# Symptoms-Based Network Intrusion Detection System

Qais Saif Qassim[1(✉)], Norziana Jamil[2], and Mohammed Najah Mahdi[2]

[1] University of Technology and Applied Sciences – Ibri, Ibri, Sultanate of Oman
qais.aljanabi@ibrict.edu.om
[2] College of Computing and Informatics, Universiti Tenaga Nasional, Kajang, Malaysia
{norziana,najah.mahdi}@uniten.edu.my

**Abstract.** Protecting the network perimeters from malicious activities is a necessity and essential defence mechanism against cyberattacks. Network Intrusion Detection system (NIDS) is commonly used as a defense mechanism. This paper presents the Symptoms-based NIDS, a new intrusion detection system approach that learns the normal network behaviours through monitoring a range of network data attributes at the network and the transport layers. The proposed IDS consists of distributed anomaly detection agents and a centralised anomaly classification engine. The detection agents are located at the end nodes of the protected network, detecting anomalies by analysing network traffic and identifying abnormal activities. These agents will capture and analyse the network and the transport headers of individual packets for malicious activities. The agents will communicate with the centralised anomaly classification engine upon detecting a suspicious activity for attack prioritisation and classification. The paper presented a list of network attributes to be considered as classification features to identify anomalies.

**Keywords:** Signature · Anomaly · False alarms · Classification · Features · Machine learning

## 1 Introduction

Protecting asset's information from inside and outside threats can be a very demanding task. The primary purpose of an Intrusion Detection System (IDS) is to identify attackers trying to expose vulnerable resources on information systems and network services. Practically most of the existing intrusion detection systems are signature-based [1]. The performance of these systems is limited by the signature database of previously seen instances or attacks [2]. Therefore, the inability of signature-based IDS to detect novel attacks whose nature is unknown has stimulated the need for intelligent and efficient intrusion detection methods. The anomaly-based intrusion detection system is designed to uncover


H. Badioze Zaman et al. (Eds.): IVIC 2021, LNCS 13051, pp. 482–494, 2021.
https://doi.org/10.1007/978-3-030-90235-3_42

abnormal behaviour patterns [3]. It establishes a baseline of normal usage patterns and flags anything that widely deviates from it as a possible intrusion. The major benefit of anomaly-based detection methods is that they can effectively detect previously unknown threats. However, they cannot provide utterly accurate detection and are prone to generate high false alarms [4]. This is a serious concern in information security because false alarms can severely impact the protected information systems, such as the disruption of information availability because of IDS blockage in suspecting an attack attempt is overburdened by false alarm.

Deploying an anomaly-based intrusion detection system is usually implemented in two stages [5]; during the first stage, the system learns the network's normal behaviours under the assumption of the absence of attacks and/or malicious activities. In the second stage, the system monitors network traffic and system activates and compares them to the learned normal behavioural patterns. If a mismatch occurs, a level of "suspicion" is raised, and when the suspicion, in turn, trespasses a given threshold, the system triggers an alarm.

Typical attack definition consists of a combination of attack symptoms that are abnormal values of the observed network variables. Context knowledge can significantly improve intrusion detection accuracy and minimise the rate of false alarms. This work presents a new two-tier intrusion detection system that learns the normal ranges of values for network data attribute at the network and the transport layers. The proposed IDS work within a distributed multi-agent Intrusion Detection System architecture; the algorithm uses attack symptoms vectors for attack prioritisation and classification.

The rest of the paper is structured as follows; in Sect. 2, we present a brief review of intrusion detection systems and highlight the primary goal of this work. In Sect. 3, we introduce our symptoms based intrusion detection system. Section 4 presents the system implementation and finally Sect. 5 describes the future research plan and concludes the paper.

## 2 Literature Review

It is well known that intrusion detection systems play a vital role as the second line of defence against network-based and host-based attacks behind the firewall [6]. The key usage of an intrusion detection system is to detect abnormal or suspicious activities and raise the alarm whenever such activities are detected. Therefore, intrusion detection systems are becoming a prominent tool for many organisations after deploying firewall technology at the network perimeter [7].

