different algorithms in the network. This work attempts to growth the nice of provider of network communiqués, making sure errors-unfastened communication through tracking the network. The monitoring process aids to identify the intrusion by the modification of the existing models. The Intrusion Detection System (IDS) attempted [1] using the standardization of protocol structure and evaluate the MAC and IP address, to identify the suspected devices through Genetic functionalities based cross over and mutation values and the modified J48 Decision tree algorithm.

### 1.1 Flow of IDS

Network Intrusion Detection is rapidly becoming a crucial instrument for identifying and analysing potential security risks to a communication network.

It enhances other network protection techniques, consisting of firewalls, by way of presenting facts approximately the frequency and nature of attack. In recent times, the device attacks have increased more than the information attacks. Devices that are dynamically connected in the network can potentially begin and execute information attacks.

These network attacks are identified through the observation of packets and protocols analysis on transaction across the network. Using a combination of the Genetic Algorithm method [2] and an updated version of the J48 decision Tree algorithm, this study employs Network NIDS to standardise protocols and record observations.

### 1.2 Motivation and Problem Statement

The system file is examined by intrusion detection, which notifies an administrator of any modifications to the system binary file. When a system binary file is changed, it means the system has been corrupted, since normal users don't have any good reason to update these files. Consequently, intrusion detection systems play a crucial role in detecting corrupted system files and potential file modifications made by malicious users. In cases where standardising the protocol structure is essential, intrusion detection systems (IDS) might employ additional resources to detect corrupted sources using protocols. Analysis of data packets reveals resource assaults. The methods cannot be processed in a common manner due to the dissimilar sequences of the packet information. It is not possible to anticipate or assess the contents of incoming packets. Many methods for assessing the protocol's ability to identify IDS have been proposed by researchers, however the information sequence between structures and protocols has been identified as a limiting factor in determining effective detection [3].

### 1.3 Types of IDS and its components

In terms of its three primary operational components, the Intrusion Detection System can be described as follows:

Information Sources: The many types of event data utilised to determine if an incursion has occurred in a certain area. them can originate from several levels of the system; the most typical of them are monitoring of the network, hosts, and alertness.

Analysis: The component of intrusion detection systems responsible for gathering and analysing data from data assets, and determining whether certain events signal the presence or absence of intrusions. Misuse detection and anomaly detection are the most common evaluation approaches.

Response: The collection of moves that the system takes as soon as it detects intrusions. Active measures often involve some kind of system-level automatic intervention, while passive measures involve the reporting of IDS results to humans, who are then expected to take action based on those reports.

### 1.4 Anomaly Detection

In a network or host, anomaly detectors look for out-of-the-ordinary activity. Attacks, according to their logic, are different from "normal" (legitimate) activities and can be identified by systems that look for these differences. Detectors like this compile profiles that show how hosts, users, or network connections often act. These profiles are built from records that have been collected over the course of normal functioning. When the observed activity starts to act strangely, the detectors will start collecting event data and apply a cascade of steps to figure out why. Shown in figure 1 is the schematic depiction.

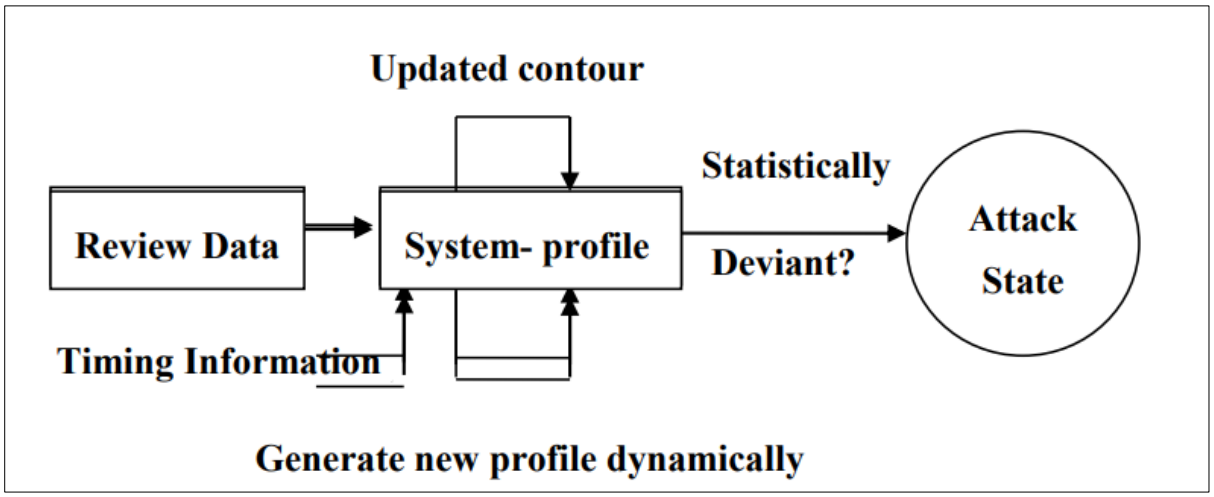


**Figure 1** Anomaly detection System

Anomaly detection employs statistical metrics, rule-based measures, and threshold detection as its tools and methodologies. Due to the fact that individuals and networks might behave in unexpected ways, anomaly detection technologies can generate a large number of false alarms. Widespread "training sets" of system event recordings are often needed for anomaly detection processes to characterise normal behaviour patterns. Since they perform comparable tasks, several vendors classify other pieces of hardware that work in tandem with intrusion detection systems as intrusion detection products [4].

