AI-Driven Features for Intrusion Detection and Prevention Using Random Forest

Mohammed B. Al-Doori[1,*], Khattab M. Ali Alheeti[1]

[1]Department of Computer Networking Systems, College of Computer Sciences and Information Technology, University of Anbar, Al Anbar, Ramadi, Iraq

Emails: mohamed.basem.aldouri@gmail.com; co.khattab.alheeti@uoanbar.edu.iq

Abstract

In this research, we investigate sophisticated methods for Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), leveraging AI-based feature optimization and diverse machine learning strategies to bolster network intrusion detection and prevention. The study primarily utilizes the NSL-KDD dataset, an enhanced version of the KDD Cup 1999 dataset, chosen for its realistic portrayal of various attack types and for addressing the shortcomings of the original dataset. The methodology includes AI-based feature optimization using Particle Swarm Optimization and Genetic Algorithm, focusing on maximizing information gain and entropy. This is integrated with the use of Random Forest (RF) to reduce class overlapping, further enhanced by boosting techniques. Grey Wolves Optimization (GWO) alongside Random Forest. This innovative approach, inspired by grey wolf hunting strategies, is employed for classification tasks on the NSL-KDD dataset. The performance metrics for each intrusion class are meticulously evaluated, revealing that the GWO-RF combination achieves an accuracy of 0.94, precision of 0.95, recall of 0.93, and an F1 score of 0.94.

Keywords: Intrusion Detection System; Intrusion Prevention System; Cloud Computing; Anomaly Detection; Deep learning; Software Defined Network

1. Introduction

When discussing cloud computing, the problem of data protection and safety is among the most important considerations. There is a wide variety of software and hardware solutions available to deal with the problems that are related with cloud computing security, such as the installation of firewalls and intrusion detection and prevention systems [1-5]. The firewall was at first considered to be part of broad frameworks; nevertheless, it lacks the power to identify complex assaults, including those that are carried out by an inside party. Utilizing Intrusion Detection and Prevention Systems (IDPS) to bolster the safety of cloud infrastructure and services is an effective method for warding off these kinds of assaults [6-10]. Therefore, the primary contribution that the Intrusion Detection and Prevention System makes is the identification of any unexpected behavior that, as a result of the detection, causes an alert to be generated. After the threat has been identified, the system immediately initiates damage control procedures to protect the cloud infrastructure from further malicious assaults. There are two distinct stages of IDPS, in addition to the hybrid approach to IDPS, which are as follows:

Intrusion detection systems that rely on signatures match specified rules with existing signatures for recognized types of assaults. An alert is created and forwarded to the administrator if a signature is found to be a match. This adaptable method saves new signatures without making any changes to the existing collection. It is only capable of recognizing known external and insider assaults, and it can only identify known attacks. For this approach, software applications like as Snort, Bro, and Suricata are available.

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

An anomaly-based intrusion detection system is one that does not rely on matching signatures or patterns but rather on identifying aberrant user, system, or application activity. When there is a deviation from the usual, an alarm is triggered, which notifies the administrator. Several methods, such as ANN-based, fuzzy-based, or FCM-ANN, are among those that are used.

The hybrid intrusion detection system is capable of detecting both internal and external threats since it combines known/signature-based detection with unknown/anomaly- based detection. Through the use of snort detection, it recognizes data packets and discards them if they are a match for the current database. On the other hand, this strategy is not very prevalent in cloud computing due to the growing significance of the internet in people's everyday lives as well as the emergence of new forms of malicious software, viruses, and hacking methods.