Typically, the IDS systems are classified based on the method used in detecting malicious activities into one of the two approaches [8]; anomaly detection and signature detection. An anomaly detection approach is used to detect deviations from a previously learned behaviour, whereas any activity that significantly deviates from the normal behaviour is considered as intrusive. On the other hand, the misuse detection approach detects intrusions in terms of the characteristics of known attacks or system vulnerabilities; any action that conforms to the pattern of a known attack or vulnerability is considered intrusive.

Due to the diversity of cyberattacks and zero-day attacks signature-based IDS will likely miss an increasingly large share of attack attempts. In spite of this, most intrusion detection systems in use today are signature-based; whilst few anomaly-based IDSs have been deployed to date [9]. The reason behind that is, a signature-based IDS is easier to implement and simpler to configure and maintain than the anomaly-based. On the other hand, the deployment of an anomaly-based IDS typically requires training time, crucial system's attributes to monitor, and expert personnel [10,11]. To configure the anomaly-based IDS, several parameters need to be set, such as the duration of the training phase and the similarity metric. In addition to that, different environments may require a different set of attributes to monitor and parameters to configure. Therefore, common detection guidelines for anomaly-based IDS are hard to dedicate. Each anomaly detection mechanism has its unique requirements such as training time, system parameters and dataset collection, while all signature-based IDSs perform similarly in various environments.

Anomaly-based IDSs are mainly criticised based on three aspects, each of which increases the security specialist effort needed to configure and run. Firstly, as discussed previously, anomaly-based IDS generally raise a high number of false alarms. Secondly, an anomaly-based IDS usually works as a black-box [12]. Lastly, an anomaly-based IDS raises alarms without a precise classification or context detection information clarifies the rationale of generating the alarm [13]. As a matter of fact, the classification of a certain instance for a signature-based IDS is predetermined. In contrast, the classification for an anomaly-based IDS depends on the training dataset. Thus, different anomaly-based model instances could classify the same instance differently [14].

False alarms are well-known problems of IDSs in general and anomaly-based IDSs in particular [15]. Security analysts have to verify each raised alarm; thus, systems prone to raise a high amount of false alarms will require many personnel and excessive time for alarm verification. Two distinct trends affect the rate of false alarms; primarily, most anomaly-based detection engines utilise statistical models, a distance function, and a threshold value to detect anomalies [13]. For that reason, there is an intrinsic tie between attacks detected and false alarms raised; when adjusting the threshold value to detect a larger number of attacks, the number of false alarms increases as well. Therefore, it is practically difficult to achieve ideal attack detection with no false alarms all at once [16,17].

Consequently, anomaly-based IDS have to be tuned by setting an appropriate threshold value. Secondly, since intrusions are rare events, and because detection engines cannot achieve both optimal detection rate and a negligible false alarm rate, a greater rate of false alarms will be generated than the desired and expected rate. This problem is commonly known as the base-rate fallacy, and it stems directly from Bayes' theorem [18].

Another limitation of an anomaly-based system is that it carries out the detection process as a black-box [19]. System administrators have little control over the process flow and its configuration; reasonably, they can merely configure the similarity metric used to discern legitimate traffic from malicious activities. Most anomaly-based IDSs employ complex mathematical models (such as neural networks, genetic algorithms and data mining algorithms) [20]. Therefore,

system administrators can neither precisely understand how the IDS engine distinguishes normal instances nor refine the IDS model to avoid certain false alarms or improve attack detection.

Unlike signature-based IDS, the anomaly-based IDS lack attack classification. The main concept of an anomaly-based IDS is that it raises the alarm every time it detects an activity that deviates from the baseline model of the normal behaviour [19]. Therefore, the cause of the anomaly itself is unknown to the intrusion detection system. The generated alarm holds little information to determine the attack class. Network-based systems generally include the targeted IP address, network port used, and the IP source of the attack. This is because the detection engine's model is implemented based on learning the normal behaviours during a certain time. Therefore, it is difficult to develop an offline classifier suitable for any anomaly-based IDS instance.

