchical structure increases the communication overhead as the number of the hierarchy layers' increases.

From the previous discussion, the learning of the multi-agent IDS architectures has two aspects: 1) individual agent learning, 2) cooperative multi-agent learning. From the selected studies, the learning mechanisms that used with the current multi-agent IDS architectures have limitations that can be concluded in: incremental agent learning, agents learning from the environment, and cooperative multi-agent learning. The proposed approaches also ignored how agents infer knowledge by their own(reasoning). For this reason, the multi-agent IDS architectures need more improved frameworks, models, and algorithms for enhancing learning capacities.

6) COMMUNICATION OVERHEAD

In a multi-agent system, the communications related to negotiations or the transfer of high volumes of information causes significant overhead that leads to delay in the systems with strict time and bandwidth limits [96]. Real-time applications such as IDSs should strictly enforce rules on agents' communications to prevent system bottlenecks and delays that lead to low throughput [41]. Multi-agent IDS architectures typically fall into this type of system, because they need to transfer huge data in real-time. From the selected studies, it was observed that communication overhead issues had not been addressed in the existing multi-agent IDS architectures, but there are only a few studies that deal with this issue.

The proposed architecture in [106] uses a mechanism to keep the number of messages constant for each time interval. Keeping the number of messages constant will delay the communications among all the architecture agents and reduce the system throughput, which is not suitable with a real-time IDS.

Another solution proposed by [67], was to keep the message size small, only 20 bytes. In this technique, the system needs to break the data of the network traffic into smaller packets and send them in multiple rounds. This will cause additional overhead due to a large number of messages, especially when DDoS and Worm attacks launched in the network.

A mechanism used by the architecture proposed in [97], divides the network into segments to allow anti-worm mobile agents spread to clean the infected machines.If there is no worm detected in certain segments for a certain time, the anti-worm mobile agents will stop spreading in those segments. This mechanism reduces the overhead system cost in certain cases, but will fail when worm invasions acyually happen, the anti-worm mobile agents spread to clean the infected machines, which will make the matter worst in the infected segment due to the communication overhead among the architecture's agents.

The hierarchical architectures suggested in [19] consists of multiple layers that cause long communication cycles among the architecture agents. The more layers the hierarchy has, the more communication overhead will happen.

From this discussion, there are approximately no mechanisms to control the vast communications in the existing multi-agent IDS architectures. The few studies discussed here provide limited solutions to reduce the message size and number during agent communications. We argue that these solutions, also, cannot scale up to vast communications, especially when massive attacks, such as DDoS, strike. Therefore, it is necessary to develop protocols to control the communications in multi-agent IDS architectures to improve the performance of attack detection. Also, it is critical to update the measurement methods for evaluating communications in IDS.

D. RESEARCH QUESTION RQ 4 (WHAT METRICS ARE USED TO MEASURE AND EVALUATE MULTI-AGENT IDS ARCHITECTURES?)

Based on [98], the metrics and measurements used for evaluating the characteristics of real-time distributed IDS systems can be divided into three categories: logistical, architectural, and performance metrics. The logistical metrics are used for measuring characteristics such as manageability, configurability, maintainability, and other platform requirements. The architectural metrics are used in measuring characteristics such as scalability, load-balancing, system throughput, learning, adjustable sensitivity, robustness, and supportability of multi-sensor and multi-analyzer. The performance metrics are used for measuring how well the system is performing, such that analysis capacities, false and accurate detection, recall, accuracy, response time, and traffic latency. Some of these metrics are quantitative, and some are descriptive. The results show that the logistical characteristics of multi-agent IDS architectures were not sufficiently evaluated in the literature, except in [5], [37]. In these architectures, the proposed systems reported they have good anti-destroy, self-restore, and self-configuration abilities.

As previously discussed, there are interrelationships among the characteristics of multi-agent IDS architectures. Therefore, in addition to the performance metrics derived from the confusion matrix such as FPR(False Positive Rate), FNR(False Negative Rate), accuracy, and detection rate. In most of the studies, the architectural and performance characteristics were correlated and evaluated using combined metrics, for measuring the effects of multi-agent features such as coordinated team learning, adaptation, and scalability, on intrusions and attack detection. For instance, the metrics such as network latency, bandwidth consumption, number of data packets per second, and detection rates concerning the number of instances and data packets were used for evaluating the performance of the scalable multi-agent IDS architecture proposed in [6].

To evaluate learning, scalability, and adaptation of the distributed hierarchical architecture proposed for detecting SQL injection attacks in [9], processing time, response time, similarity measure, detection rate, FPR, and FNR were used. The same metrics were also utilized to evaluate the same characteristics in the proposed architectures in [11], [73], [74]. Additionally, a metric named error related to the number of cases that were also used for assessing the fault tolerance of these proposed architectures. The same authors used a similar version of these studies in [72], but for evaluating the system on DoS threats in web services.

For measuring the performance of the adaptive immune multi-agent IDSs in [42], [89], [90], [99], the proposed architectures were evaluated by using comparisons of a value called Mature Context Antigen Value(MCAV), used for measuring danger value, concerning the number of hosts those used to calculate it, bandwidth saturation, network connections, memory loading, and CPU usage. Also, the self-adaptive immune multi-agent IDS architectures in [7], [64] were evaluated by measuring the number of the generated memory cells(attacks data) in ten rounds and the detection rates of all hosts. The bandwidth allocation over time, attack's spread rate and network status during known and unknown attacks, and convergence were used for evaluating the multi-agent-based architectures inspired by the human immune system for detecting client's misbehaviour [56], [68]. The TPR and FPR were used to evaluate the evolutionary multi-agent approach to anomaly detection and cyber defence [54].

The performance of the adaptive intelligent qualia-based IDS in [38], evaluated using detection accuracy, false detection, precision, and recall. Another measurement of adaptation impacts, on the performance of multi-agent architecture for DoS, was the effectiveness related to the numbers of patterns [43].

Collaborative multi-agent IDS architectures were evaluated using different metrics such as precision that used with the collaborative distributed multi-agent IDS in SCADA (Supervisory Control and Data Acquisition) [10]. Also, the collaborative multi-agent IDS architecture for detecting DDoS, [106], assessed using detection rate, FPR and FNR rates concerning varied numbers of agents and gateways, and the number of times collaborative agents need to communicate with each other in different sized networks. The proposed multi-agent-based architecture in [46] that used coordination and interaction between agents for network audit and attack detection used two metrics, detection time and entropy values of different properties of UDP flooding attacks were used for evaluating.

The cooperative multi-agent architecture for detecting worms [8], is evaluated by using the relationship between the probability of worms to discover new vulnerable nodes, and the percentage of worms payloads with respect to the total payloads sent. Another metric that also used was the percentages of the infected nodes. In [49], [100], the attack traffic, network traffic before and after a filtering process, and botnet propagation were used for evaluating the cooperative multi-agent-based systems against botnets. Also, agent learning rates, botnet presence degree in computer systems, and training errors were used to evaluate the cooperative multi-agent system of botnets in [50].

In [16]–[18], metrics such as attack intensity per packet size, energy consumption over time, attacks detection rates

to the percentage of attacks, successful detection with respect to the percentage of malicious nodes, number of alive nodes over time, consumed energy over time, the total energy consumption of nodes were used to evaluate the learning of cooperative intelligent agents in detecting and preventing intrusions. Detection rate with respect to the numbers of agents and gateways was also used to evaluate the collaborative architecture in [106].

The adaptive and cooperative multi-agent architecture for botnet detection in [95] used metrics called relationship, response, and synchronization. The relationship metric characterizes the distribution of clients. A too high value of this metric is considered as abnormal. The response metric is the difference between broadcasting requests and receiving responses. The synchronization metric characterizes the synchronicity in the behaviour of clients. The multi-agent distributed information security system that characterized by collaboration and adaptation was evaluated by using metrics such as threat levels of attacks [91]. False responses and non-response rates were used for evaluating the learning, collaboration, and adaptation capacities of the distributed multi-agent intrusion detection architecture by [48].

The study in [65], used suitability value (a value produced by fuzzy logic), botnet presence degree, detection rate, and FPR to evaluate the intelligent multi-agent based approach for botnet detection by using fuzzy logic.

In the proposed architecture in [67], system latency, accuracy with respect to the percent samples tested were used for comparing a bunch of machine learning algorithms integrated with a distributed multi-agent IDS to defend multi-agent malicious behaviours. Also, the multi-agent-based architectures for unusual network behaviour detection that integrated with several anomaly detection techniques were evaluated using detection rate, clusters' number, accuracy, FPR [4], [70].

To evaluate the robust and fault-tolerant distributed intrusions detection system by [44], RAM, and CPU usage with regard to users' numbers, detection rate, FPR, and FNR were used. Also, false and negative detection were used For evaluating the proposed architecture in [45], [62].

The multi-agent system for attack classification based on a reputation algorithm was evaluated by using classification accuracy using and without using reputation [60], [76].