## 2 Literature review

The proliferation of internet use has led to a meteoric rise in cyberattacks targeting banks, government agencies, and energy providers. Businesses with huge websites are targets for hackers and intruders. Some of the ways they can attack include spyware, worms, malware, viruses, and fraudulent logins. In order to prevent harmful assaults and misuse of their networks, organisations require security apps. By identifying and blocking intrusions, intrusion detection systems (IDS) can find and halt data breaches in their tracks. Two forms of intrusion detection include anomaly detection and misuse detection. Anomaly detection is based on actions, whereas misuse detection is based on data or trends [5]. There may be a lot of false positives caused by the current intrusion detection systems' high detection rate. Decrease the occurrence of false positives in an intrusion detection system. Since various machine learning algorithms can uncover useful information from datasets, they are utilised by various IDS. It is possible that these methods can reduce the occurrence of false positives. A lot of machine learning techniques use artificial neural networks. These include intrusion detection systems, genetic algorithms, and association of rules. "Ensemble learning" combines multiple ML techniques [6]. Scientists have found that using an ensemble method for ML helps cut down on false positives.

Industries are becoming more vulnerable as a result of technological progress. The IoT, artificial intelligence (AI), and massive data analysis are the three main pillars that support Industry 4.0. Cyber-attacks can no longer compromise the security of IoT-based monitoring systems for safe computer numerical control devices. The use of the internet of things has also led to significant advancements in deep neural networks (DNN) for detecting intrusions in AGVs [7]. The intricacy of the model, with no obvious indication of the behaviour or explanation of the choice, is the worst downside of any model. Because of this, the focus of research interpretation has shifted to areas like NLP, bioinformatics, robotics, and computer vision. Improved ID transparency, according to the SHapley Additive explanations (SHAP) [5] architecture, is a boon to cybersecurity since it allows for more accurate identification of ID judgements.

We used the CIC-IDS-2017 network traffic data collection for this study. Preprocessing steps included cleaning, normalising, SMOTE, and feature selection, and the data set was then sent on. Following that, numerous ML techniques, including decision tree (DT), random forest (RF), and support vector machine (SVM), were employed to categorise the retrieved feature. To back up the credibility, competence, and dependability of the AI-based IDS solutions, the model correctness was reported and studied using explainable artificial intelligence (XAI). Due to its black-box concept and intended purpose of enhancing accuracy, models are too difficult to understand and explain. XAI [8] provides a way for humans to comprehend and make sense of these outcomes. To make sure everyone is on the same page, we're going to offer a white-box solution that will help everyone comprehend the model and make more accurate predictions [9]. Inconsistencies between the model and the data can be identified and fixed at an early stage of the modelling process.

The results of the experiment proved that XAI methods, like the LIME framework, improved the model's precision and consistency.

Machine learning and artificial intelligence have recently made great strides, mostly due to deep neural networks, which are complicated nonlinear models. Ensemble algorithms and support vector machines are examples of "black box" models that are notoriously difficult to decipher and comprehend. Nonlinear algorithms, those with several parameters, and those undergoing complex transformations often necessitate a large number of samples [10]. Also, it's hard to explain the model's learning process because the training sets are so large. The model's inability to articulate its learning process or identify which data points had a major impact on its output causes human response delays and introduces uncertainty into the model. Also, sensitive but high-profile fields like banking, healthcare, and security have their fair share of ethical dilemmas. Machine learning and artificial intelligence are rapidly becoming key components of defence and security solutions. The responsible and ethical use of AI systems has been the focus of numerous government initiatives aimed at reducing instances of unethical behaviour in this area. As part of its "right of explanation" policy, the European Union allows citizens to ask for evidence in order to contest algorithmic decisions [11]. Another option is to check machine learning systems for prejudice and bias; if found, adjustments may be necessary. The Department of Defence has recognised the reasons for bias. In this case, XAI steps in to explain the black box and show the forecast, which solves the problem. Also, in case you have never heard of XAI before, I'll give you the rundown on how it works. There are three ways to characterise an all-encompassing AI model [12]:

The ability to explain If a learning model can articulate its steps in a straightforward manner, we say that it is explainable. By dissecting training models, this study elucidates their operation. When people's lives are on the line, the intellectual appeal of critical applications utilised in the real world is offset by the gravity of the risks involved. One important aspect of any model is its interpretability, which lets users know how the model works and what conclusions they can take. Also, without user input, a learning model is said to be transparent if it shows signs of being understandable. When a model can be understood by itself, without any external help, we say that it is transparent [13]. One of the most important features of complete learning models is their transparency, which includes their explainability and interpretability. Interpretability has been a hotly contested topic in academia and business, with both camps arguing over which models—ML and DL—are more effective in terms of causation, reliability, and practicality. The words explanation and interpretability are not synonymous, despite the fact that they are defined differently in various works of literature. Mathematical explanations of models are made possible by interpretability, which is defined as the ability for algorithms to explain their judgements and for the inner intricacies of a model to be understood [14]. Interpretability and explainability will be used interchangeably in this context, maintaining generality and following LIME's approach. All AI models should be open and honest about their processes and the reasoning behind their judgements.