An anomaly-based IDS is hypothetically supposed to detect unknown attacks or slight modifications of well-known attacks. Manual classification and the application of some heuristics-based approaches are possible options [19]. However, manual classification is not feasible due to a large number of false alarms generated, and the heuristics deliver results in a restricted context only because the "traits" of each attack must be known. In addition to that, because alarms are generated unclassified and hold little information to determine the attack class, no automatic countermeasure can be activated to react to a certain threat [6]. Because of all the limitations listed above for an anomaly-based IDS, the primary objectives of this work is to propose a method that can enhance and improve the usability of the anomaly-based intrusion detection system

## 3 Symptoms-Based Network Intrusion Detection System

The proposed IDS consists of two interacting components: the Anomaly Detection Agent (ADA) and the Attack Classification Engine (ACE). The ADA processes network traffic, analyse traffic statistics and extracts significant network traffic attributes. Once ADA has detected an abnormal activity, the collected information is passed to the ACE that automatically determines the attack class. This section presents the proposed system in detail.

### 3.1 An Overview

The proposed IDS is a completely network-based intrusion detection system that identifies intrusions by examining network traffic. As illustrated in Fig. 1, it consists of anomaly detection agents distributed in the networks' end nodes and a centralised correlation engine. The distributed agents are responsible for interpreting the data stream (network traffic) arriving at the particular node. The main function of these agents is to capture all network traffic generated by the specified host and analyse the content of individual packets for malicious activities. The ADA communicates with the anomaly classification engine upon detecting suspicious activities for attack prioritisation and classification.

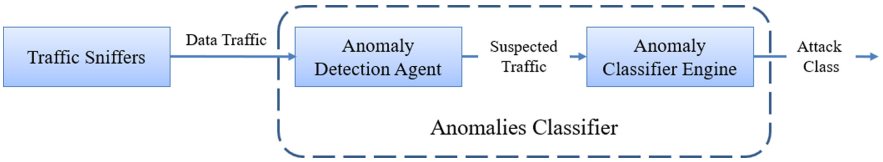


**Fig. 1.** conceptual framework of the proposed system

### 3.2 Attribute Vector

Before proceeding with system architecture any further, some concepts used in this paper should be explained in detail. The proposed IDS learns the normal behaviour of the network by monitoring and analysing several network data attributes at the network and the transport layers. The attributes have to be collected from the network protocol stack. The collected attributes will be used to produce a feature set containing statistical information that reflects the amount of change within each time interval. The monitored attributes can be arranged under two general categories: attributes extracted from the IP header and attributes extracted from the TCP header listed in Table 1.

As illustrated, twenty attributes to be monitored and analysed have been collected from the network protocol stack. The monitored attributes will be represented as a vector of 20 elements as shown in Eq. 1, where each element represents its designated value as described in Table 1.

$$f = [F_1, F_2, F_3, F_4, ....F_{20}] \tag{1}$$

For instance, $F_1$ represents the value of the IP header's Time To Live (TTL) field, an eight-bit field that holds a value specified in seconds and helps prevent datagrams from persisting on the Internet.

### 3.3 Anomaly Detection Agent (ADA)

Any network attack causes a certain abnormal behaviour, and it is imperative to be able to identify this abnormal behaviour accurately. To meet this challenge, the proposed IDS utilises decision correlation metrics to measure the attributes of the network that will most likely identify an attack in progress.

The proposed IDS works as follows; the distributed agents monitor the incoming traffic by analysing selected network data attributes at the network and the transport layers and estimate its deviation from the normal behaviours (baseline) learned during the training phase. The agents then generate symptoms vector based on the magnitude that the monitored network data attributes that deviate from the baseline, which represents the strength of the participation of each monitored attribute.