As discussed, multi-agent IDS architectures evaluated using logistical, architectural, and performance metrics. The logistical metrics were very rare, while most of the proposed architectures evaluated using combinations of architectural and performance metrics. From this discussion, we can notice the absence of measurements related specifically to multi-agent, such as the metrics used for evaluating team learning. Therefore, it is necessary to implement these measurements and metrics to evaluate multi-agent IDS architecture. Also, proposing new methods and guidelines for using the current metrics to evaluate these architectures.

V. VALIDITY THREATS

The selected studies investigated in this SLR were retrieved using keywords and terms related to multi-agent IDS architectures. Then the retrieved studies were filtered out manually using selection criteria. There may be some risks the selected studies do not reflect the actual state of the art of multi-agent IDS architectures. First, during the retrieval and selection of the studies, some papers possibly were missed out due to the incompatibility of the keywords and terms used in some publications. Second, after retrieving the studies, there may be new publications in the online databases that were supposed to be included for answering the research questions, but they were not. Third, the citations of the listed studies may vary from the actual status of the materials due to the changes in citation numbers everyday. However, recurred search and checking were repeatedly conducted after the retrieval of the studies to see whether there are new studies published or any citations status change.

VI. CONCLUSION AND FUTURE WORK

The purpose of this research was to investigate the existing multi-agent IDS architectures to identify the most challenging limitations that impact intrusions detection performance. In order to achieve that, this research used the protocol described by [34] to conduct SLR in software engineering. The specified protocol was utilized to design the research plan of this study, including the formulation of the research questions. The plan was executed to retrieve, assess, and filter out studies to select the most relevant ones for answering the research questions. The first and second research questions were to identify and categorize the components of the existing multi-agent IDS architectures, and the characteristics that affect the performance of intrusions and attacks detection. Then, the third research question highlighted the issues of the multi-agent IDS architectures. Finally, the fourth research question was to find out the metrics used for evaluating the current architecture to see if lacks of measurements exist. The review was carried out, and the data of the primary studies were selected, assessed, and synthesized. Then, the results were discussed in the context of the formulated research questions. The objectives of this study have been achieved by answering the research questions and identifying the issues in the existing multi-agent IDS architectures.

To conclude this SLR, as discussed, the results of this SLR emphasize that multi-agent IDS architectures have several advantageous characteristics that can help to develop performant IDSs. It is also discovered that there are several issues in multi-agent IDS, exhibited by the selected studies, that can degrade the performance of intrusions and attacks detection. The techniques, mechanisms, and schemes used to deal with multi-agent IDS scalability, adaptation and learning, load balancing, fault-tolerance, and self-management suffer issues discussed previously in this article. For example, most of the multi-agent architectures use supervised learning based on individual agent which is completely impractical to cope with the very rapid growth of network intrusions and attacks.

TABLE 3. Multi-Agent IDS Architectural Properties and Characteristics

#	Ref.	Sensor agents	Data analysis agents	Decision making	characteristics
			Hierarchical structure (27 studies)		
1	[4]	Distributed	Distributed	Centralized	Cooperation and collaboration
2	[6]	Distributed	Centralized.	Centralized.	Adaptation, coordination and mobility
3	[7]	Distributed	Centralized/ Distributed	Centralized/ Distributed	Cooperation, scalability, adaptation and robustness
4	[9], [11], [72], [94]	Distributed	Distributed	Centralized	Scalability, load balancing, fault tolerance, and reasoning
5	[15]	Distributed	Centralized	Centralized	Self-learning and adaptation
6	[17], [18]	Distributed	Decentralized	Centralized	Distribution, Cooperation, adaptation and learning
7	[19]	Distributed	Centralized	Centralized	Lightweight, adaptation, dynamics
8	[20]	Distributed	Centralized.	Centralized.	Learning and reasoning
9	[38]	Distributed	Centralized	Centralized	Learning and scalability
10	[42], [89], [90], [100]	Distributed	Distributed	Centralized	Coordination and learning
11	[43]	Distributed	Distributed	Centralized	Adaptation, learning and reasoning
12	[46]	Distributed	Centralized	Centralized	Cooperation, coordination, robustness
13	[47]	Distributed	Centralized	Centralized	Cooperation, scalability and reasoning
14	[55]	Distributed	Distributed	Centralized	Isolation
15	[63]	Distributed	Centralized	Centralized	Scalability, configurability, robustness, and security
16	[64]	Distributed	Centralized	Centralized	Load balancing, adaptation, robustness and extensibility
17	[73]	Distributed	Distributed	Centralized	Learning, reasoning, robustness
18	[74]	Distributed	Distributed	Centralized	Scalability, load balancing, fault tolerance and self-adaption
19	[78]	Distributed	Distributed	Centralized	Load balancing, and fault tolerance
21	[104]	Distributed	Centralized	Centralized	Self-learning and adaptation
			Decentralized structure (24 studies)		
1	[8]	Distributed	Decentralized	Decentralized	Abstraction, cooperation, mobility, robustness and fault-tolerance
2	[10]	Distributed	Decentralized Centralized.	Centralized/ Decentralized	Collaboration
3	[14]	Distributed	Distributed.	Decentralized	Coordination and scalability
4	[44]	Distributed	Decentralized	Decentralized	Fault-tolerance
5	[45]	Distributed	Decentralized	Decentralized	Robustness and fault tolerance
6	[49], [50], [57], [106]	Distributed	Distributed	Decentralized	Cooperation
7	[51]	Distributed	Distributed	Decentralized	Cooperation, learning and scalability
8	[54]	Distributed	Decentralized	Centralized	Self-orgnization
9	[59]	Distributed	Centralized	Centralized	Self-healing
10	[61]	Distributed	Distributed	Decentralized	Coordination, scalability and load balancing
11	[65]	Distributed	Distributed	Decentralized	Adaptation, self healing and cooperation
12	[66]	Distributed	Decentralized	Decentralized	Cooperation, coordination, scalability and learning
13	[68]	Distributed	Decentralized	Decentralized	Cooperation, learning, and self-adaptation
14	[70]	Distributed	Decentralized	Decentralized	Adaptation and intelligence
15	[95], [101]	Distributed	Decentralized	Decentralized	Cooperation
16	[79]	Distributed	Distributed	Decentralized	Cooperation, extendability,load balancing and fault tolerance
17	[87]	Distributed	Distributed	Centralized	Cooperation and reasoning
18	[93]	Distributed	Decentralized	Decentralized	Autonomy and cooperation
19	[102]	Distributed	Distributed	Decentralized	fault tolerance
20	[107]	Distributed	Decentralized	Decentralized	Cooperation, scalability
			Dynamically adaptive structure (20 studies)		
1	[5]	Distributed	Dynamic.	Centralized	Adaptation and collaboration
2	[16]	Distributed	Decentralized	Centralized	Collaboration, adaptation, reliability
3	[35]	Distributed	Decentralized	centralized	Adaptation, coordination and learnung
4	[37]	Distributed	Centralized	Centralized	self management
5	[48]	Distributed	Decentralized	Centralized/ Decentralized	Scalability, collaborative, learning, and robustness
7	[52]	Distributed	Mobile	Centralized	adaption, cooperation, collaboration and mobility.
8	[53]	Centralized	Decentralized	Centralized	Cooperation, collaboration, coordination and adaptation
9	[60]	Distributed	Distributed	Decentralized	Adaptation and scalability
10	[56]	Distributed	Decentralized	Decentralized/ Centralized	Cooperation
11	[58]	Distributed	Centralized	Centralized	Scalability
12	[62]	Distributed	Decentralized	Decentralized	Cooperation, adaptation, scalability and learning
13	[67]	Distributed	Centralized	Centralized	Adaptation, cooperation and intelligence
14	[76], [103]	Distributed	Distributed	Centralized	Scalability and self-organization
15	[85]	Distributed	Decentralized	Centralized	Cooperation and intelligence
16	[86]	Distributed	Decentralized	Decentralized	Interoperability, reasoning, scalability and flexibility.
17	[91]	Distributed	Distributed	Decentralized	Adaptation, collaboration and expansibility
18	[92]	Distributed	Decentralized	Centralized	Collaboration and adaptation
19	[98]	Distributed	Dynamic.	Distributed	Collaboration
20	[105]	Distributed	Decentralized	Centralized	Distribution, cooperation, intelligence