However, machine learning mechanisms must be both transparent about their techniques and their functionality in order to withstand adversarial attacks. We can anticipate good AI results, decide whether to trust AI or fully account for human aspects, and handle risks to AI-based systems with the use of an open AI model [15]. The XAI algorithm not only aids the accountable intrusion detection system (RIDS) in producing more precise forecasts, but it also clarifies to the classifiers how to spot particular assaults within a dataset.

Using hybrid intelligent systems, the authors of [16] simulated intrusion detection systems. Their study laid the groundwork for exploring new intrusion detection methods, which they then tested against the industry-standard KDD Cup 99 dataset. Sparse maximum likelihood (SVM) and data temporal correlation (DT) were the main areas of their research. Following this, they created a DT-SVM hybrid model and an ensemble method using DT, SVM, and DT-SVM models as their foundational classifiers. With respect to all classes, the results showed that DT offers either higher or comparable accuracy to Probe, U2R, and R2L. Whether it's for one class or all of them, a hybrid DT-SVM method outperforms direct SVM. When it comes to the Probe and R2L classes, the ensemble method provides the most performance. The ensemble method found that the Probe class was 100% accurate, which means that other classes could be able to do the same with the right base classifiers. The last suggestion was to build an intelligent IDS model with a hierarchical structure that can make the most of the ensemble method and the best individual base classifiers.

To achieve robustness and fairness through federated learning systems, a Ditto architecture was suggested in [17]. According to their statistical analysis, heterogeneous networks are environments where fairness (defined as device-level performance uniformity) and attack resilience (measured in percentages) vie for resources. Our suggestions for dealing with these limitations include the Ditto framework and a scalable solution. In order to find out if Ditto can simultaneously attain robustness and fairness, a class of linear issues was theoretically analysed. In addition to producing more accurate, robust, and fair models when compared to typical robust and fair baselines, they provided empirical evidence that Ditto achieves competitive performance when compared to current customisation methods.

Bringing ML safety to open-world tasks and applications is a complex and exciting endeavour, as discussed in [18]. First, we take a look at the problems with ML in uncontrolled open-world scenarios and compare them to traditional safety criteria in order to determine where ML algorithms fall short in terms of reliability. Identifying errors during runtime constitutes the third group of strategies. In order to accomplish the ML dependability goal of safe design, enhance model performance and robustness, and increase run-time error detection, these three methodologies were utilised. An area of focus in machine learning (ML) research, ML safety seeks to mitigate ML's possible hazards in the long run. Cases where general machine learning capabilities surpass safety or where safety challenges are expected to grow more hard are the primary emphasis for the coming decade.

## 3 Methodology

### 3.1 Genetic approach TO IDS

Intrusion Network monitoring, protocol analysis, and policy standardisation are all part of the detection process, which aims to locate intruders. Genetic Algorithm (GA) is one example of an evolutionary algorithm used in the detection method; it uses the survival of the fittest principle to iteratively improve its approximations to a solution by running them through a population of potential responses. Using operations taken from natural genetics, genetic approach selects individuals based on their health inside the problem area and breeds them together to generate a new set of approximations. Like in nature, this process causes populations of individuals to evolve into something better adapted to their environment than what they were originally composed of.

#### *3.1.1 The genetic approach has four phases, which are described below*

This study uses the 64-byte standardised structured protocol as a starting point for the evolutionary algorithm, which retrieves data packets from the network using the ARP, TCP, and IGMP protocols. An incremental data file of about 5 MB capacity stores all the packets that have been observed. The evaluation method begins with selecting the ARP, TCP, and IGMP protocols from the original population. In order to create the mutation and cross-over capabilities, the 64-byte protocol packet structure is utilised to determine the placement of the device address byte.

The packet is presented as a sequence of values that covers the frame details, IP and MAC address and related information. The specified location of the MAC address value is fetched to process the crossover function. The MAC address is presented as 48 bit binary digits. The Most Significant Bits (16 Bits) and Least Significant Bits (16 Bits) values of the MAC address are exchanged and their offspring are generated.

The generated offspring values are given as an input for the mutation process. The single bit value of offspring is exchanged in the process of mutation. It is possible to change bit values, because of the network packet travel and its sequence rearrangements. This bit exchange process is to identify the manufacture value as per IEEE OUI standards that range from 00-00-00 to FC-FA-F7. The mutation is carried out with 216 combinations from the generated offspring values for the crossover process.

