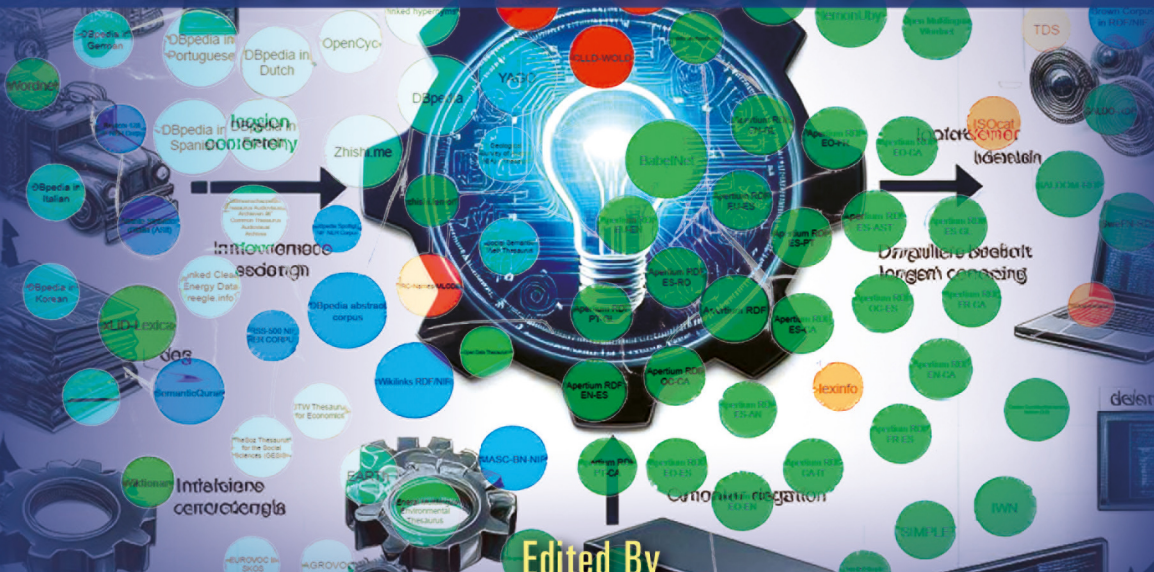
NATURAL LANGUAGE PROCESSING FOR SOFTWARE ENGINEERING



Edited By
Rajesh Kumar Chakrawarti, Ranjana Sikarwar,
Sanjaya Kumar Sarangi, Samson Arun Raj Albert Raj, Shweta Gupta,
Krishnan Sakthidasan Sankaran and Romil Rawat

Scrivener Publishing

WILEY

Natural Language Processing for Software Engineering

Scrivener Publishing
100 Cummings Center, Suite 541J
Beverly, MA 01915-6106

Publishers at Scrivener
Martin Scrivener (martin@scrivenerpublishing.com)
Phillip Carmical (pcarmical@scrivenerpublishing.com)

Natural Language Processing for Software Engineering

Edited by

Rajesh Kumar Chakrawarti

Ranjana Sikarwar

Sanjaya Kumar Sarangi

Samson Arun Raj Albert Raj

Shweta Gupta

Krishnan Sakthidasan Sankaran

and

Romil Rawat

WILEY

This edition first published 2025 by John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA and Scrivener Publishing LLC, 100 Cummings Center, Suite 541J, Beverly, MA 01915, USA

For more information about Scrivener publications please visit www.scrivenerpublishing.com.

Wiley Global Headquarters
111 River Street, Hoboken, NJ 07030, USA

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Library of Congress Cataloging-in-Publication Data

ISBN 9781394272433

Front cover images supplied by Adobe Firefly
Cover design by Russell Richardson

Set in size of 11pt and Minion Pro by Manila Typesetting Company, Makati, Philippines

Printed in the USA

10 9 8 7 6 5 4 3 2 1

Contents

Preface

The book's goal is to discuss the most current trends in applying natural language processing (NLP) approaches. It makes the case that these areas will continue to develop and merit contributions.

The book focusses on software development that is based on visual modelling, is object-orientated, and is one of the most significant development paradigms today. To reduce issues throughout the documentation process, there are still a few considerations to make. To assist developers in their documentation tasks, a few aids have been developed. To aid with the documentation process, a variety of related tools (such as assistants) may be made using natural language processing (NLP). The book is focused on software development and operation using data mining, informatics, big data analytics, artificial intelligence (AI), machine learning (ML), digital image processing, the Internet of Things (IoT), cloud computing, computer vision, cyber security, Industry 4.0, and health informatics domains.