TABLE 4. Agent Types Exhibited by Multi-Agent IDS Architectures

#	Ref.	Agent types
1	[4]	Adaptive autonomous, cooperative, and intelligent
2	[5]	Autonomous, cooperative, and collaborative
3	[6]	Autonomous, intelligent, and mobile
4	[7]	Adaptive, autonomous, and intelligent
5	[8]	Autonomous,cooperative, intelligent, and mobile
6	[9], [10], [72], [94]	Adaptive, cooperative, and Intelligent
7	[11]	Adaptive and intelligent
8	[14]	Cooperative and intelligent
9	[35]	Adaptive and intelligent
10	[15]	Intelligent
11	[16]–[18]	Autonomous,adaptive, collaborative, cooperative, and intelligent
12	[19]	Cooperative, intelligent, and mobile
13	[20]	Intelligent and interactive
14	[37]	Autonomous and collaborative
15	[38]	Cooperative and intelligent
16	[42]	Intelligent,
17	[43]	Adaptive,deliberative, and intelligent
18	[44]	Autonomous and cooperative
19	[45]	Autonomous,adaptive,cooperative, deliberative,intelligent, and reflexive
20	[46]	Autonomous, cooperative, and intelligent
21	[47]	Autonomous, cooperative, and intelligent
22	[48]	Collaborative and intelligent
24	[49]	Autonomous, competitive, cooperative, and reactive
25	[50]	Autonomous, cooperative, collaborative and intelligent
26	[51]	Autonomous and cooperative
27	[52]	Adaptive, autonomous, cooperative, intelligent, and mobile
28	[53]	adaptive, autonomous, cooperative, and collaborative,
29	[54]	Adaptive and intelligent
30	[55]	Autonomous and collaborative
31	[56]	Autonomous, cooperative, collaborative, intelligent, and mobile,
32	[57]	Autonomous ,cooperative , and intelligent
33	[58]	Cooperative and intelligent
34	[59]	Intelligent
35	[60]	Autonomous, cooperative, and intelligent
36	[61]	Autonomous, Intelligent
37	[62]	Adaptive, autonomous, intelligent , and mobile
38	[63]	Autonomous and mobile
39	[64]	Autonomous, cooperative, collaborative, intelligent
40	[65]	Autonomous, cooperative, collaborative, and ntelligent
41	[66]	Cooperative, and intelligent
42	[67]	Adaptive, autonomous, cooperative, competitive, and intelligent
43	[68]	Adaptive, autonomous, intelligent, cooperative, and reactive
44	[70]	Adaptive, autonomous, cooperative, intelligent
45	[73]	Adaptive, cooperative, and intelligent
46	[74]	Adaptive and intelligent,
47	[76]	Adaptive, autonomous, intelligent, cooperative, and mobile
48	[78]	Autonomous and cooperative
49	[79]	Autonomous,cooperative, and intelligent
50	[85]	Cooperative and intelligent
51	[86]	Intelligent and mobile.
52	[87]	Intelligent and cooperative.
53	[89], [90]	Autonomous, collaborative, and intelligent
54	[91]	Adaptive and collaborative
55	[92]	Adaptive, cooperative, and intelligent
56	[93]	Autonomous, cooperative, and mobile
57	[94]	Adaptive, and intelligent
58	[95]	Autonomous and cooperative
59	[98]	Collaborative and mobile,
60	[100]	Collaborative and intelligent.
61	[101]	Autonomous, cooperative and reactive
62	[102]	Adaptive
63	[103]	Autonomous, cooperative, intelligent, and mobile
64	[104]	Cooperative
65	[105]	Intelligent, cooperative
66	[106]	Intelligent, autonomous, cooperative and collaborative
67	[107]	Collaborative

This SLR also found out that there are limitations in the measurement and metrics used for evaluating the multi-agent IDS architectures.

For future work, we are striving to develop a cooperative learning model for multi-agent IDS architectures. The proposed model is based on a reinforcement learning algorithm to let the agents learn by experience without prior knowledge. The cooperative learning model is to enable the agents to cooperate and learn faster. Additionally, we aim to make an adaptation model to enable the agents to choose the most suitable locations on the network for efficient execution. A new architecture is developed, and the experimental results will be presented. We recommend researchers to adapt suitable agent-based system methodologies to design and develop multi-agent IDSs. Researchers can also use the available network simulation software such as NS-2 and OMNeT++ for testing their proposed models. For example, one can study how a cooperative and adaptive multi-agent IDS using a machine learning algorithm can perform better than the traditional monolithic system. Developers can adapt the tested methodologies and frameworks of multi-agent IDSs to their solutions. In UML and AUML there are several diagrams and tools available for designing multi-agent systems. For implementation, Java Agent-Based Modelling (JABM) and Mesa framework in Python 3+ can be used to develop multi-agent IDSs.

APPENDIX A
MULTI-AGENT IDS ARCHITECTURAL PROPERTIES AND CHARACTERISTICS

See Table 3.

APPENDIX B
AGENT TYPES EXHIBITED BY MULTI-AGENT IDS ARCHITECTURES

See Table 4.

APPENDIX C
DATA ANALYSIS METHODS

See Table 5.

TABLE 5. Data Analysis Methods

Category	Algorithm	Reference	Remarks
Biologically inspired methods	Biological immune cells	[4], [70]	Advantages:
	Genetic algorithm	[7], [54], [64]	Disadvantages:
	Negative selection	[7], [52], [56], [64], [68]	High computational cost
	Danger Theory	[17], [19], [42], [58], [59], [70], [89], [90], [100]	
	Inspired by the human nervous system	[37]	
	Evolutionary game theory	[85]	
Machine learning	ANN	[9]–[11], [43], [48], [50], [72]–[74], [91], [94]	
	Association rules	[6]	
	Bayian Networks	[86]	
	Classification analysis	[5]	
	Clustering	[4]–[6], [14], [38], [46], [54], [59], [61], [70], [73]	
	Decision Tree	[43], [61], [73], [74], [86], [94]	Advantages:
	Extreme Learning Machine	[35]	Effectiveness
	Naïve Bayes	[11], [19], [44], [45], [54], [56], [68], [86], [94]	Disadvantages:
	Neural projection	[73]	Complexity
	Pattern recognition	[70], [73], [78], [107]	High computational cost
	Random Forest	[86]	
	Random Tree	[86]	
	Reinforcement learning	[16]–[18], [66], [92], [104], [105]	
	Rules based analysis	[6], [8], [46], [47], [51], [86], [95]	
	Similarity based	[20]	
	Sequential Minimal Optimization	[94]	
	Support victor machine	[35], [38], [58]–[60], [62], [73], [76], [103]	
	Online LaRank	[67]	
	Online random forest	[67]	
	Online random tree	[67]	
	Online multi-class Gradient Boost	[67]	
	Online multi-Class linear programming boost	[67]	
Statistical and mathematical models	Adaptive CUSUM	[102]	
	Adaptive Z-score	[102]	Advantages:
	Correlation analysis	[5]	Simplicity
	Gaussian mixture	[59]	Low computational cost
	Information theory	[46], [46]	Disadvantages:
	Minimum euclidean distance	[76]	Less effective
	Simple mathimatical models	[42], [89], [90], [100]	
	Time series	[9]	
Soft Computing methods	Evolutionary algorithm	[54], [85], [103]	Advantages:
	Fuzzy logic	[15]–[18], [50], [57], [65], [92], [106]	Disadvantages:
	Game theory	[16], [18], [85]	High computational cost
	Temporal logic	[4], [47], [108]	
Others	Pattern matching, interdisciplinary, antiworm, SNORT and expert systems	[49], [53], [63], [65], [78], [93], [98], [101]–[103], [107], [108]	Advantages: Simplicity Disadvantages: Less,effective