The generated values are evaluated with the existing registered device of the network where the list is maintained by the Current Active Directory List (CADL). If the generated value exists in CADL, then the device is authenticated. If the value does not exist in CADL, then the packet and the corresponding device are suspected. The packets and CADL contain huge data set. To improve the search process, the modified J48 decision tree is executed.

### 3.2 Genetic Algorithm approach for IDS

Many intrusion detection systems have made use of the Genetic Algorithm (GA). University of Texas at Austin's Applied Research Laboratories developed AI rules for intrusion detection systems using a variety of machine learning methods, including GA, decision trees, and finite state machines. An intrusion detection rule based on one network connection and its associated behaviour can be applied to real-time connections. These regulations can be represented in the population as chromosomes. The evaluation criteria are satisfied when the population evolves according to the packets that were communicated. To determine if a network connection and any associated activity constitute possible intrusions, the created rule set can serve as information within the intrusion detection system (IDS). Using artificial intelligence (AI) methods to evolve genetic algorithms, the COAST Laboratory at Purdue University built the IDS with autonomous agents (security sensors). All agents have their own built-in evaluator and are conceptualised as chromosomes.The IDS in the aforementioned methods can be seen as a rule-based system (RBS), and the GA as a means to aid in the RBS's knowledge generation. There is a lack of comprehensive coverage of these concerns in the present GA applications. This demonstrates the potential for modelling network connection information as chromosomes and

how genetic algorithm parameters might be established in this regard. The genetic architecture (GA) is a collection of chromosomes that, over many generations, are passed down in a way that maximises their environmental suitability.

**Table 1** Algorithm for Packet monitoring

***Step 1***: If data packet transfers

Capture the packet information with the frame, network, MAC and Protocol Details;

Construction of structured protocol with 64 byte format

Save the packet with packet address

Else

Observe the network

End if

***Step 2***: Load the saved packet data

***Step 3***: Load the CADL // Current Active Directory List

**Identification of IDS**

***Step 4***: Fetch IP and MAC address

***Step 5***: If the MAC exists in CADL

Update structure protocol

Else

***Step 6*** : Generate Mutation and cross over values

***Step 7*** : If each generated sequence exists in CADL

***Step 8*** : Declare Authenticated Device

Else

***Step 9*** : Declare non-authenticated device.

***Step 10***: Recommended for verification. End if

End if

***Step 11***: Continue for monitoring; go to step 1.

In addition to very little data, the 64-byte protocol packet format includes the port ID, IP addresses of the sending and receiving devices, and media access control (MAC) addresses of the receiving and sending devices. By utilising IP and MAC addresses, the communication process has progressed. It is the goal of the genetic approach's evaluation process to examine the source MAC address through mutation and crossover. The static network is utilised for this purpose. Figure 2 below shows the system's process flow.