1

Machine Learning and Artificial Intelligence for Detecting Cyber Security Threats in IoT Environmment

Ravindra Bhardwaj[1]*, Sreenivasulu Gogula[2], Bidisha Bhabani[3], K. Kanagalakshmi[4], Aparajita Mukherjee[5] and D. Vetrithangam[6]

[1]Deparment of Physics and Computer Science, Dayalbagh Educational Institute (Deemed to be University), Agra, Uttar Pradesh, India
[2]Department of CSE (Data Science), Vardhaman College of Engineering, Shamshabad, Hyderabad, India
[3]Department of Computer Science and Engineering, University of Engineering and Management (UEM), New Town, West Bengal, India
[4]Department of Computer Applications, SRM Institute of Science and Technology (Deemed to be University), Trichy, India
[5]Department of Computer Science and Engineering, Institute of Engineering and Management, University of Engineering and Management (UEM), New Town, Kolkata, West Bengal, India
[6]Department of Computer Science & Engineering University, Institute of Engineering, Chandigarh University, Mohali, Punjab, India

Abstract

The Internet of Things (IoT) refers to the increasing connectivity of many human-made entities, such as healthcare systems, smart homes, and smart grids, through the internet. Currently, a vast amount of material and expertise has been widely spread. These networks give rise to several security threats and privacy concerns. Intrusions refer to malevolent and unlawful actions that cause harm to the network. IoT networks are susceptible to a diverse range of security issues due to their widespread presence. Cyber attacks on the IoT architecture can lead to the loss of information or data, as well as the sluggishness of IoT devices. For the past twenty years, an Intrusion Detection System has been utilized to ensure the security of

**Corresponding author*: ravindrabhardwaj2@gmail.com

Rajesh Kumar Chakrawarti, Ranjana Sikarwar, Sanjaya Kumar Sarangi, Samson Arun Raj Albert Raj, Shweta Gupta, Krishnan Sakthidasan Sankaran and Romil Rawat (eds.) Natural Language Processing for Software Engineering, (1–14) © 2025 Scrivener Publishing LLC

data and networks. Conventional intrusion detection technologies are ineffective in detecting security breaches in the Internet of Things (IoT) because of the distinct standards and protocol stacks used in its network. Regularly analyzing the vast amount of data created by IoT is a tough task due to its endless nature. An intrusion detection system (IDS) is employed to safeguard a system or network against unauthorized access by actively monitoring and identifying any potentially malicious or suspicious activities. Machine learning technologies provide robust and efficient approaches for mitigating these distinct hazards. The establishment of a robust machine learning system is the key to acquiring networks that are free from any form of threats.

***Keywords*:** Machine learning, Internet of Things, security, privacy, attacks, vulnerability, intrusions

1.1 Introduction

The use of connected devices made ordinary chores easier and more efficient. They also provide a lot of information that is of great use. Connected automobiles, for example, may be able to take use of services that provide driver assistance. Medical devices give detailed patient records. The unfortunate reality is that a digital assault is possible on any device that is capable of establishing a connection to the internet. In worst case, many of these devices are missing even the most basic safety safeguards. According to the authors of the report, almost all of the data flow associated with the internet of things (98%) is not secured. This information may be obtained by anybody with little effort. To repeat, devices that are connected to the Internet of Things provide fraudsters with an easy target. Not only might their information be stolen, but perhaps other sensitive data as well. Using one of these devices is a frequent strategy used by hackers to gain access to a company's internal network. The sheer number of these devices and the settings they control may be enough to pique the interest of a cyber-attacker [1] as given in Figure 1.1: Increasing Number of DDOS Attacks [Source: Cisco Annual Internet Report 2018-2023] and in Figure 1.2: Threats to Internet of Things.