TABLE 6. The Selected Studies With Their Respective Bibliographic Information

#	year	Research title	R. type	Cit.	Ref.
1.	2014	Multi-Agent Heterogeneous Intrusion Detection System	C. ppr	1	[4]
2.	2011	Research on Adaptive Distributed Intrusion Detection System Model Based on Multi-Agent	C. ppr	2	[5]
3.	2012	Towards A Multi-agent Based Distributed Intrusion Detection System Using Data Mining Approaches	B. ch	13	[6]
4.	2010	Design of A New Distributed Model for Intrusion Detection System Based on Artificial Immune System	C. ppr	6	[7]
5.	2010	Worm detection using intelligent agents	C. ppr	6	[8]
6.	2010	SCMAS: A Distributed Hierarchical Multi-Agent Architecture for Blocking Attacks to Databases	J. art.	12	[9]
7.	2011	Detecting Cyber Intrusions in SCADA Networks Using Multi-Agent Collaboration	C. ppr	10	[10]
8.	2010	Agent-Based Intrusion Detection Mechanism	B. ch	2	[11]
9.	2016	Real-Time Intrusion Detection System Using Multi-Agent System	J. art.	0	[14]
11.	2016	Ontology Based Multi-Agent Model Intrusion Detection System for Detecting Web Service Attacks	J. art.	0	[15]
12.	2014	Cooperative Multi Agents for Intelligent Intrusion Detection and Prevention Systems	PhD th.	0	[16]
13.	2014	Cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks	J. art.	25	[17]
14.	2014	Cooperative game theoretic approach using fz. Q-learning 4 detecting and preventing intrusions in w.less sens. networks	J. art.	35	[18]
15.	2010	A hybrid approach for IEEE 802.11 intrusion detection based on AIS, MAS and naive Bayes	C. ppr	9	[19]
16.	2012	New Collaborative Intrusion Detection Architecture Based on Multi Agent Systems	J. art.	0	[20]
10.	2017	Real-time multi-agent system for an adaptive intrusion detection system	J. art.	0	[35]
17.	2010	Autonomic Agent-Based Self-Managed Intrusion Detection and Prevention System	C. ppr	13	[37]
18.	2010	Developing A Qualia-Based Multi-Agent Architecture for Use in Malware Detection	PhD th.	2	[38]
19.	2013	Agent-Based Artificial Immune Systems for Intrusion Detections: Inspiration from Danger Theory	B. ch	1	[42]
20.	2012	Detecting DoS Attack in web services by Using an Adaptive Multiagent Solution	J. art.	0	[43]
21.	2010	A Robust and Fault-Tolerant Distributed Intrusion Detection System	C. ppr	5	[44]
22.	2010	An Agent-Based Intrusion Detection System for Local Area Networks	J. art.	16	[45]
23.	2012	A Novel Network Attack Audit System Based on Multi-Agent Technology	J. art.	4	[46]
24.	2012	A Temporal Logic Based Approach to Multi-Agent Intrusion Detection and Prevention	J. art.	5	[47]
25.	2014	Multi-Agent Distributed Intrusion Detection System Model Based on BP Neural Network	J. art.	5	[48]
26.	2010	Agent-Based Modeling and Simulation of Botnets and Botnet Defense	C. ppr	43	[49]
27.	2013	Botnet Detection Technique for Corporate Area Network	C. ppr	2	[50]
28.	2010	A Multi-Agent-Based Distributed Intrusion Detection System	C. ppr	19	[51]
29.	2014	A Distributed Intrusion Detection System Using Multi-Agent AIS Approach	J. art.	17	[52]
30.	2012	A Model of Collaborative Intrusion Detection System Based on Multi-Agents	C. ppr	4	[53]
31.	2011	An Evolutionary Multi-Agent Approach to Anomaly Detection and Cyber defence	C. ppr	4	[54]
32.	2010	Detecting Malwares in Honeynet Using A Multi-Agent System	B. ch	1	[55]
33.	2014	An Immune Inspired Behavior-Based Multi-Agent Model for Detecting Network Clients' Misbehavior	J. art.	0	[56]
34.	2013	Multi-Agent Based Technique of Botnet Detection in Computer Systems	J. art.	12	[57]
35.	2010	Agent-Based Immunity for Computer Virus: Abstraction from Dendritic Cell Algorithm with Danger Theory	C. ppr.	8	[58]
36.	2010	Immune Multi Agent System for Intrusion Prevention and Self-Healing System Implement A Non-LinearClassification	C. ppr	7	[59]
37.	2011	Multi Agent System for Network Attack Classification Using Flow-Based Intrusion Detection	C. ppr	9	[60]
38.	2015	Hybrid Modified K-Means with C4.5 For Intrusion Detection Systems in Multiagent Systems	J. art.	0	[61]
39.	2010	Distributed and Cooperative Multi-Agent Based Intrusion Detection System	J. art.	3	[62]
40.	2010	An Agent Based Distributed Security System for Intrusion Detection in Computer Networks	J. art.	1	[63]
41.	2013	Distributed Agent Based Model for Intrusion Detection System Based on Artificial Immune System	J. art.	2	[64]
42.	2013	Multi-Agent Based Approach for Botnet Detection in A Corporate Area Network Using Fuzzy Logic	B. ch	1	[65]
43.	2014	Distributed Reinforcement Learning for Network Intrusion Response	PhD. th	9	[66]
44.	2012	Multi-Agent Malicious Behaviour Detection	PhD th.	3	[67]
45.	2014	A Two-Level Autonomous Intrusion Detection Model Inspired by The Immune System	J. art.	0	[68]
46.	2010	Immune Multiagent System for Network Intrusion Detection Using Non-Linear Classification Algorithm	J. art.	3	[70]
47.	2010	A Multi-Agent Based Solution to Detect and Block DoS Threats on Web Services	J. art.	0	[72]
48.	2013	Intrusion Detection Based on MAS to Detect and Block SQL Injection Through Data Mining	J. art.	23	[73]
49.	2011	An Adaptive Hier. Distributed Multi-Agent Architecture for Blocking Malicious SOAP Messages within W. Services	J. art.	42	[74]
50.	2011	A Multi Agent System for Flow-Based Intrusion Detection Using Reputation and Evolutionary Computation	M. th.	5	[76]
51.	2014	An Enhanced Multi-Agent Based Network Intrusion Detection System Using Shadow Log	J. art.	0	[78]
52.	2019	A Multi-Agent Model for Network Intrusion Detection	C. ppr.	0	[79]
53.	2018	A game theoretic approach to cooperative intrusion detection	J. art.	2	[85]
54.	2019	Performance Analysis of Classification Techniques by using Multi Agent Based Intrusion Detection System	J. art.	1	[86]
55.	2017	Multi-Agent Based Intrusion Prevention and Mitigation Architecture for Software Defined Networks	C. ppr.	2	[87]
56.	2011	Intrusion Detection Systems Adapted from Agent-Based Artificial Immune Systems	C. ppr	13	[89]
57.	2012	Host-Based Intrusion Detection Systems Adapted from Agent-Based Artificial Immune Systems	J. art.	41	[90]
58.	2011	Dynamic Distributed Information Security System Based on Multi-Agent	J. art.	0	[91]
59.	2019	Improving the Efficiency of IDPS by using Hybrid Methods from Artificial Intelligence	C. ppr.	0	[92]
60.	2018	Distributed Agent Based Technique For Detecting Distributed Denial-of-Service (DDoS) Attacks In WLAN	J. art.	1	[93]
61.	2011	Real-time CBR-agent with a mixture of experts in the reuse stage to classify and detect DoS attacks	J. art.	13	[94]
62.	2013	Experiments with Simulation of Botnets and defense Agent Teams	C. ppr	0	[95]
63.	2010	Design of A Multiagent System for Worm Spreading Reduction	J. art.	5	[98]
64.	2013	Multiagent-Based Computer Virus Detection Systems: Abstraction from Dendritic Cell Algorithm with Danger Theory	J. art.	6	[100]
65.	2012	Agent-Based Simulation of Cooperative Defense Against Botnets	J. art.	25	[101]
66.	2019	Distributed intrusion detection scheme for next generation networks	J. art.	0	[102]
67.	2013	A Multi Agent System for Flow-Based Intrusion Detection	M. th.	0	[103]
68.	2018	Some Security Model Based on Multi Agent Systems	C. ppr.	0	[104]
69.	2019	Adversarial environment reinforcement learning algorithm for intrusion detection	J. art.	0	[105]
70.	2012	Multi-Agent Based Approach of Botnet Detection in Computer Systems	B. ch	15	[106]
71.	2010	Multi-Agent Pattern Recognition Mechanism for Detecting Distributed Denial of Service Attacks	J. art.	6	[107]

J. art.=Journal article, C. ppr.=Conference pape, PhD th.=PhD thesis, B. ch.=Book Chapter, R. type=Resource type, Cit.=Citations, Ref.=Reference

APPENDIX D
THE SELECTED STUDIES WITH THEIR RESPECTIVE BIBLIOGRAPHIC INFORMATION

See Table 6.

APPENDIX E
SOURCES OF THE SELECTED STUDIES

See Table 7.

TABLE 7. Sources of the Selected Studies

#	Title	Type	Indexing
1.	Journal of Communication and Computer	J.	NONE
2.	Neurocomputing	J.	Sciencedirect, SCOPUS
3.	Telecommunication Systems	J.	SCOPUS
4.	International Journal of Computer Applications	J.	None
5.	International Journal of Digital Content Technology and its Applications(JDCTA)	J.	NONE
6.	International Journal of Innovative Computing, Information and Control	J.	SCOPUS
7.	Journal of Information Assurance and Security	J.	NONE
8.	Information Sciences	J.	Sciencedirect
9.	Journal of Information Assurance and Security	J.	NONE
10.	Expert Systems with Applications	J.	Sciencedirect, SCOPUS
11.	Advances in Distributed Computing and Artificial Intelligence Journal	J.	NONE
12.	IET Information Security	J.	NONE
13.	Physics Procedia	J.	Sciencedirect
14.	International Journal of Communication Network & Security	J.	NONE
15.	International Journal of Communication Networks and Information Security	J.	SCOPUS
16.	International Journal of Security and Its Applications	J.	NONE
17.	Concurrency and Computation: Practice and Experience	J.	NONE
18.	IAENG International Journal of Computer Science	J.	SCOPUS
19.	The Scientific World Journal	J.	SCOPUS
20.	International Journal of Engineering and Computer Science	J.	NONE
21.	International Journal of Research in Computer Science	J.	ISI
22.	Indian J. of Science and Technology	J.	SCOPUS
23.	Journal of Intelligent Information Systems	J.	SCOPUS
24.	Key Engineering Materials	J.	SCOPUS
25.	Oriental Journal of Computer Science & Technology	J.	NONE
26.	Computer Science & Information Technology (CS & IT)	J.	NONE
27.	Engineering Applications of Artificial Intelligence	J.	Sciencedirect, SCOPUS
28.	Journal of Applied Security Research	J.	SCOPUS
29.	Hybrid Intelligent Systems (HIS)	C.	IEEE
30.	International conference on Internet Monitoring and Protection (ICIMP), 2010 Fifth	C.	IEEE
31.	International Conference on Fuzzy Systems (FUZZ), 2011	C.	IEEE
32.	International Conference on Advanced Information Management and Service (IMS), 2010 6th	C.	IEEE
33.	16th International Conference on Intelligent System Application to Power Systems (ISAP),2011	C.	IEEE
34.	The 8th International Conference for Informatics and Information Technology (CIIT 2011)	C.	NONE
35.	23th International Workshop on Concurrency, Specification and Programming(CS&P)	C.	NONE
36.	Proceedings of the South African Information Security Multi-Conference	C.	IEEE
37.	1st International Conference on Parallel Distributed and Grid Computing (PDGC), 2010	C.	IEEE
38.	The 7th International Conference on Informatics and Systems (INFOS), 2010	C.	IEEE
39.	Modern information and electronic technologies	C.	NONE
40.	Conference on Cyber Conflict proceeding	C.	NONE
41.	7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013	C.	IEEE
42.	27th European Conference on Modelling and Simulation (ECMS)	C.	SCOPUS, ISI
43.	The 2nd International Conference on Computer and Automation Engineering (ICCAE),2010	C.	IEEE
44.	International Symposium in Information Technology (ITSim), 2010	C.	IEEE
45.	International Conference on Computer Science and Automation Engineering (CSAE), 2011	C.	IEEE
46.	International Conference on Computer Science & Service System (CSSS), 2012	C.	IEEE
47.	Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research	C.	ACM
48.	IEEE Congress on Evolutionary Computation(CEC), 2011	C.	IEEE
49.	Agents and Data Mining Interaction	B.	Springer
50.	Networked Digital Technologies	B.	Springer
51.	Agent and Multi-Agent Systems in Distributed Systems- Digital Economy and E-Commerce	B.	Springer
52.	Computational Intelligence in Security for Information Systems	B.	Springer
53.	Trends in Practical Applications of Agents and Multiagent Systems	B.	Springer
54.	Computer Networks	B.	Sciencedirect, SCOPUS
55.	Intelligent Distributed Computing IV	B.	Springer
56.	International Journal of Advanced Research in Computer Science;	J.	Google Scholar
57.	International Journal Computer Network and Information Security	J.	MECS
58.	2017 International Conference on Information and Communication Technology Convergence (ICTC)	C.	IEEE
59.	2019 International Conference on Information Technologies (InfoTech)	C.	IEEE
60.	1st International Conference on Smart Systems and Data Science (ICSSD)	C.	IEEE
61.	Journal of Network and Computer Applications	J.	Sciencedirect
62.	Journal of Computational Science	J.	Sciencedirect
63.	2018 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)	C.	IEEE