Because breaches in network security may result in large losses for big organizations, network safety is of the utmost importance for service providers. To defend against dangers such as ransomware and hackers, the security sector is continually developing new solutions. Intrusion detection and prevention systems (IDS and IPS) are the primary focus of this thesis since they are responsible for a significant portion of overall network security [11-15]. IDS is designed to identify rogue software's destructive behaviors, while IPS focuses on detecting as well as preventing dangerous behavior. The major emphasis is placed on Snort, a free and open-source intrusion prevention and detection system (IPS and IDS) application that monitors IP-based network packets and does real-time traffic analysis. Snort is a popular tool that may actively block or passively detect a variety of attacks and probes, such as buffer overflows, covert port scanning, web application assaults, and SMB sensing. These are just few of the examples. The investigation is broken up into two parts: a theoretical piece that investigates several IPS and IDS possibilities, and a practical component that aims to build up an IPS that is based on Snort. The theoretical section will be presented first

2. Related Work

A. Cyber Kill-Chain Attacks

The cyber kill-chain model outlines the stages an attacker takes to carry out a covert cyberattack, ranging from reconnaissance to execution. This model helps reduce the risk of an adversary being successful, maximizes resources, and reduces cybersecurity costs. It helps understand the decision- making process from the adversary's perspective, enabling the development of a reliable intrusion detector. The model includes reconnaissance, entry, attack launch, and persistence, with components mapped around the kill chain to detect attackers early and predict their next steps. This helps develop dependable intrusion detectors for cybersecurity.

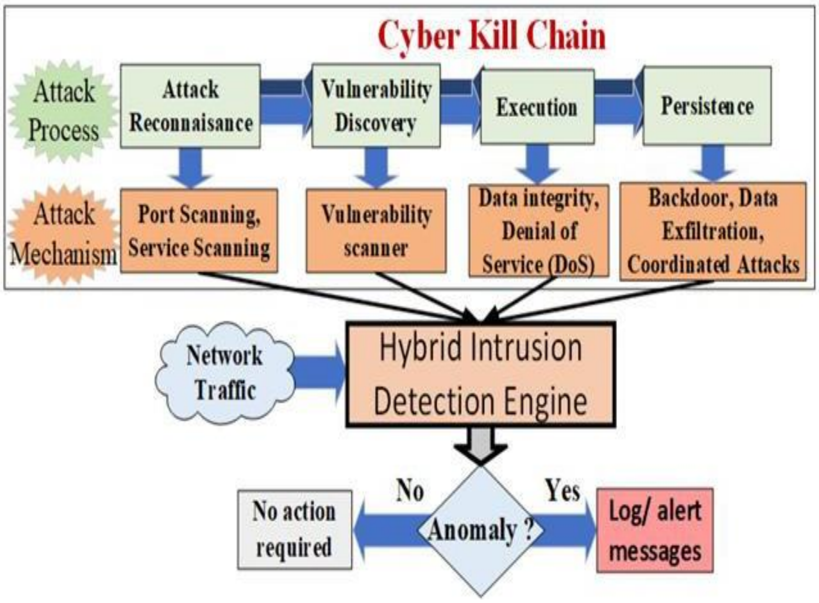

Figure 1. Cyber Kill Chain Model

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

B. MITRE Attacks

As part of the Energy Shield project, which aims to evaluate, monitor, and secure vital infrastructures like the ones used in the Electrical Power and Energy Systems (EPES) sector, a cyber-security cultural framework was created. 65% of IT security experts surveyed by Clarity in 2020 were more worried about cyberattacks on vital infrastructure than data breaches. This emphasizes the need of examining the security of both IT and OT networks in vital facilities. It was determined that the best approach to finding possible external threats was a combination of the MITRE ATT&CK for Enterprise and ICS strategies.

The relationship between various facets of security culture and actual threats and adversarial tactics is the primary subject of this research. The MITRE ATT&CK for Enterprise and ICS hybrid mitigation list may be evaluated by a culture assessment tool that takes into account several layers, dimensions, and domains of cybersecurity. Finding, classifying, and evaluating gaps or vulnerabilities in an organization's security framework (its infrastructure, rules, procedures, and strategies) is made easier with this information.

By presenting a general cyber-security culture framework, this study hopes to close a gap in the existing literature by helping businesses and other organizations assess their current level of cyber security and spot vulnerabilities that may be exploited by hackers. The end objective is to help organizations and individuals learn to anticipate and counteract cyberattacks by gaining insight into the minds and actions of their opponents.