The central tendency rule backed by the arithmetic mean plays a vital role in the detection mechanism. They have been used to calculate the central value that the magnitude of deviations is trend to cluster around. In other words, if

**Table 1.** IP and TCP considered features

| Attribute | Description |
|---|---|
| F1 | The number of distinct values of the Time To Live field of the IP header |
| F2 | The number of foreign IP addresses |
| F3 | The number of inbound packets that were discarded due to errors in their IP headers |
| F4 | The number of inbound packets for which this host was not their final IP destination |
| F5 | The number of inbound packets received successfully but discarded because of an unknown or unsupported protocol |
| F6 | The number of inbound packets which were discarded because of the IP checksum |
| F7 | The average number of packets sent and received |
| F8 | The ratio of packets sent and received to the number of foreign IP addresses |
| F9 | Weigthed sum of TOS field |
| F10 | The number of distinct values of the header checksum field of the IP header |
| F11 | The number of TCP connections |
| F12 | The number of half open connections |
| F13 | The maximum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds |
| F14 | The minimum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds |
| F15 | The limit on the total number of TCP connections the host can support |
| F16 | The number of times TCP connections have made a direct transition to the SYN SENT state from the CLOSED state during the observation period |
| F17 | The number of times TCP connections have made a direct transition to the SYN RCVD state from the LISTEN state during the observation period |
| F18 | The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE WAIT state during the observation period |
| F19 | The total number of segments received, including those received in error |
| F20 | The total number of segments sent |

the central tendency of the generated symptoms vector is greater than zero (or a selected threshold value), it indicates the presents of an anomaly due to the deviation from the normal network behaviours. To reduce the rate of false alarms, a threshold value greater than zero to be considered based on the protection level required. Therefore, in this work, ranges of threshold values have been considered

for comparison. Once the central tendency exceeds the defined threshold, an alarm will be generated indicating malicious activity. When a malicious activity has been indicated, the symptoms vector will be directed to the correlation module to identify the attack class.

### 3.4 Anomaly Classification Engine (ACE)

The anomaly classification engine is responsible for identifying the anomaly mechanism based on a predefined set of patterns of known attack mechanisms that are defined in the CAPEC and CVE databases. The ACE represents a modified signature-based intrusion detection system. However, it traces the symptoms vector to the most identical attack mechanism instead of detecting an attack. Identifying the attack mechanism (class) is an easy and effortless task comparing to detecting the attack itself. The attack class identified will help measure the risk exposed by the detected attack. The ACE is trained with several types of attack mechanisms to build a classification model. The attack mechanism information can be provided in several ways, either manually by an operator or automatically by extracting specific information from the known attack signatures.

### 3.5 The Proposed IDS Architecture

This section describes the main components and the working modes of the proposed IDS in detail. Figure 2 depicts system architecture and its principal sub-systems. As mentioned earlier, the proposed IDS consists of two interacting components: the ADA and the ACE. As described earlier, the anomaly detection agent processes network traffic, analyse traffic statistics and extracts significant information. Once ADA has detected an abnormal activity, the collected data is passed to the ACE, which automatically determines the attack class.