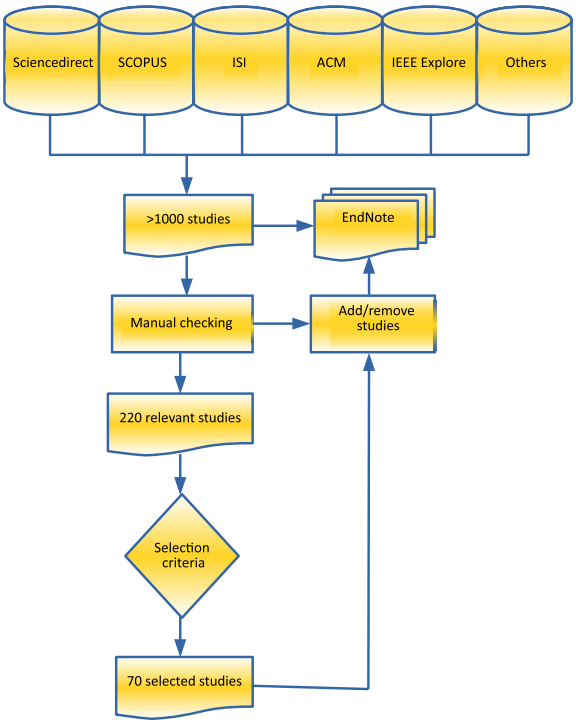
J. =Journal
C. =Conference
B. =Book

APPENDIX F
SOURCES OF THE SELECTED STUDIES

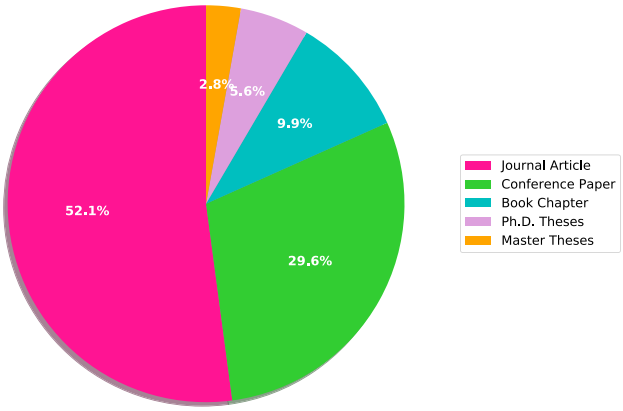
See Table 8.

#	Reference	Q1	Q2	Q3	Q4	Q5	Sum	Rate
1.	[4]	1	1	1	1	1	5	100%
2.	[5]	1	1	0	0	1	3	60%
3.	[6]	1	1	1	1	1	5	100%
4.	[7]	1	1	0	0	1	3	60%
5.	[8]	1	1	1	0.5	1	4.5	90%
6.	[9]	1	1	1	1	1	5	100%
7.	[10]	1	1	1	1	1	3	60%
8.	[11]	1	1	1	1	1	5	100%
9.	[14]	1	1	1	1	1	5	100%
10.	[15]	1	1	1	1	1	5	100%
11.	[16]	1	1	1	1	1	5	100%
12.	[17]	1	1	1	1	1	5	100%
13.	[18]	1	1	1	1	1	5	100%
14.	[19]	1	1	1	0.5	1	4.5	90%
15.	[20]	1	1	0	0	1	3	60%
18.	[35]	1	1	1	1	1	5	100%
16.	[37]	1	1	0	0	1	3	60%
17.	[38]	1	1	1	1	1	3	60%
19.	[42]	1	1	1	1	1	5	100%
20.	[43]	1	1	0	0	1	3	60%
21.	[44]	1	1	1	0.5	0.5	4	80%
22.	[45]	1	1	0	0	1	3	60%
23.	[46]	1	1	1	0.5	1	4.5	90%
24.	[47]	1	1	0	0	1	3	60%
25.	[48]	1	1	1	.5	1	4.5	90%
26.	[49]	1	1	1	1	1	5	100%
27.	[50]	1	1	1	0.5	1	4.5	90%
28.	[51]	1	1	1	0	0	3	60%
29.	[52]	1	1	1	1	1	5	100%
30.	[53]	1	1	0	0	1	3	60%
31.	[54]	1	1	1	0.5	1	4.5	90%
32.	[55]	1	1	0	0	1	3	60%
33.	[56]	1	1	1	1	1	5	100%
34.	[57]	1	1	1	0.5	1	4.5	90%
35.	[58]	1	1	1	1	1	5	100%
36.	[59]	1	1	1	1	0.5	4.5	100%
37.	[60]	1	1	1	1	1	5	100%
38.	[61]	1	1	1	1	1	5	100%
39.	[62]	1	1	0	0	1	3	60%
40.	[63]	1	1	0	0	1	3	60%
41.	[64]	1	1	1	1	1	5	100%
42.	[65]	1	1	1	1	1	5	100%
43.	[66]	1	1	1	1	1	5	100%
44.	[67]	1	1	1	1	1	1	100%
45.	[68]	1	1	1	1	1	5	100%
46.	[70]	1	1	1	1	1	5	100%
47.	[72]	1	1	0.5	0.5	1	4	80%
48.	[73]	1	1	1	1	1	5	100%
49.	[74]	1	1	1	1	1	5	100%
50.	[76]	1	1	1	1	1	5	100%
51.	[78]	1	1	0	0	1	3	60%
52.	[79]	1	1	1	0	0	3	60%
53.	[85]	1	1	1	1	1	5	100%
54.	[86]	1	1	1	1	1	5	100%
55.	[87]	1	1	1	1	1	5	100%
56.	[89]	1	1	1	1	1	5	100%
57.	[90]	1	1	1	1	1	5	100%
58.	[91]	1	1	1	1	1	5	100%
59.	[92]	1	1	1	1	.5	4.5	90%
60.	[93]	1	1	1	1	.5	4.5	90%
61.	[94]	1	1	1	1	1	5	90%
62.	[95]	1	1	1	1	1	5	100%
63.	[98]	1	1	1	1	1	5	100%
64.	[100]	1	1	1	1	1	5	100%
65.	[101]	1	1	1	1	1	5	100%
66.	[102]	1	1	1	1	1	5	100%
67.	[103]	1	1	1	1	1	5	100%
68.	[104]	1	1	1	1	.5	4.5	90%
69.	[105]	1	1	1	1	1	5	100%
70.	[106]	1	1	1	1	1	5	100%
71.	[107]	1	1	1	1	1	5	100%

APPENDIX G
FLOWCHART OF SEARCH AND STUDY SELECTION



APPENDIX H
SELECTED STUDIES PER MATERIAL TYPES



REFERENCES

[1] *Executive Summary, 23d Internet Security Threat Report*, NortonLife-Lock, Tempe, AK, USA, 2018.

[2] R. A. Kemmerer and G. Vigna, "Intrusion detection: A brief history and overview," *Computer*, vol. 35, no. 4, pp. 27–30, Apr. 2002.