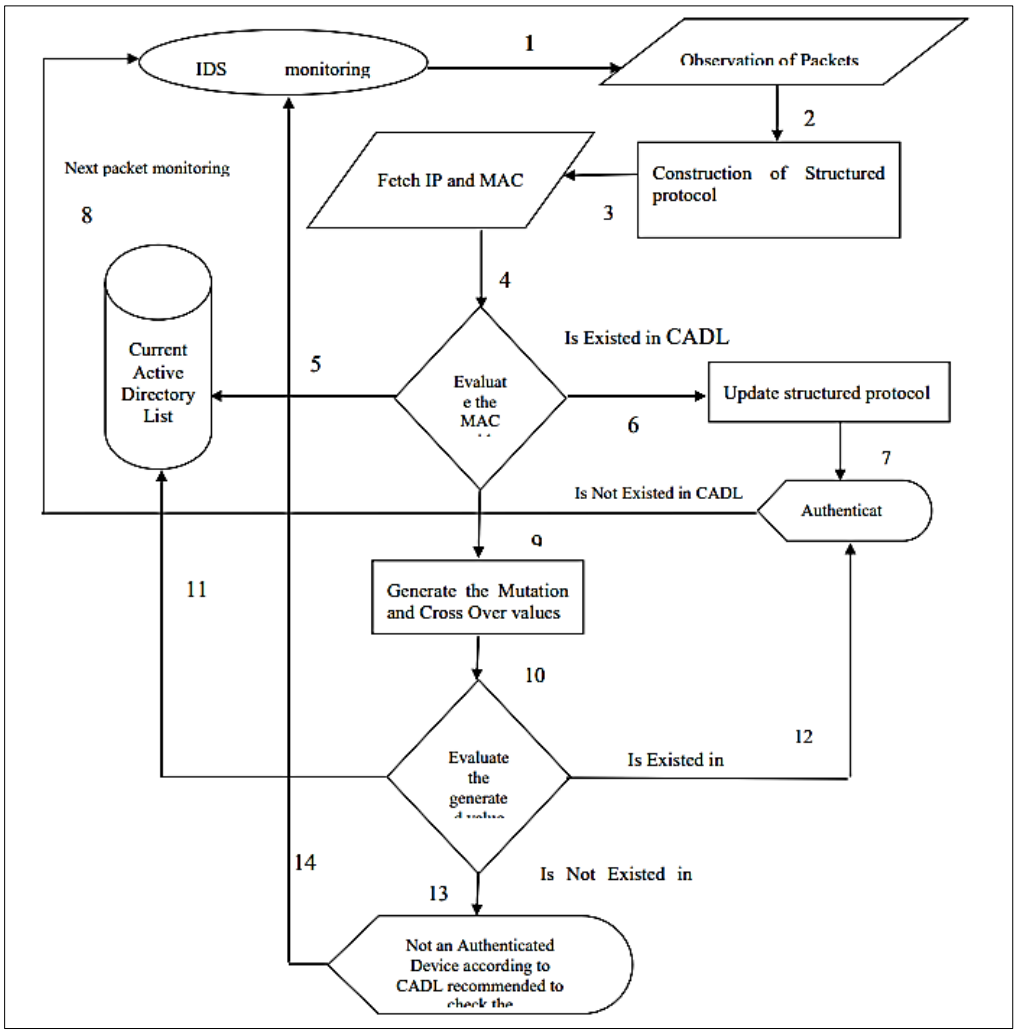


**Figure 2** Process flow of IDS

## 4 Results and discussion

### 4.1 Selection of Population

The functionalities of the genetic approach are implemented on the selected population. In networks, packets represent the initial population selection process. For data transmission across a network to take place, packets must adhere to a specific protocol that includes an initiation, request, and acknowledgement phase. The packets are monitored using the program for monitoring networks. After the packet is collected, the screen is shown in Figure 3.

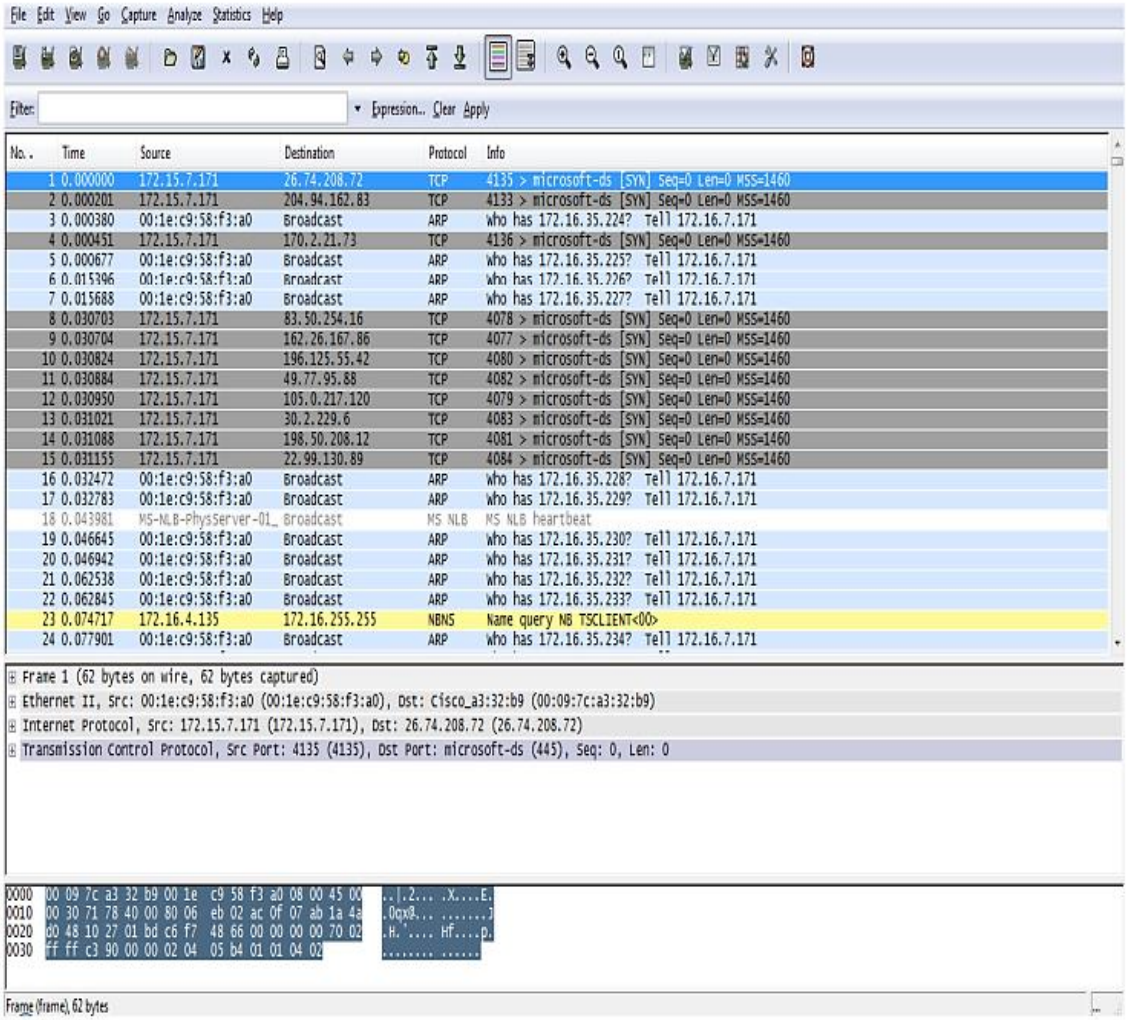


**Figure 3** Total packets captured

Based on the interface settings, the tool monitors data passing from the node or the server. Initial inspection reveals the following details about each packet: number, time, source, destination, protocol, and information. In addition to the fetched protocol details, the preceding figure also shows the frame details and Ethernet information. Below in Figure 4 you can see a summary of each of the chosen periodicals.