In a smart environment, any number of items, including databases of user credentials, electronic sensors, CCTV installations, access controls, personal electronic devices, recorded biometrics, and so on, might be the target of an attack. It is essential to protect the confidentiality, integrity, availability, authentication, and authorization features of the IoT architecture from a security point of view [2]. DDoS attacks are becoming more common, and Cisco's Annual Internet Report (2018-2023) White Paper forecasts that the total number of DDoS attacks would more than double

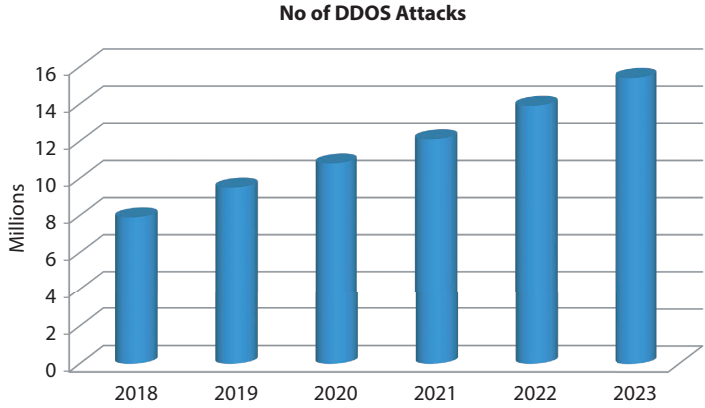


Figure 1.1 Increasing number of DDOS attacks [Source: Cisco Annual Internet Report 2018-2023].

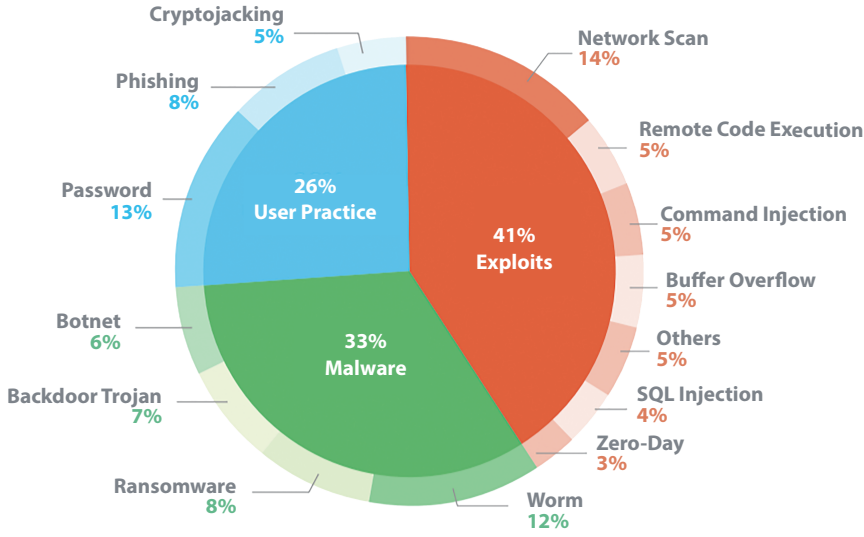


Figure 1.2 Threats to Internet of Things.

from the 7.9 million that were seen in 2018 to anywhere over 15 million by 2023 as shown in Figure 1.1.

According to the survey, 57% of IoT devices that are connected via this insecure traffic are susceptible to medium- to high-severity attacks, making them an easy target for cybercriminals [3]. In addition, the survey found that 41% of attacks target IoT vulnerabilities by scanning them

against publicly available databases of known security flaws. The analysis is shown in Figure 1.2.

According to the Internet of Things Threat Report published by Palo Alto Networks in March 2020, 98% of all traffic from IoT devices is unencrypted, giving attackers a chance to eavesdrop. This network contains sensitive and private information that is easily accessible to attackers, who may then sell the information on the dark web for a profit.

1.2 Need of Vulnerability Identification

Vulnerabilities in IoT network are increasing every year. As shown in Figure 1.3, IoT environment is experiencing, a large number of new vulnerabilities every year. All the Internet of Things applications—smart city, smart farming, smart healthcare, smart transportation, and smart traffic—are experiencing new vulnerabilities and increasing number of attacks every year. Also, vulnerabilities and attacks are increasing every year. Number of vulnerabilities has increased threefold in the last decade and twofold in last five years as represented in Figure 1.3: Number of New Vulnerabilities Identified in IOT [Source- IBM X-Force Threat Intelligence Index 2022].