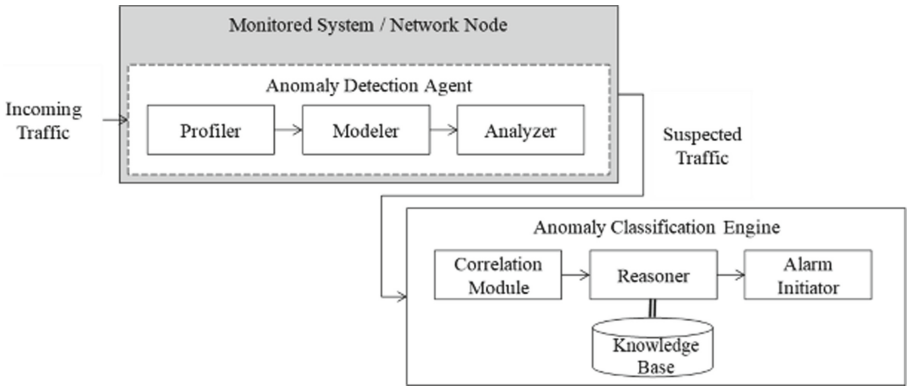


**Fig. 2.** System architecture.

The general modules of the system are described as follow; The Profiler subsystem is responsible for extracting the required information from the incoming traffic to generate an appropriate traffic representation (traffic vector); it is responsible for tracing the presence of every monitored network attribute in the IP and TCP header packets for a specified period of time. The Modeler sub-system is responsible for applying the specified conditional mapping criteria to transform the generated traffic vector, and as a result, it generates the symptoms vectors. At this stage, the analyser module estimates the weight of abnormalities by calculating the central tendency. The analyser then tests the result in opposition to the defined threshold value. Once the central tendency exceeds the defined threshold, an alarm will be generated indicating malicious activity. When a malicious activity has been indicated, the symptoms vector will be directed to the anomaly classification engine to identify the attack class.

The alarm correlation module sub-system plays an essential role in anomaly classification; it examinations the correlation strength of the detected anomaly against the defined pattern. This work uses matrices to describe the correlation strengths between the detected anomalies against the defined patterns; the strengths are calculated based on weighted absolute differences.

The alarm correlation module works in solidarity with the Reasoner. The reasoned or reasoning engine is in charge of inferring logical evaluation of correlation strength of the detected anomaly against the defined pattern. The reasoning engine will generate the attack probability score and direct the result to the Alarm Initiator. The Alarm Initiator generates an alarm with a standard data format using the Intrusion Detection Message Exchange Format (IDMEF).

### 3.6 System Implementation

The proposed detection system has been implemented in two stages; during the first stage, the system learns the network's normal behaviour (the normal range of values of the monitored network attributes) under the assumption of the absence of attacks and malicious activities. In the second stage, the system monitors network traffic activates and compares them to the learned normal behavioural patterns. If a mismatch occurs, a level of "suspicion" is raised, and when the suspicion, in turn, trespasses a given threshold, the system triggers an alarm. The alarm is not considered an incident yet and is not forwarded to the prevention system; instead, it will be forwarded to the correlation engine for further analysis. To better understand the mechanism of the proposed system, the operational phases of the system have been divided into two phases as follows;

**Phase I: Learning the Normal Activities.** During the learning phase, the agents learn the normal network behaviours by monitoring network data attributes at the network and the transport headers. The agents observe and record every network attribute in the IP and TCP header packets for a specified period of time. The proposed IDS uses the average rate of occurrence (arithmetic mean) and the variation from the average during the training phase to estimate

an anomaly's chance while in the detection phase. If a network attribute is observed n times with m values for a defined period of time, then the mean $M_{Fi}$ and the standard deviation $\sigma_{Fi}$ for each attribute can be calculated as follow;

$$M_{Fi} = \frac{1}{n}\sum_{j=1}^{n} m_j \tag{2}$$

$$\sigma_{Fi} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(m_j - M_{Fi})^2} \tag{3}$$

The process flow of the learning phase is illustrated in Fig. 3. As shown, the proposed IDS starts extracting the required data from the IP and TCP header packets. A new vector should be created for every new connection.