[3] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[4] M. Pataky and D. P. Gruska, "Multi-agent heterogeneous intrusion detection system," in *Proc. Int. Workshop Concurrency, Specification Program.*, 2014, pp. 184–195.

[5] D. Huailin, X. Tianmao, W. Qingfeng, and L. Yangbin, "Research on adaptive distributed intrusion detection system model based on multi-agent," in *Proc. IEEE Int. Conf. Comput. Sci. Automat. Eng.*, Jun. 2011, pp. 182–185, doi: 10.1109/csae.2011.5953199.

[6] I. Brahmi, S. B. Yahia, H. Aouadi, and P. Poncelet, "Towards a multiagent-based distributed intrusion detection system using data mining approaches," in *Agents and Data Mining Interaction* (Lecture Notes in Computer Science), vol. 7103, L. Cao, A. L. C. Bazzan, A. L. Symeonidis, V. I. Gorodetsky, G. Weiss, and P. S. Yu, Eds. Berlin, Germany: Springer, 2012.

[7] F. Hosseinpour, K. A. Bakar, A. H. Hardoroudi, and A. F. Dareshur, "Design of a new distributed model for intrusion detection system based on artificial immune system," in *Proc. 6th Int. Conf. Adv. Inf. Manage. Service (IMS)*, 2010, pp. 378–383.

[8] S. M. Hussein and R. M. Bahgat, "Worm detection using intelligent agents," in *Proc. 7th Int. Conf. Inform. Syst. (INFOS)*, 2010, pp. 1–6.

[9] J. Bajo, J. M. Corchado, C. Pinzón, and B. Pérez-Lancho, "A distributed hierarchical multi-agent architecture for blocking attacks to databases," *Int. J. Innov. Comput., Inf. Control*, vol. 6, no. 9, pp. 3787–3817, 2010.

[10] A. F. Shosha, P. Gladyshev, S.-S. Wu, and C.-C. Liu, "Detecting cyber intrusions in SCADA networks using multi-agent collaboration," in *Proc. 16th Int. Conf. Intell. Syst. Appl. Power Syst.*, Sep. 2011, pp. 1–7.

[11] C. Pinzón, M. Navarro, and J. Bajo, "AIDeM: Agent-based intrusion detection mechanism," in *Proc. 8th Int. Conf. Practical Appl. Agents Multi-Agent Syst.*, 2010, pp. 347–354.

[12] H. Kozushko. *Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems*. Accessed: 2003. [Online]. Available: https://www.scribd.com/document/40215303/Intrusion-Detection-Paper

[13] X. Juang, X. Dongyan, X. Wang, and D. Xu, "Stealthy malware detection through vmm-based 'out-of-the-box' semantic view reconstruction," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 128–138.

[14] W. A. Yaseen, Z. A. Othman, and M. Z. Nazri, "Real-time intrusion detection system using multi-agent system," *Int. J. Comput. Sci.*, vol. 43, no. 1, pp. 80–90, 2016.

[15] K. Anusha and E. Sathiyamoorthy, "Omamids: Ontology based multi-agent model intrusion detection system for detecting Web service attacks," *J. Appl. Secur. Res.*, vol. 11, no. 4, pp. 489–508. 2016.

[16] S. Shamshirband, "Cooperative multi agents for intelligent intrusion detection and prevention systems," Ph.D. dissertation, Fac. Comput. Sci. Inf. Technol., Univ. Malaya, Kuala Lumpur, Malaysia, 2014.

[17] S. Shamshirband, N. Anuar, L. Kiah, V. Rohani, D. Petkovic, S. Misra, and A. Khan, "Co-FAIS: Cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 42, pp. 102–117, Jun. 2014.

[18] S. Shamshirband, A. Patel, N. B. Anuar, M. L. M. Kiah, and A. Abraham, "Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks," *Eng. Appl. Artif. Intell.*, vol. 32, pp. 228–241, Jun. 2014.

[19] M. Danziger and F. B. de Neto, "A hybrid approach for IEEE 802.11 intrusion detection based on AIS, MAS and naive Bayes," in *Proc. 10th Int. Conf. Hybrid Intell. Syst. (HIS)*, 2010, pp. 201–204.

[20] M. El Ajjouri, S. Benhadou, and H. Medromi, "New collaborative intrusion detection architecture based on multi agent systems," presented at the Int. Conf. Wireless Netw. Mobile Commun., Oct. 2015.

[21] M. Gajewski, J. M. Batalla, G. Mastorakis, and C. X. Mavromoustakis, "A distributed IDS architecture model for smart home systems," *Cluster Comput.*, vol. 22, no. S1, pp. 1739–1749, Jan. 2019.

[22] K. P. Sycara, "Multi-agent systems," *AI Mag.*, vol. 19, no. 2, p. 79, 1998.

[23] M. Glavic, "Agents and multi-agent systems: A short introduction for power engineers," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NISTIR 6416, 2006, [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=151202

[24] D. Dasgupta, "Immunity-based intrusion detection system: A general framework," in *Proc. 22nd Nat. Inf. Syst. Secur. Conf.*, Arlington, VA, USA, vol. 1, 1999, pp. 147–160.

[25] H. Albag, "Network & agent based intrusion detection systems," TU Munich Dept. Comput. Sci., İstanbul Tech. Univ., İstanbul, Turkey, 2001.

[26] S. Sathyanath and F. Sahin, "AISIMAM—An artificial immune system based intelligent multi agent model and its application to a mine detection problem," Rochester Inst. Technol., Rochester, NY, USA, RIT Scholar Works, 2002. [Online]. Available: https://scholarworks.rit.edu/other/455

[27] P. Kabiri and A. A. Ghorbani, "Research on intrusion detection and response: A survey," *IJ Netw. Secur.*, vol. 1, no. 2, pp. 84–102, 2005.

[28] A. Herrero and E. Corchado, "Multiagent systems for network intrusion detection: A review," in *Computational Intelligence in Security for Information Systems*. Berlin, Germany: Springer, 2009, pp. 143–154.

[29] S. A. Onashoga, A. D. Akinde, and A. S. Sodiya, "A strategic review of existing mobile agent-based intrusion detection systems," *Issues Informing Sci. Inf. Technol.*, vol. 6, pp. 669–682, Jan. 2009.

[30] I. A. Saeed, A. Selamat, and A. M. Abuagoub, "A survey on malware and malware detection systems," *Int. J. Comput. Appl.*, vol. 67, no. 16, pp. 25–31, 2013.

[31] A. D. Kulkarni and P. R. B. Joshi, "Review of intrusion detection systems (IDS) and agents based IDS," presented at the Int. Conf. Ind. Automat. Comput., Apr. 2014.

[32] R. Mitchell and I.-R. Chen, "A survey of intrusion detection in wireless network applications," *Comput. Commun.*, vol. 42, pp. 1–23, Apr. 2014.

[33] M. Wooldridge, "Agent-based computing," *Interoperable Commun. Netw.*, vol. 1, pp. 71–98, Jan. 1998.

[34] B. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering," Tech. Rep. EBSE-2007-01, 2007. [Online]. Available: https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

[35] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Real-time multi-agent system for an adaptive intrusion detection system," *Pattern Recognit. Lett.*, vol. 85, pp. 56–64, Jan. 2017.

[36] M. Shaw, "Writing good software engineering research papers," in *Proc. 25th Int. Conf. Softw. Eng.*, 2003, pp. 726–736.

[37] A. Patel, Q. Qassim, Z. Shukor, J. Nogueira, J. Júnior, C. Wills, and P. Federal, "Autonomic agent-based self-managed intrusion detection and prevention system," in *Proc. South Afr. Inf. Secur. Multi-Conf.*, 2010, pp. 223–234.

[38] B. D. Birrer, "Developing a qualia-based multi-agent architecture for use in malware detection," Ph.D. dissertation, Dept. Air Force, AIR Univ., Wright-Patterson Air Force Base, OH, USA, 2010.

[39] M. Glavic, "Agents and multi-agent systems: A short introduction for power engineers," Dept. Elect. Eng. Comput. Sci., Univ. Liege, Liège, Belgium, Tech. Rep., 2006. [Online]. Available: http://www.montefiore.ulg.ac.be/~glavic/MAS-Intro_Tech_report.pdf

[40] E. H. Spafford and D. Zamboni, "Intrusion detection using autonomous agents," *Comput. Netw.*, vol. 34, no. 4, pp. 547–570, Oct. 2000.

[41] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Auton. Agents Multi-Agent Syst.*, vol. 11, no. 3, pp. 387–434, Nov. 2005.

[42] C.-M. Ou, C. R. Ou, and Y.-T. Wang, "Agent-based artificial immune systems (ABAIS) for intrusion detections: Inspiration from danger theory," in *Agent and Multi-Agent Systems in Distributed Systems—Digital Economy and E-Commerce*. Berlin, Germany: Springer, 2013, ch. 4, pp. 67–94.

[43] C. I. Pinzón, N. Beliz, J. C. Rangel, and C. S. Hong, "Detecting DoS attack in Web services by using an adaptive multiagent solution," *Adv. Distrib. Comput. Artif. Intell. J.*, vol. 1, no. 2, pp. 57–63, 2012.