C. IDS/IPS Attacks

The detection rate of a DoS detection system is the proportion of real assaults detected, whereas the false alarm rate is the proportion of benign traffic misidentified as malicious. Because of the risk of disruption to valid traffic or the need for human effort to assess output, a system with a high false alarm rate is unsuitable. Depending on the detection paradigm, there might be a compromise between detection and false alarm rate [20-22]. While signature detection has a low false alarm rate, it is not very effective against assaults that were not expected. A high detection rate comes at the expense of a greater false alarm rate in anomaly detection. Between these two extremes is classification-based detection. In order to take use of the strengths of both signature and anomaly detection, hybrid systems combine the two. A. Models for the Detection of Denial-of-Service Attacks

- Signature detection is a method for DoS detection that creates a repository of previously discovered vulnerabilities or attacks, providing low false alarm rates and forensic information. However, it cannot identify attacks outside its understanding, and manual extraction is time-consuming and labor-intensive.
- Anomaly detection is a method that detects attacks by comparing normal behavior from past or synthesized traffic to a profile. If the traffic deviates from the profile by more than a certain threshold, an alert is produced. Examples include MULTOPS and PAYL.
- Classification-based detection uses machine learning to create classifiers from labelled datasets containing both attack and normal traffic. It combines signature and anomaly detection benefits but may not always signal attacks due to learning phase not covering full network traffic behavior.

D. Mitigation in IDS/IPS

DDoS (distributed denial of service) is a type of network attack that can be effectively stopped by implementing two strategies: Pushback and PRA mitigation. Pushback involves eliminating suspicious traffic at upstream routers closer to the source of the attack, while PRA mitigation uses shuffling the deck to identify possible probe markers. This method involves minor changes to alert data, which can help detect a very small fraction of monitors.

The shuffling process in SPM is stochastic and can be implemented using a pseudo-random function. However, it only allows for the detection of a very small fraction of monitors, such as the destination port value in D-Shield data. The primary benefit of using this strategy is that global statistics will not be affected even if the data is reorganized but not changed. Additionally, the adversary has no way of knowing whether the CIDS has recognized the existence of the PRA or whether the system has triggered protection mechanisms in response to the threat.

Learning algorithms also have vulnerabilities, with the first part being the assumptions made about the training data and the second part coming from retraining procedures. These procedures can be used by the adversary to amplify weak attacks into stronger ones by coordinating over many retraining iterations.
In conclusion, DDoS and PRA mitigation strategies aim to protect networks by identifying potential threats and implementing appropriate mitigation strategies.

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

Table 1: type of mitigation techniques.

Mitigation Technique	PRA Defense Level	CIDS Usability Level
None	○○○○○	●●●●●
Non-public CIDS	●●●●●	○○○○○
Hashing	●●●●○	●○○○○
Encryption	●●●●●	●○○○○
Sampling	●●○○○	●●●○○
Shuffling (SPM)	●●●●○	●●●●○
Noise addition	●○○○○	●●○○○
IP anonymization	●○○○○	●●●○○
Bloom Filters	●●●●○	●○○○○

Algorithms that facilitate learning do so by making assumptions about the nature of the training data and the relevance of hypotheses. Assumptions like this may make systems susceptible to corruption by an adversary. Both the learning model and the training algorithm make assumptions throughout the learning process. Assumptions made by the learning model include that the data is linear, distinguishable, and feature-independent. The assumptions that data and tokens are separate and that only tokens contribute to a message's label are two of the model's weaknesses in Spam Bayes. The linearity assumption of the PCA-based detector states that typical data may be adequately represented by a low-dimensional subspace of the link space. However, this presumption may be broken by a cunning foe.