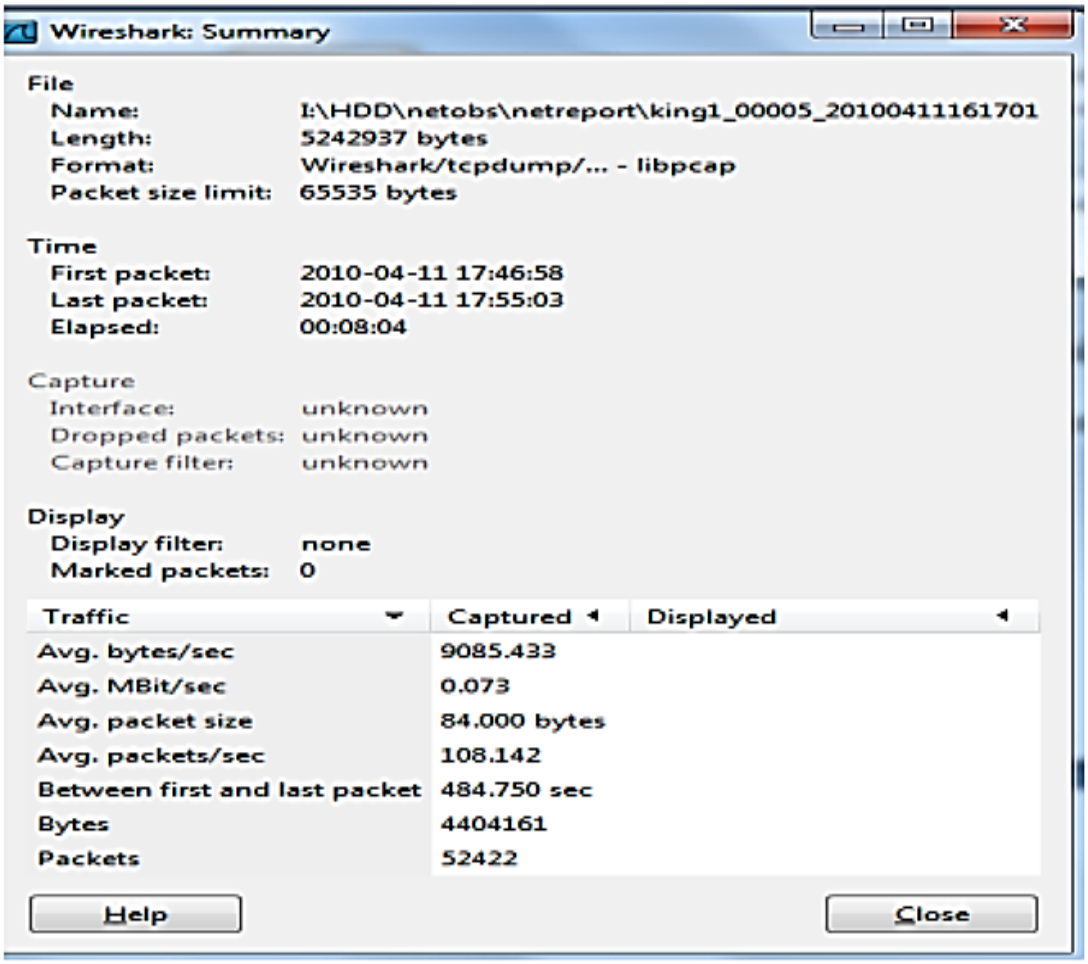


**Figure 4** Summary of Packet.

The number of packets chosen for the IDS population is detailed in this summary. It also computes the overall amount of time spent observing. Also collected and shown below on the screen, as in Figure 5, are the protocols that are offered in the selected population.