[44] J. Sen, "A robust and fault-tolerant distributed intrusion detection system," presented at the 1st Int. Conf. Parallel, Distrib. Grid Comput. (PDGC), Oct. 2010.

[45] J. Sen, "An agent-based intrusion detection system for local area networks," *Int. J. Commun. Netw. Inf. Secur.*, vol. 2, no. 2, pp. 128–140, 2010.

[46] W. Jianping, C. Min, and W. Xianwen, "A novel network attack audit system based on multi-agent technology," *Phys. Procedia*, vol. 25, pp. 2152–2157, Jan. 2012.

[47] P. Das and R. Niyogi, "A temporal logic based approach to multi-agent intrusion detection and prevention," *Int. J. Commun. Netw. Secur.*, vol. 1, no. 1, pp. 1–9, 2012.

[48] Z. Shuang-Can, H. Chen-Jun, and A. Z. Wei-Ming, "Multi-agent distributed intrusion detection system model based on BP neural network," *Int. J. Secur. Appl.*, vol. 8, no. 2, pp. 183–192, 2014.

[49] I. Kotenko, A. Konovalov, and A. Shorov, "Agent-based modeling and simulation of botnets and botnet defense," in *Proc. Conf. Cyber Conflict CCD COE Publications*, Tallinn, Estonia, 2010, pp. 21–44.

[50] O. Savenko, S. Lysenko, A. Kryshchuk, and Y. Klots, "Botnet detection technique for corporate area network," in *Proc. IEEE 7th Int. Conf. Intell. Data Acquisition Adv. Comput. Syst. (IDAACS)*, Sep. 2013, pp. 363–368.

[51] W. Huang, Y. An, and W. Du, "A multi-agent-based distributed intrusion detection system," in *Proc. 3rd Int. Conf. Adv. Comput. Theory Eng. (ICACTE)*, vol. 3, Aug. 2010, pp. V3-141–V3-143.

[52] N. A. Seresht and R. Azmi, "MAIS-IDS: A distributed intrusion detection system using multi-agent AIS approach," *Eng. Appl. Artif. Intell.*, vol. 35, pp. 286–298, Oct. 2014.

[53] Z. Ran, "A model of collaborative intrusion detection system based on multi-agents," in *Proc. Int. Conf. Comput. Sci. Service Syst. (CSSS)*, Aug. 2012, pp. 789–792.
[54] M. Carvalho and C. Perez, "An evolutionary multi-agent approach to anomaly detection and cyber defense," in *Proc. 7th Annu. Workshop Cyber Secur. Inf. Intell. Res. (CSIIRW)*, Oak Ridge, TN, USA, 2011, Art. no. 2179329.
[55] M. Szczepanik and I. Józwiak, "Detecting malwares in honeynet using a multi-agent system," in *Networked Digital Technologies* (Communications in Computer and Information Science), vol. 88, F. Zavoral, J. Yaghob, P. Pichappan, and E. El-Qawasmeh, Eds. Berlin, Germany: Springer, 2010, pp. 396–401.
[56] B. E. Noeparast, R. Ravanmehr, and R. Nasiri, "An immune inspired behavior-based multi-agent model for detecting network clients' misbehavior," *Int. J. Eng. Comput. Sci.*, vol. 3, no. 2, pp. 3822–3829, Feb. 2014.
[57] P. S. Savenko and A. F. Kryschuk, "MultiAgent based technique of botnet detection in computer systems," *Mod. Inf. Electron. Technol.*, vol. 291, pp. 171–180, May 2013.
[58] C.-M. Ou and C. R. Ou, "Agent-based immunity for computer virus: Abstraction from dendritic cell algorithm with danger theory," in *Advances in Grid and Pervasive Computing* (Lecture Notes in Computer Science), vol. 6104. Berlin, Germany: Springer, 2010, pp. 670–678.
[59] M. Elsadig, A. Abdullah, and B. B. Samir, "Immune multi agent system for intrusion prevention and self healing system implement a non-linear classification," in *Proc. Int. Symp. Inf. Technol. (ITSim)*, 2010, pp. 1–6.
[60] D. L. Hancock and G. B. Lamont, "Multi agent system for network attack classification using flow-based intrusion detection," presented at the IEEE Congr. Evol. Comput. (CEC), Jun. 2011.
[61] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Hybrid modified K-means with C4.5 for intrusion detection systems in multiagent systems," *Sci. World J.*, vol. 2015, Jun. 2015, Art. no. 294761.
[62] J. A. Renjit, "Distributed and cooperative multi-agent based intrusion detection system," *Indian J. Sci. Technol.*, vol. 3, no. 10, pp. 1070–1074, Oct. 2010.
[63] A. Saxena and A. K. Sharma, "An agent based distributed security system for intrusion detection in computer networks," *Int. J. Comput. Appl.*, vol. 12, no. 3, pp. 18–27, Dec. 2010.
[64] F. Hosseinpour, S. Ramadass, A. Meulenberg, P. V. Amoli, and A. Z. Moghaddasi, "Distributed agent based model for intrusion detection system based on artificial immune system," *Int. J. Digit. Content Technol. Appl.*, vol. 7, no. 9, p. 206, May 2013.
[65] O. Pomorova, O. Savenko, S. Lysenko, and A. Kryshchuk, "Multi-agent based approach for botnet detection in a corporate area network using fuzzy logic," in *Computer Networks* (Communications in Computer and Information Science), vol. 370, A. Kwiecień, P. Gaj, and P. Stera, Eds. Berlin, Germany: Springer, 2013, pp. 146–156.
[66] K. Malialis, "Distributed reinforcement learning for network intrusion response," M.S. thesis, Dept. Graduate Stud., Univ. Manitoba, Winnipeg, MB, Canada, 2014.
[67] R. P. D. Wegner, "Multi-agent malicious behaviour detection," Ph.D. dissertation, Dept. Comput. Sci., Univ. Manitoba, Winnipeg, MB, Canada 2012.
[68] E. B. Noeparast and R. Ravanmehr, "A two-level autonomous intrusion detection model inspired by the immune system," *Int. J. Res. Comput. Sci.*, vol. 4, no. 1, p. 11, 2014.
[69] T. Preisler and W. Renz, "Structural adaptations for self-organizing multi-agent systems," in *Proc. 7th Int. Conf. Adapt. Self-Adapt. Syst. Appl.*, 2015, pp. 413–425.
[70] M. E. Mohamed, B. B. Samir, and A. Abdullah, "Immune multiagent system for network intrusion detection using non-linear classification algorithm," *Int. J. Comput. Appl.*, vol. 12, no. 7, pp. 7–12, Dec. 2010.
[71] Microsoft. (Jan. 12, 2007). *Understanding Availability, Reliability, and Scalability*. [Online]. Available: https://technet.microsoft.com/en-us/library/aa996704(v=exchg.65).aspx
[72] C. I. Pinzón, F. D. Juan, D. L. Ana, G. Ana, and B. Javier, "A multi-agent based solution to detect and block DoS threats on Web services," *J. Inf. Assurance Secur.*, vol. 5, pp. 455–463, Jan. 2010.
[73] C. I. Pinzón, J. F. De Paz, Á. Herrero, E. Corchado, J. Bajo, and J. M. Corchado, "IdMAS-SQL: Intrusion detection based on MAS to detect and block SQL injection through data mining," *Inf. Sci.*, vol. 231, pp. 15–31, May 2013.
[74] C. I. Pinzón, J. Bajo, J. F. De Paz, and J. M. Corchado, "S-MAS: An adaptive hierarchical distributed multi-agent architecture for blocking malicious SOAP messages within Web Services environments," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5486–5499, 2011.
[75] P. Bellavista, A. Corradi, C. Federici, R. Montanari, and D. Tibaldi, "Security for mobile agents: Issues and challenges," in *Handbook of Mobile Computing*. Boca Raton, FL, USA: CRC Press, 2003.
[76] D. Hancock, "A multi agent system for flow-based intrusion detection using reputation and evolutionary computation," M.S. thesis, Dept. Air Force Air Univ., Air Force Inst. Technol., Wright-Patterson AFB, OH, USA, 2011.
[77] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi, and J. Al-Jaroodi, "A survey of load balancing in cloud computing: Challenges and algorithms," in *Proc. 2nd Symp. Netw. Cloud Comput. Appl.*, Dec. 2012, pp. 137–142.
[78] N. Singh, S. Krishan, and U. Kumar Singh, "An enhanced multi-agent based network intrusion detection system using shadow log," *Int. J. Comput. Appl.*, vol. 100, no. 9, pp. 1–5, Aug. 2014.
[79] S. Ouiazzane, M. Addou, and F. Barramou, "A multi-agent model for network intrusion detection," in *Proc. 1st Int. Conf. Smart Syst. Data Sci. (ICSSD)*, Oct. 2019, pp. 1–5.
[80] R. De Lemos *et al.*, "Software engineering for self-adaptive systems: A second research roadmap," in *Software Engineering for Self-Adaptive Systems II*. Berlin, Germany: Springer, 2013, pp. 1–32.
[81] C. Canal, J. M. Murillo, and P. Poizat, "Software adaptation," *L'Objet*, vol. 12, no. 1, pp. 9–31, 2006.
[82] S. Sen and G. Weiss, "Learning in multiagent systems," in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1999, p. 259.
[83] B. H. C. Cheng, R. De Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, B. Becker, N. Bencomo, Y. Brun, and B. Cukic, "Software engineering for self-adaptive systems: A research roadmap," in *Software Engineering for Self-Adaptive Systems*, B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, Eds. Berlin, Germany: Springer, 2009, pp. 1–26.
[84] N. Antzoulatos, E. Castro, D. Scrimieri, and S. Ratchev, "A multi-agent system architecture for self-configuration," in *Precision Assembly Technologies and Systems* (IFIP Advances in Information and Communication Technology), Chamonix, France, S. Ratchev, Ed. Berlin, Germany: Springer, Feb. 2014, pp. 118–125.
[85] Y. Guo, H. Zhang, L. Zhang, L. Fang, and F. Li, "A game theoretic approach to cooperative intrusion detection," *J. Comput. Sci.*, vol. 30, pp. 118–126, Jan. 2019.
[86] A. K. Saxena, S. Sinha, and P. Shukla, "Performance analysis of classification techniques by using multi agent based intrusion detection system," *Int. J. Comput. Netw. Inf. Secur.*, vol. 10, no. 3, p. 17, 2018.
[87] V. Sharma, "Multi-agent based intrusion prevention and mitigation architecture for software defined networks," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2017, pp. 686–692.
[88] F. G. Lobo, D. E. Goldberg, and M. Pelikan, "Time complexity of genetic algorithms on exponentially scaled problems," in *Proc. 2nd Annu. Conf. Genet. Evol. Comput.*, 2000, pp. 151–158.
[89] C.-M. Ou, Y.-T. Wang, and C. R. Ou, "Intrusion detection systems adapted from agent-based artificial immune systems," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jun. 2011, pp. 115–122.
[90] C.-M. Ou, "Host-based intrusion detection systems adapted from agent-based artificial immune systems," *Neurocomputing*, vol. 88, pp. 78–86, Jul. 2012.
[91] W. Peng, "Dynamic distributed information security system based on multi-agent," *Key Eng. Mater.*, vols. 460–461, pp. 433–438, Jan. 2011.
[92] G. Tsochev, R. Trifonov, R. Yoshinov, S. Manolov, and G. Pavlova, "Improving the efficiency of IDPS by using hybrid methods from artificial intelligence," in *Proc. Int. Conf. Inf. Technol. (InfoTech)*, 2019, pp. 1–4.
[93] H. Singh, "Distributed agent based technique for detecting distributed denial-of-service (DDoS) attacks in WLAN," *Int. J. Adv. Res. Comput. Sci.*, vol. 9, no. 1, pp. 375–380, Feb. 2018.
[94] C. I. Pinzón, J. F. De Paz, M. Navarro, J. Bajo, V. Julián, and J. M. Corchado, "Real-time CBR-agent with a mixture of experts in the reuse stage to classify and detect DoS attacks," *Appl. Soft Comput.*, vol. 11, no. 7, pp. 4384–4398, Oct. 2011.
[95] I. Kotenko, "Experiments with simulation of botnets and defense agent teams," in *Proc. 27th Eur. Conf. Modeling Simulation (ECMS)*, 2013, pp. 61–67.