The process of determining how vulnerable a system is to attack is referred to as a vulnerability scan. This kind of scan is carried out to identify potential entry points into a computer or network so that appropriate preventative measures may be taken. Automated scanning methods check applications to see if they have any security problems to establish whether

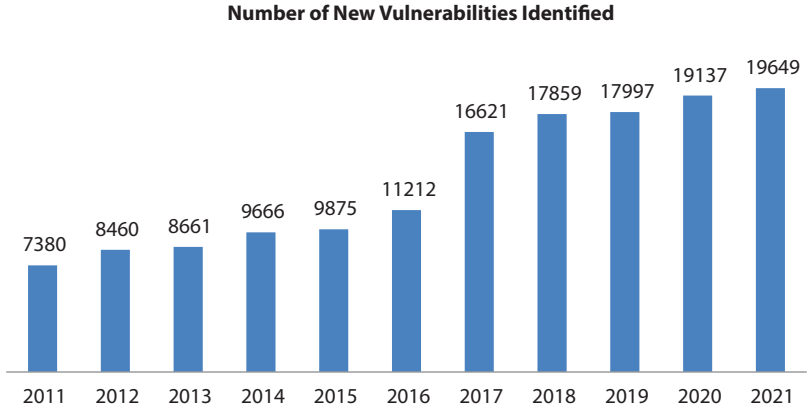


Figure 1.3 Number of new vulnerabilities identified in IoT [Source- IBM X-Force Threat Intelligence Index 2022].

or not there are vulnerabilities in an organization's internal network. Users are spared the time and effort required to carry out hundreds or even thousands of manual tests for each kind of vulnerability since vulnerability scanners automate the process of searching for security issues in a system.

To maintain the integrity of the system's protections, it is essential to assign vulnerabilities a severity ranking before putting into action any remedial procedures. Common Vulnerability Scoring System (CVSS) is a tool that administrators may use to prioritize security problems according to the severity level associated with each fault. The CVSS score of vulnerability is a standard metric that is not developed for unique network architecture. Despite the fact that the frequency and impact of vulnerabilities affect the security risk level of a specific network, the CVSS score of vulnerability is a standard metric. In addition to the severity score, a number of other factors also affect the level of security risk that is posed by the organization's underlying infrastructure. These factors include the age and frequency of vulnerabilities already present in the system, as well as the impact that exploiting vulnerability has on the system. For this reason, it is advised that, when doing risk level calculations, these components, together with the CVSS severity score, be used. This will allow for effective network security risk management.

1.3 Vulnerabilities in IoT Web Applications

The authors of [4] provide a code inspection-based strategy. To identify a number of mistakes hidden inside the process, this method makes use of code inspection. It is said that the offered approach may be used to locate each and every vulnerability in the NVD. Using this classifier might assist in more accurately identifying potential security flaws.

In addition, a web crawler was developed by Guojun and his colleagues [5]. This web spider collects papers that are connected to one another. The TF-IDF is essential to the methodology. Medeiros *et al.* [6] were the ones who first proposed the approach for evaluating the quality of the code. The concepts that underlie data mining are built on this methodology, which acts as the basis for those concepts. New techniques for identifying web server vulnerabilities were developed by [7].

Authors [8] have developed an innovative method for locating vulnerabilities in web applications. In addition to this, static analysis and data mining directly from the source code are used. Researchers [9] came to the conclusion that XML injection is a critical issue that exists in all web applications.

The vast majority of recently published web apps continue to be plagued by XML injection difficulties.

According to research by [10], a large percentage of such norms rely on online application security. Security measures designed to prevent code injection attacks on web applications were the primary focus of these studies. But even if the notion of acceptance is clearly defined and extensively concealed in almost all international standard regulations, the number of assaults is rising because of flaws in the infusion of code. This is the opinion of the developers. To reduce safety gauges, it is crucial to inform engineers and clients about the relevance of these metrics and to urge them to fulfil the standards with meticulous care. The time we waste waiting for this type of instruction and support is just not acceptable.

Authors [11] spoke about the significant factors that are engaged in the life cycle of product innovation. In addition, a number of software engineers have introduced security mechanization tools and processes that can be used at any stage of the software development life cycle (SDLC) to enhance the stability and quality of even the most fundamental digital systems. In addition to this, they requested that all organizations working to improve networks place a higher priority on planning, education, risk assessment, threat modelling, audits of architecture configuration, secure coding, and assessments of data that has been sent and received after it has been processed.

Wang and Reiter [12] developed a method for mitigating denial of service attacks by making use of a website's diagrammatic structure to counter flooding assaults. When visiting the destination website, a valid customer has the opportunity to quickly get a reward URL by clicking on a referral link provided by a reputable source. The proposed paradigm has no requirements in terms of infrastructure, and it does not call for any changes to be made to the code that users use when they access websites. The WRAPS framework, in addition to the intentions that its creator had for it, was provided. Nearly all of the smart assaults on websites recycled old strategies and methods from earlier attacks. There is a wide number of guises under which one may launch an assault against a strategy or an approach. They may also be seen in circumstances that are not related to the web. Attacks on a website's business logic may be harmful to the website itself, but attackers can also utilize websites as a go-between to accomplish their goals.