E. Anomaly Detection

Anomaly detection [26] is an important skill that has applications outside of the realm of security alone. In a broader sense, the term "anomaly detection" refers to any procedure that is used to locate occurrences that are inconsistent with an expectation. To notice early warning signs of system breakdown and so encourage operators to undertake early or preventive checks, anomaly detection may be utilized in instances when the dependability of a system is of the highest relevance. For instance, the power company may be able to save money by repairing electrical power grid anomalies before a power surge affects outages in other system components. When other parts of the system cease working due to a power surge, this kind of damage might occur. Anomaly detection also plays an important role in the fight against fraud. Fraud in the financial industry is often feasible despite the vast number of legitimate transactions taking place, but it may be uncovered via the study of patterns of normal occurrences and the identification of cases of deviation.

1) Anomaly Detection Vs Machine Learning

Anomaly detection and pattern recognition can be confusing, making supervised learning a useful solution for identifying fraudulent credit card transactions. This method can learn from both legitimate and fraudulent transactions and can identify patterns more likely to be present in fraudulent transactions. However, finding a representative pool of positive instances can be challenging. Zero-day attacks, or software vulnerabilities, can lead to server security breaches, making it difficult to construct profiles of intrusion methods. Class imbalance and the lack of frequency make supervised learning more challenging. Anomaly detection is ideal for addressing these issues, as it can help identify patterns and improve security measures.

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

2) Feature Engineering in Anomaly Detection

Anomaly detection is a crucial aspect of machine learning, and feature engineering is essential for identifying anomalies. Online methods require time series data streams, which can be obtained from system monitoring modules or by building own data streams. Web application intrusion detection, network intrusion detection, and host intrusion detection are the three primary areas of concentration. To extract characteristics from each, certain factors and methods are needed.

Accurate feature extraction is crucial for establishing a reliable data source for algorithms, which is applicable not only to host and network anomaly detection but also to use cases such as fraud detection and spotting irregularities in public API requests. You have to do some exploring and testing to find the algorithm that works best for your application. When deciding on a strategy, it is essential to consider the data source and its quality.

The process of outlier detection requires an algorithm that is immune to minute deviations, which could lower the quality of the trained model. It is not always easy to choose the right strategy, as cleaning a dataset can be labor-intensive and sometimes impossible.

This study synthesizes various anomaly detection methods from scientific literature and the business world into a classification system based on the primary tenets of each algorithm. Different categories include statistical metrics, forecasting (supervised machine learning), learning by machine without supervision and techniques based on density. Each category considers a unique strategy for locating irregularities.

3) ML in Anomaly Detection

Machine learning has been successful in recommendation systems, identifying users' latent preferences and driving active demand through collaborative filtering. However, mistakes in anomaly detection can lead to catastrophic flaws and damage trust in the system. A fully automated system is rare, as there is usually a human in the loop to validate the meaning of alerts. Machine learning encounters the semantic gap, which prolongs incident investigation cycles by making it hard to justify the anomalous detection of an event. Knowing how to interpret or describe results is critical in the actual world. Particularly in anomaly detection systems that dynamically modify their decision models, allocating engineering resources to system components capable of producing alarm explanations comprehensible to humans. Given the wide variety of real-world anomalies, it could be more difficult to design an effective evaluation technique for anomaly detection systems than the systems themselves. Even more difficult than building the system is coming up with a trustworthy evaluation plan.

F. Iterative Training

Iterative retraining is a crucial component of learning in hostile environments, as it helps detectors function properly and counter an opponent's ability to learn about and adjust to a detector. Iterative learning also includes a comprehensive theory for combining classifiers, which can be used in highly competitive environments. However, retraining can be exploited by adversaries to increase the effect of weak assaults across multiple iterations if executed incorrectly.

Previous experience can be beneficial for improving classifiers, but it must be used efficiently and safely. The past can influence the development of future models, making small attacks more effective. The behavior of the classifier can also influence the retraining process, making users more vulnerable to attacks.

The learning model benefits from retraining hyper- sphere anomaly detectors, as it requires controlling the number of data points needed to shift the model. However, this approach makes the model less adaptable to regular changes in data, as it relies on irrelevant information.

Researchers Kloft and Laskov expanded on this model by considering more realistic data ageing regulations and environments for attacks. The challenge of retraining models in a risk-free manner remains a question that needs to be addressed.