[96] M. Berna-Koes, I. Nourbakhsh, and K. Sycara, "Communication efficiency in multi-agent systems," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, vol. 3, Apr./May 2004, pp. 2129–2134.
[97] M. Zaki and A. A. Hamouda, "Design of a multi_agent system for worm spreading_reduction," *J. Intell. Inf. Syst.*, vol. 35, no. 1, pp. 123–155, 2010.
[98] G. A. Fink, B. L. Chappell, T. G. Turner, and K. F. O'Donoghue, "A metrics-based approach to intrusion detection system evaluation for distributed real-time systems," in *Proc. 16th Int. Parallel Distrib. Process. Symp.*, 2002, p. 8.
[99] C.-M. Ou, "Multiagent-based computer virus detection systems: Abstraction from dendritic cell algorithm with danger theory," *Telecommun. Syst.*, vol. 52, no. 2, pp. 681–691, Jun. 2011.
[100] I. Kotenko, A. Konovalov, and A. Shorov, "Agent-based simulation of cooperative defense against botnets," *Concurrency Comput., Pract. Exper.*, vol. 24, no. 6, pp. 573–588, 2012.
[101] J. Manan, A. Ahmed, I. Ullah, L. Merghem-Boulahia, and D. Gaïti, "Distributed intrusion detection scheme for next generation networks," *J. Netw. Comput. Appl.*, vol. 147, Dec. 2019, Art. no. 102422.
[102] D. A. Ryan, "A multi agent system for flow-based intrusion detection," M.S. thesis, Dept. Air Force, Air Force Inst. Technol., Wright-Patterson AFB, OH, USA, 2013.
[103] G. Tsochev, R. Trifonov, R. Yoshinov, S. Manolov, G. Popov, and G. Pavlova, "Some security model based on multi agent systems," in *Proc. Int. Conf. Control, Artif. Intell., Robot. Optim. (ICCAIRO)*, May 2018, pp. 32–36.
[104] G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Comput. Netw.*, vol. 159, pp. 96–109, Aug. 2019.
[105] O. Savenko, S. Lysenko, and A. Kryschuk, "Multi-agent based approach of botnet detection in computer systems," in *Proc. Int. Conf. Comput. Netw.*, in Communications in Computer and Information Science, vol. 291, A. Kwiecien, P. Gaj, and P. Stera, Eds. Berlin, Germany: Springer, 2012, pp. 171–180.
[106] Z. A. Baig and K. Salah, "Multi-agent pattern recognition mechanism for detecting distributed denial of service attacks," *IET Inf. Secur.*, vol. 4, no. 4, pp. 333–343, 2010.
[107] I. A. Saeed and A. Selamat, "Multi-agent architecture with dynamic model checking for malware detection," *Labuan School Informat. Sci.*, vol. 15, no. 16, p. 47, 2013.

IMTITHAL A. SAEED received the B.Sc. (Hons.) and M.Sc. degrees in computer science from the Sudan University of Science and Technology, Sudan. She is currently pursuing the Ph.D. degree with Universiti Teknologi Malaysia. She is also a Lecturer with Prince Sattam Bin Abdulaziz University (KSA). Her research interests include software engineering, machine learning, and network security.

ALI SELAMAT (Member, IEEE) received the B.Sc. degree (Hons.) in IT from Teesside University, U.K., in 1997, the M.Sc. degree in distributed multimedia interactive systems from Lancaster University, U.K., in 1998, and the Dr.Eng. degree from Osaka Prefecture University, Japan, in 2003. He is currently a Professor with Universiti Teknologi Malaysia (UTM). He is also the Chair of IEEE Computer Society Malaysia Chapter. He is also the Editorial Boards of Knowledge Based Systems, Elsevier, *International Journal of Intelligent Information and Database Systems* (IJIDS), Inderscience Publications, *Vietnam Journal of Computer Science*, and Springer Publications.

MOHD FOAD ROHANI (Member, IEEE) received the B.E. degree (Hons.) in electrical and electronic engineering from University Malaya (UM), Kuala Lumpur, in 1994, the M.Sc. degree in electrical and electronic engineering from the University of Wales, Cardiff, U.K., in 1998, and the Ph.D. degree in the area of network security and pattern recognition (computer science) from Universiti Teknologi Malaysia (UTM), in 2013. He is currently a Lecturer with the Department of Computer Science, Faculty of Computing, UTM. His research interests include pattern recognition, digital signal processing (DSP), computer architectures, network communication, and security. He is a member of Malaysia Section and the Information Assurance and Information Security Research Group (IASRG).

ONDREJ KREJCAR received the M.Sc. degree in control and information systems and the Ph.D. degree in technical cybernetics from the Technical University of Ostrava, Ostrava, Czech Republic, in 2002 and 2008, respectively. In 2011, he became an Associate Professor in technical cybernetics with Technical University of Ostrava. He is currently a Professor and the Vice-Dean of Science and Research with the Faculty of Informatics and Management, University of Hradec Kralove. His research interests include biomedicine, image segmentation and recognition, video processing, biometrics, technical cybernetics, and ubiquitous computing.

JUNAID AHSENALI CHAUDHRY (Senior Member, IEEE) received the Ph.D. degree in cyber security from Ajou University, Suwon, South Korea, in 2009. He is currently a Faculty of Cyber Security with the College of Security and Intelligence, Embry–Riddle Aeronautical University, Prescott, AZ, USA. He has more than 15 years of rewarding experience in academia, industry, law enforcement, and in corporate world in information and cyber security domain. After completing the Ph.D. degree, he obtained training at the Harvard Business School, University of Amsterdam, and the Kaspersky Research Laboratory in cyber hunting and training. His research interests include critical infrastructure protection, digital forensics, and context aware network security problems. He is also a Practicing Engineer and a member of the High Technology Crime Investigation Association, Australian Computing Society, and Australian Information Security Association.