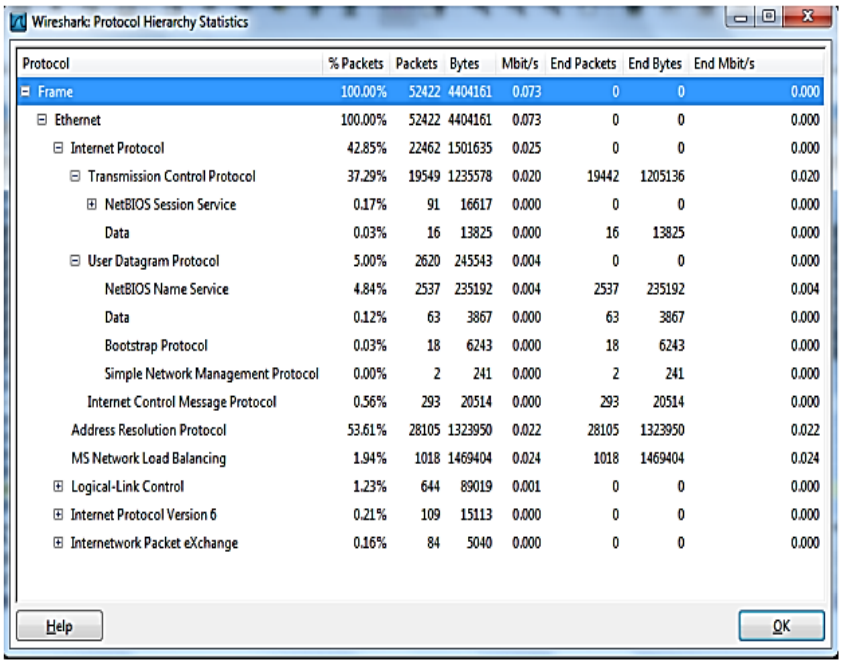


**Figure 5** Structure of Protocols

**Table 2** Intrusion detection results using GA-based packet analysis

| S.No | No. of Packets | Initiated Intrusion | No. of Identified intruder packets |
|---|---|---|---|
| 1 | 52489 | 6652 | 2258 |
| 2 | 52489 | 4109 | 1404 |
| 3 | 49677 | 3881 | 816 |
| 4 | 49677 | 5252 | 1065 |
| 5 | 51307 | 8606 | 2457 |
| 6 | 51307 | 7480 | 2135 |
| 7 | 49783 | 3780 | 1265 |
| 8 | 57758 | 3729 | 1034 |
| 9 | 63384 | 6869 | 1249 |
| 10 | 54761 | 4477 | 1380 |
| Min | 49677.00 | 3729.00 | 816 |
| Max | 63384.00 | 8607.00 | 2457 |
| Aver | 53263.10 | 5483.80 | 1636.5 |

The quantity of packets detected in each assessment for ten files is detailed in table 1 above. A 5 MB file is used to store each evaluation procedure. The packet range in the file is 49677 to 63384. The review process typically uses an average of 53263 packets per file. For every observation, the intrusion starts with an external, unauthorised device. Between 3,729 and 8,607 is the range of the initial intrusion. Based on the results of the study that followed the established process, the table displays the detected intrusion.At least 816 packets were marked as intrusion in the third file, and as many as 2457 packets were marked as incursion in the fifth file. But on average, 1637 packets reveal the IDS process.

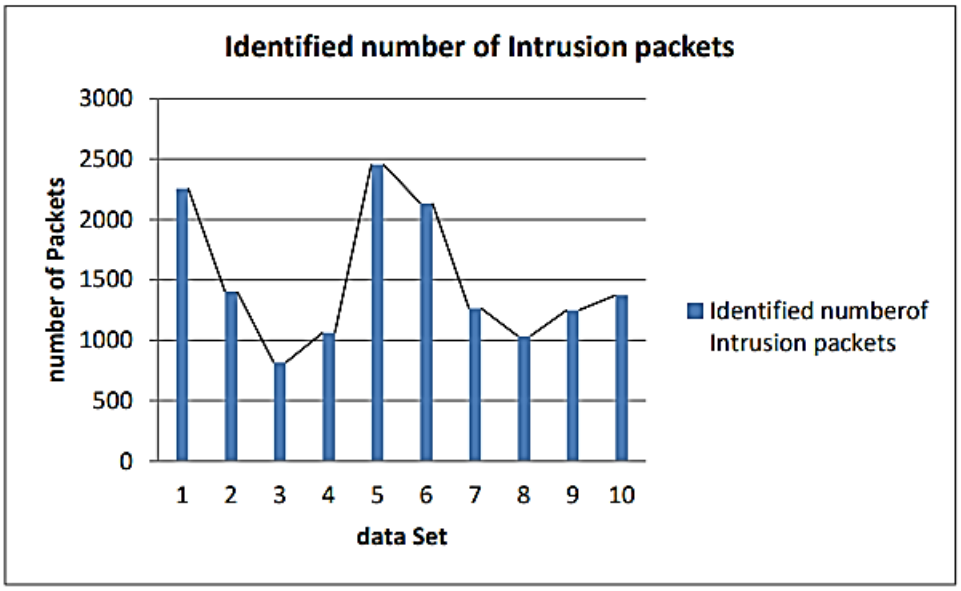


**Figure 6** Graph illustrating the use of GA for intrusion detection.

## 5 Conclusion and future work

The new generation of intrusion detection systems rely on protocol analysis as their primary technological tool for attack detection. These systems use a high level of regularity, matching the reported location of the ARP, TCP, and IGMP protocols, to analyse only the constructive information for intrusion detection. The efficiency of intrusion detection is enhanced since the search space is reduced when only a portion of the payload is searched for, rather than the complete payload, thanks to protocol analysis technology. Protocol abstraction and structured standardisation not only lowered the time required to identify object values, but they also identify values during the detection and search processes from a fixed position.