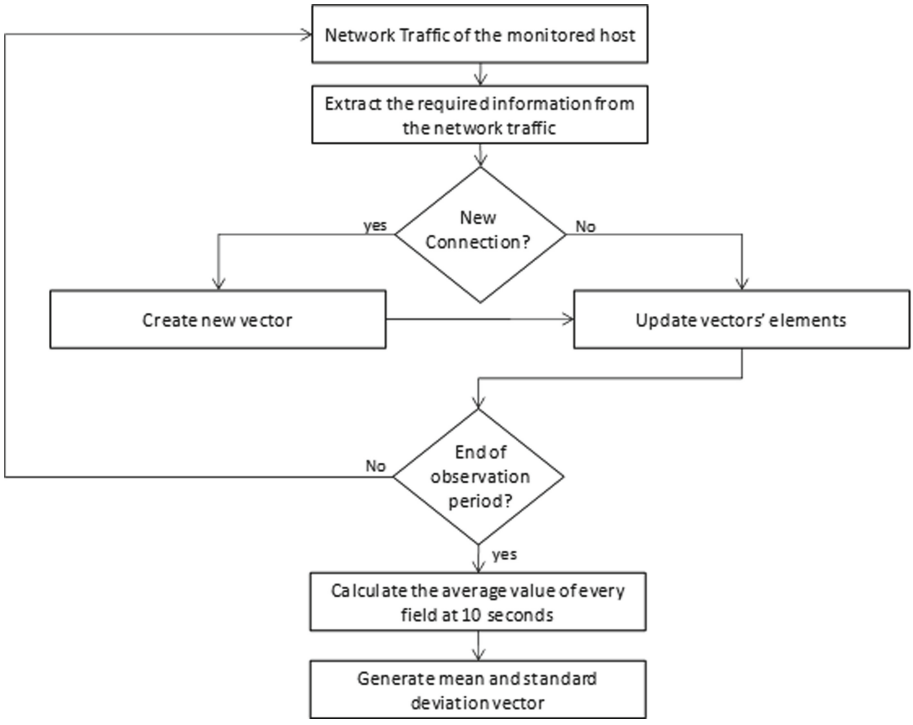


**Fig. 3.** Functional model of the system (training phase)

At the end of the observation period, the system starts to calculate the average value ($M_{benign}$) and the standard deviation ($\sigma_{benign}$) of every connection at t seconds window to generate the normal behaviour vectors. The calculated

mean and standard deviation are represented as vectors, where each element represents its designated value as follows;

$$M_{benign} = [M_{F1}, M_{F2}, M_{F3}, M_{F4}, ....M_{F20}] \tag{4}$$

$$\sigma_{benign} = [\sigma_{F1}, \sigma_{F2}, \sigma_{F3}, \sigma_{F4}, ....\sigma_{F20}] \tag{5}$$

**Phase II: Testing Phase/Detection Phase.** In the detection phase, the agents observe and record (count the presence of) the 20 attributes from the network protocol stack for every t seconds. The process flow of this phase is illustrated in Fig. 4. The monitored attributes will be represented as a vector (traffic vector), as shown in Eq. 1, where each element represent its designated value. The agents then generate a symptoms vector based on the magnitude that the monitored network data attributes have deviated from the baseline (normal behaviour vector). The elements of the symptoms vector represent the strength of the participation of each monitored attribute. The estimated symptoms elements are represented as a vector (symptoms vector), as shown in Eq. 6.

$$F_{symptoms} = [S_1, S_2, S_3, S_4, ....S_{20}] \tag{6}$$

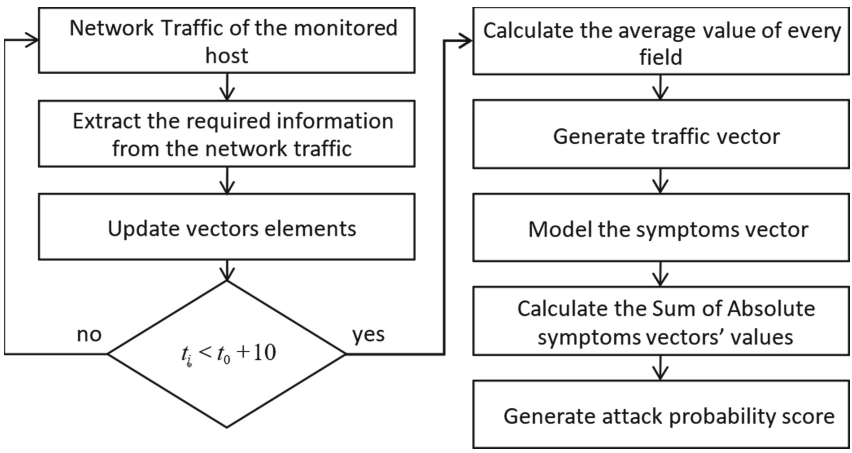


**Fig. 4.** Functional model of the system (detection phase)

The agents will use predefined conditional criteria to estimate the values of the symptoms vector. In this work, we have proposed five conditional rules defined as follow:

$$S_i = \begin{cases} 2, & \text{if } (M_{Fi} + 2 * \sigma_{Fi} \leq F_i) \\ 1, & \text{if } (M_{Fi} + 2 * \sigma_{Fi} < F_i < M_F i + \sigma_{Fi}) \\ 0, & \text{if } (M_{Fi} + \sigma_{Fi} \leq F_i \leq M_F i - \sigma_{Fi}) \\ -1, & \text{if } (M_{Fi} - \sigma_{Fi} < F_i < M_F i - 2 * \sigma_{Fi}) \\ -2, & \text{if } (F_i \leq M_{Fi} + 2 * \sigma_{Fi}) \end{cases}$$

For example, symptoms elements $S_i$ can be represented by value 2, if the network attribute $F_i$ is greater than or equal to its normal behaviours' mean value $M_{Fi}$ plus twice its standard deviation $\sigma_{Fi}$. The sign of deviation (positive or negative) represents the direction of that difference (it is larger when the sign is positive and smaller if it is negative). The magnitude of the value indicates the size of the difference. The agents will be generating the new symptoms vector every t seconds. Then, the agents will calculate the weight of abnormalities by calculating the central tendency using the following formula:

$$M_{Fi} = \frac{1}{n} \sum_{j=0}^{n} m_j \tag{7}$$

If the central tendency T is greater than or equal to a defined threshold value, then an alarm will be generated indicating a malicious activity; otherwise, it is considered as benign activity. Once a malicious activity has been detected, the symptoms vector will be directed to the correlation module for identifying the attack class. Upon receiving the traffic vector, the central system will compare it with the knowledge base K of the attack pattern and generate an attack symptoms correlation matrix.

The symptoms correlation matrix will determine the possible attack class based on correlating the symptoms vector of the examined instance to the symptoms vector of the attack patterns stored in the knowledge base. The attack pattern that scores the highest value of  will determine the class of the attack.

$$\mu = (1 - \sum_{i=1}^{n} |K_i - F_i|) * 100\% \tag{8}$$

## 4 Conclusion and Future Work

With the rapidly increasing rate of network attacks in recent years, network-based intrusion detection systems have become a critical network defence mechanism. However, these systems typically generate a vast amount of alarms that can be unmanageable and mixed with a large number of false alarms, especially in large-scale networks, which result in a huge challenge on the efficiency and accuracy of network attack detection. Recent research has shown that anomaly-based intrusion detection systems are more vulnerable to positive false alarms than signature-based detection systems. This is because of the detection nature of anomaly-based IDS. It raises the alarm every time it detects an activity that

deviates from the baseline model of the normal behaviour. Therefore, the cause of the anomaly itself is unknown to the intrusion detection system.

The alarm classification approaches have become a popular solution in the alarm management process. It tends to enhance the quality of the generated alarms through filter-out the false alarms. Many researchers have considered classifying the generated alarms to reduce false alarms in intrusion detection systems. As a result, the amount of alarms presented to the security personnel is reduced, and the time required to validate and manage the IDS alarms is minimised. Therefore, this work considers the alarm classification approach to propose a new alarm classification method to enhance the quality of the generated alarms by filter-out false alarms. The main goal of this work is to present a new anomaly-detection system that embraces two stages for malicious activity detection and alarm classification. The first stage involves detecting unusual activities based on a previously learnt model. On the other hand, The alarm classification stage is intended to classify the detected activity based on known attack patterns. The alarms which are unable to be classified will be tagged as possible false alarms for further analysis. This work presents the concept of the proposed detection system, out future research will attempt to construct a testbed to generate the required dataset to evaluate the proposed architecture.