The SQLProb [13] will remove the user input and check to see whether it complies with the syntactic requirements of the query. This is accomplished by applying the formula that was inherited and then improving it. The SQLProb is a comprehensive discovery approach that does not need

any modifications to be made to either the application or the database. This allows it to avoid the complexity of polluting, learning, and instrumenting code. In addition, neither education nor metadata are required in order to go on with the material's approval procedure.

Authors presented a complete stream-based WS-security handling architecture in their paper [14]. This design improves the level of preparedness in the administration processing and raises the level of resistance to different kinds of DoS assaults. When leaking is used as a strategy, their engine is able to handle standard WS-Security application scenarios.

The author [15] has examined the vast majority of the conventional criteria that are used to judge Web service quality. The majority of the measures, including performance, consistency, adaptability, limit, strength, exception handling, correctness, uprightness, openness, accessibility, interoperability, and security, all fall below the average level.

Hoquea *et al.* [16] took into consideration the activities that may be taken as well as the probable results or degrees of harm. Following that, the designer divides the assaults into a number of distinct categories. They consistently offered a scientific classification of attack equipment to assist in the organization of security specialists. This was done to help in the prevention of potential threats. They delivered a detailed and well-organized examination of existing tools and frameworks that may aid attackers as well as system defenders. Their focus was on tools and frameworks that are available now. The writers have included a description of both the benefits and drawbacks of the tools and frameworks in the event that you are interested in learning more about them.

Binbin Qu *et al.* [17] provided an explanation of the method that lies behind a model design. The construction of a pollutant dependency diagram for the program requires many steps, one of which is a static examination of the program's source code. They employ a limited state automaton to adhere to the attack model while communicating the pollutant string estimate and verifying the robustness of the program's protections for user input. All of this takes place while maintaining the integrity of the attack model. They utilized the framework model for computerized recognition based on the examination of the spoils and placed it into operation.

1.4 Intrusion Detection System

An incursion refers to any malevolent or dubious activity that jeopardizes the security of a computer or network. Intruders may originate from either

internal or external sources. Internal intruders conceal themselves within the targeted network and acquire elevated privileges to deliberately harm the network infrastructure. External intruders surreptitiously extract data from the target network while remaining concealed outside of it. Internal attacks are initiated by nodes that are either malevolent or compromised, whereas external assaults are initiated by entities that are external to the system. An intrusion detection system (IDS) refers to any hardware or software that can identify and alert to potentially malicious activity on a network or computer system. Moreover, it may also be employed to detect any dubious activities or breaches within the system. Typically, when a network or system behaves abnormally, it suggests the occurrence of anything violent, harmful, or illegal. Although the majority of intrusion detection systems (IDS) mostly depend on identifying and reporting anomalies, there are a handful that excel in detecting intrusions that are overlooked by conventional firewalls. In terms of safeguarding the system from harm, intrusion detection systems (IDS) function similarly to firewalls by preventing unauthorized individuals from gaining access.

There are a total of three categories of intrusion detection systems based on the source of data, four groups based on the technique of analysis, and an additional three groups in total.

The Host-Based Intrusion Detection System (HIDS) software is placed on a computer to monitor, evaluate, and gather data on the traffic and suspicious activities of that specific system. In addition, it analyses not just the traffic activity, but also the system calls, file system changes, inter-process communication, and program running on the computer (Zarpelĺo *et al.*, 2017). HIDS utilizes data collected from the operating system and application software to detect suspicious activities. When a host-based intrusion detection system (HIDS) is deployed, it is capable of detecting intrusions solely on the host where it is installed. Installation of HIDS eliminates the need for extra software to identify threats on the system. Intruder detection systems are designed to detect and identify instances of unauthorized access or attacks from within a protected area. The installation cost is substantial due to the requirement of individual Host-based Intrusion Detection Systems (HIDS) for each device as given in Figure 1.4: Host-based IDS.