The developed approach seems to be better in terms of identifying the intruder in less time and identifying instances of attacks that match those that have been profiled before. An intruder, who uses a method of attack that is similar to a previous attack, but who also includes certain unique steps, is much less likely to be identified. The IDSs also require a substantial amount of an initial configuration process to fetch the Current Active Directory List and on-going support to remain effective. Some of the systems that are available currently provide updated services for their profiles, but many still require an on-site security official to keep the profiles current.

### 5.1 Future Enhancement

Future enhancements on this thesis work, lot of continuation could be carried out. The problems and limitations that have existed in IDSs in the past have been compounded with the advent of broadband networks. The overall process is carried out with IPV4 but all the applications and protocols are migrating to IPV6; therefore the research could be carried out and improved further in the new protocol versions. The IDs processes are carried out with the server based approach, but the nodes must be active. The residing programs and automatic tools and antivirus are not able to observe the detection process, because the evaluation is based on the protocol IP and MAC address. The research could be continued to identify the tools, automatic updates and operating system.

## Compliance with ethical standards

*Disclosure of conflict of interest*

No conflict of interest to be disclosed.

## References


[1] Wang, M.; Zheng, K.; Yang, Y.; Wang, X. An explainable machine learning framework for intrusion detection systems. *IEEE Access* 2020, *8*, 73127–73141. [Google Scholar] [CrossRef]

[2] Vigneswaran, R.K.; Vinayakumar, R.; Soman, K.; Poornachandran, P. Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security. In Proceedings of the 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 10–12 July 2018; pp. 1–6. [Google Scholar]

[3] Tran, M.-Q.; Elsisi, M.; Liu, M.-K.; Vu, V.Q.; Mahmoud, K.; Darwish, M.M.F.; Abdelaziz, A.Y.; Lehtonen, M. Reliable deep learning and IoT-based monitoring system for secure computer numerical control machines against cyber-attacks with experimental verification. *IEEE Access* 2022, *10*, 23186–23197. [Google Scholar] [CrossRef]

[4] Elsisi, M.; Tran, M.-Q. Development of an IoT architecture based on a deep neural network against cyber attacks for automated guided vehicles. *Sensors* 2021, *21*, 8467. [Google Scholar] [CrossRef]

[5] Scott, S.-l.l.; Lundberg, M. A unified approach to interpreting model predictions. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30. [Google Scholar]

[6] Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should I trust you?": Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144. [Google Scholar]

[7] Ribeiro, M.T.C. Lime. 2020. Available online: https://github.com/marcotcr/lime (accessed on 17 July 2022).

[8] Sahu, S.K.; Sarangi, S.; Jena, S.K. A detail analysis on intrusion detection datasets. In Proceedings of the 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, India, 21–22 February 2014. [Google Scholar]

[9] AI Explainability 360 (v0.2.0). 2019. Available online: https://github.com/Trusted-AI/AIX360 (accessed on 17 July 2022).

[10] Mane, S.; Rao, D. Explaining network intrusion detection system using explainable AI framework. *arXiv* 2021, arXiv:2103.07110t. [Google Scholar]

[11] Ando, S. Interpreting Random Forests. 2019. Available online: http://blog.datadive.net/interpreting-random-forests/ (accessed on 17 July 2022).

[12] Mohseni, S.; Wang, H.; Yu, Z.; Xiao, C.; Wang, Z.; Yadawa, J. Practical machine learning safety: A survey and primer. *arXiv* 2021, arXiv:2106.04823.

[13] Maseer, Z.K.; Yusof, R.; Bahaman, N.; Mostafa, S.A.; Foozy, C.F.M. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. *IEEE Access* 2021, *9*, 22351–22370. [Google Scholar] [CrossRef]

[14] Laqtib, S.; El Yassini, K.; Hasnaoui, M.L. A technical review and comparative analysis of machine learning techniques for intrusion detection systems in MANET. *Int. J. Electr. Comput. Eng.* 2020, *10*, 2701. [Google Scholar] [CrossRef]

[15] Ahmad, Z.; Shahid Khan, A.; Wai Shiang, C.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transact. Emerg. Telecommun. Technol.* 2021, *32*, e4150

[16] Gamage, S.; Samarabandu, J. Deep learning methods in network intrusion detection: A survey and an objective comparison. *J. Netw. Comput. Appl.* 2020, *169*, 102767. [Google Scholar] [CrossRef]

[17] Mohammadi, M.; Rashid, T.A.; Karim, S.H.; Aldalwie, A.H.M.; Tho, Q.T.; Bidaki, M.; Rahmani, A.M.; Hosseinzadeh, M. A comprehensive survey and taxonomy of the SVM-based intrusion detection systems. *J. Netw. Comput. Appl.* 2021, *178*, 102983

[18] Tjoa, E.; Guan, C. A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transact. Neural Netw. Learn. Syst.* 2020, *32*, 4793–4813