**Acknowledgement.** The research leading to these results has received funding from the Research Council (TRC) of the Sultanate of Oman under the Open Research Grant Program. TRC Grant Agreement No [BFP/RGP/ICT/20/377]

## References

1. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity **2**(1), 1–22 (2019). https://doi.org/10.1186/s42400-019-0038-7
2. Einy, S., Oz, C., Navaei, Y.D.: The anomaly- and signature-based IDS for network security using hybrid inference systems. Mathematical Problems in Engineering 2021 (2021)
3. Torabi, M., Udzir, N.I., Abdullah, M.T., Yaakob, R.: A review on feature selection and ensemble techniques for intrusion detection system. Int. J. Found. Comput. Sci. **12**, 1–13 (2021)
4. Singh, R.R., Gupta, N., Kumar, S.: To reduce the false alarm in intrusion detection systems using self-organizing. Int. J. Soft Comput. Eng. **1**(2), 27–32 (2011)
5. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 1–58 (2009)
6. Rhee, M.Y.: Internet Firewalls for Trusted Security. Wiley (2013)
7. Sundaramurthy, S.C., Case, J., Truong, T., Zomlot, L., Hoffmann, M.: A tale of three security operation centers. In: Proceedings of the 2014 ACM Workshop on Security Information Workers - SIW 2014, pp. 43–50. ACM Press, New York (2014)
8. Ghorbani, A.A., Lu, W., Tavallaee, M.: Network Intrusion Detection and Prevention: Concepts and Techniques. Springer, Boston (2010)
9. Xue, Y., Wang, D., Zhang, L.: Traffic classification: issues and challenges. In: 2013 International Conference on Computing, Networking and Communications (ICNC), pp. 545–549. IEEE (2013)

10. Guimaraes, M., Murray, M.: Overview of intrusion detection and intrusion prevention. In: InfoSecCD '08: Proceedings of the 5th Annual Conference on Information Security Curriculum Development. Association for Computing Machinery. Kennesaw Georgia (2008)
11. Thottan, M., Liu, G., Ji, C.: Anomaly detection approaches for communication networks. In: Cormode, G., Thottan, M. (eds.) Algorithms for Next Generation Networks, pp. 239–261. Springer, London (2010)
12. Siraj, M., Hashim, M.: Zaiton: network intrusion alert correlation challenges and techniques. Jurnal Teknologi Maklumat. **20**, 12–36 (2008)
13. Om, H., Hazra, T.: Statistical techniques in anomaly intrusion detection system. Int. J. Adv. Eng. Technol. **5**, 387–398 (2012)
14. Bolzoni, D., Etalle, S., Hartel, P.H.: Panacea: automating attack classification for anomaly-based network intrusion detection systems. In: Kirda, E., Jha, S., Balzarotti, D. (eds.) RAID 2009. LNCS, vol. 5758, pp. 1–20. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04342-0_1
15. Om, H., Kundu, A.: A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. In: 2012 1st International Conference on Recent Advances in Information Technology (RAIT), pp. 131–136. IEEE (2012)
16. Spathoulas, G., Katsikas, S.: Methods for post-processing of alerts in intrusion detection. Int. J. Inf. Secur. Sci. **2**, 64–80 (2013)
17. Stiawan, D., Yaseen, A.L.A., Shakhatreh, I., Idris, M.Y., Bakar, K.A.B.U., Abdullah, A.H.: Intrusion prevention system: a survey. J. Theoretical Appl. Inf. Technol. (2011)
18. Karasek, D.Y., Kim, J., Kemmoe, V.Y., Bhuiyan, M.Z.A., Cho, S., Son, J.: SuperB: superior behavior-based anomaly detection defining authorized users' traffic patterns. In: International Conference on Computer Communications and Networks, ICCCN. Hawaii, USA (2020)
19. Bolzoni, D.: Revisiting anomaly-based network intrusion detection systems. University of Twente, Enschede (2009)
20. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: techniques, systems and challenges. Comput. Secur. **28**, 18–28 (2009)