The Network-Based Intrusion Detection System (NIDS) safeguards network nodes by capturing and scrutinizing all network packets for malicious activities. Figure 1.5 displays the structure of the NIDS. The sensor is strategically positioned in a vulnerable region inside the

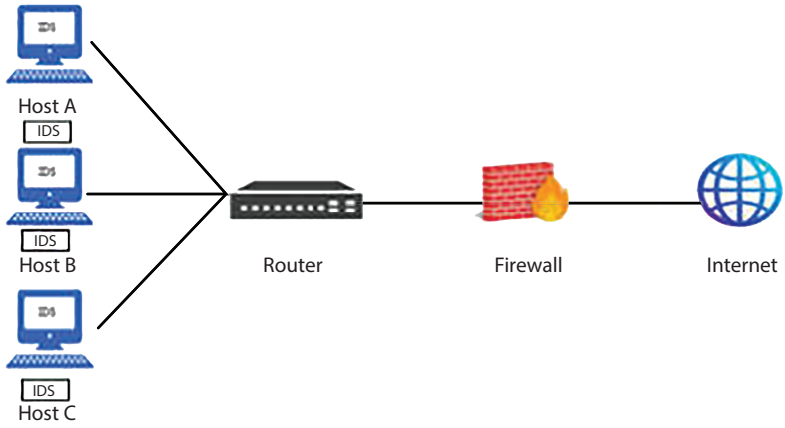


Figure 1.4 Host-based IDS.

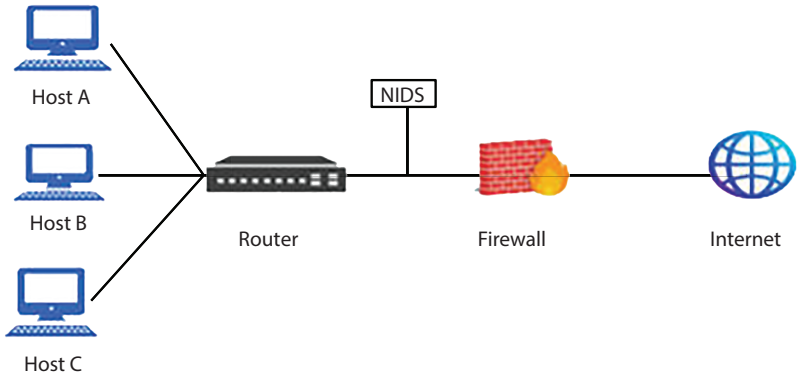


Figure 1.5 Network-based intrusion detection system.

network, bridging the server and the network. The NIDS monitors both incoming and outgoing communications. If the system identifies any network risks, it will need to respond rigorously in order to safeguard itself. One possible course of action is to prohibit network access from the specified IP address, while another alternative is to inform the responsible party through warning notifications. Determining if the NIDS has noticed their potential intrusions might provide a challenge

for a thief. Monitoring extensive networks is under the purview of only a limited number of intrusion detection systems. To mitigate potential security risks, it is imperative to implement scanners, sniffers, and network intrusion detection tools. These measures are necessary to safeguard against various malicious activities such as IP spoofing, DOS assaults, DNS name corruption, man-in-the-middle attacks, and arp cache poisoning. These vulnerabilities arise due to the inherent weaknesses in TCP/IP protocols represented in Figure 1.5 Network-Based Intrusion Detection System.

Hybrid Intrusion Detection Systems (HIDS) integrate the functionalities of several intrusion detection systems to identify and expose intrusions. A hybrid intrusion detection system integrates data from both the network and the host agent or system to create a full overview of the network system. The hybrid technique is the most effective strategy for intrusion detection. Prelude is an example of a hybrid intrusion detection system.

1.5 Machine Learning in Intrusion Detection System

Soft computing makes it possible to build intelligent machines that are able to solve challenging issues that arise in the real world but are beyond the purview of standard mathematical modelling. These kinds of problems cannot be adequately modelled using traditional methods. It has a high tolerance for approximate information, ambiguity, imprecision, and merely a partial view of the environment [18], which enables it to emulate the way individuals form their opinions and make decisions. In this section, we will have a brief discussion on the many different techniques to soft computing that may be used in the process of detecting intrusions.

The genetic algorithm (GA) is a search engine that has been in use since it was conceived in Holland. This search engine is both strong and adaptable. There it first emerged in its current shape for the first time. Because of advances in technology, it is now possible to recreate the natural process of evolution that takes place in uncontrolled environments. The GA may be seen in this way as an example of a global search process that depends on randomness. The concept of "survival of the fittest" is applied by the algorithm to the challenge of developing ever more accurate approximations of a solution to the issue.

The most experienced people in the sector are recruited to teach the next generation, which ultimately results in the development of novel solutions to the issue. If this approach is used, the newly recruited staff members could be better able to address the current challenge [19]. The fitness