- Countermeasures focuses on defending against causative attacks by removing malicious data from the training set and hardening the learning algorithm against malicious training data, specifically in SpamBayes and PCA-based detectors.
- Data sanitization is crucial for learning to achieve acceptable performance in various situations. In the development of SpamBayes, the team investigated the Reject on Negative Impact (RONI) defence, which evaluates the impact of adding each training instance and removes instances that negatively affect classification accuracy. The defender trains a classifier on a base training set, adds a malicious candidate

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

instance, and trains a second classifier. If the candidate instance increases classification mistakes, the defender permanently deletes it. The RONI defence successfully neutralizes single threats from dictionaries, resulting in a filter that correctly categorizes 80% of spam and 98% of ham communications.

- Robust Learning focuses on minimizing the impact of a small fraction of deviant training data. In this context, the majority of data comes from known well- behaved models, while a fraction comes from unknown models. To address this issue, a more robust variant of the PCA-based detector is proposed, combining the PCA-Grid algorithm with a robust Laplace cutoff threshold. This method significantly reduces the effect of outliers and rejects poisonous training data.

G. Adverbial Training

Adversarial integrity attacks are a type of attack that can bypass learning mechanisms and exploit blind spots in the learner. These attacks often involve imitating traffic statistical features to conceal intrusions. Researchers have developed techniques to circumvent intrusion detection systems and spam filters, such as polymorphic blending attacks, feature deletion attacks, mimicking attacks, and Bayes vs. Bayes attacks. Near-optimal evasion is a problem that requires partial information of the classifier's basic structure. Machine learning systems are increasingly used in critical systems to protect against adversarial attacks, but their dependability is being scrutinized. Machine learning algorithms are vulnerable to security exploits due to data stationarity, feature independence, and low-level stochasticity. Attack transferability is crucial for practical attacks on machine learning, and black-box models can be used for adversarial evasion attacks [24, 25].

3. Intrusion Detection and Prevention

A. Security Onion and Snort

Security Onion, a Linux distribution, offers a collection of security-oriented software like Snort, Bro, Suricata, Sguil, Squert, Snorby, Xplico, and NetworkMiner. These applications are autonomous decision-making instruments, but not all Linux distributions host them due to their niche nature.

- Snort is a network intrusion detection and prevention system that monitors network traffic in real time, using Sourcefire VRT as a rule set. It is available at no cost, but rulesets can only be distributed through a paid membership [16].

- Suricata is similar in design, speed, and recognition methods, but is recommended to use the same rulesets.

- Sguil is a network monitoring system that uses Snort and Suricata tools to gather and analyse network events, with a graphical client interface for real-time monitoring and report generation.

- Snorby is a web-based tool for monitoring network safety, showing Snort and Suricata event data.

  Protecting your network using the world-famous Snort IDS and IPS solution. It is built upon the following five modules:

- Packet sniffer collects network information using the DAQ library, either in passive mode or from a precompiled file, forwarding it to the decoder.

- Decoder packets: This section focuses on processing packet headers, parsing, detecting abnormalities, and analysing TCP flags, with the TCP/IP decoder stack being the primary area of interest. The decoder uses preprocessors tailored to the third, fourth, and seventh levels of the reference model for a more thorough examination and normalization of data, including frag3, stream5, http inspect, DCE / RPC2, sfPortscan, and various network protocols. The intrusion detection engine consists of two subsystems, with the rule's constructor compiling numerous major rules into a single set for traffic inspection.

- Modulated output: Snort is a widely used intrusion detection system that provides messages in various formats, including file, syslog, ASCII, PCAP, and Unified2. It has been downloaded over 4 million times and is the industry standard. The language used to describe network security policy violations is straightforward and can be easily whipped up in under an hour. However, users can create complex network event handlers using filters, complex queries, rules, thresholds, and slots. Snort's popularity stems from its ability to provide a simplified and efficient message for network security violations.

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

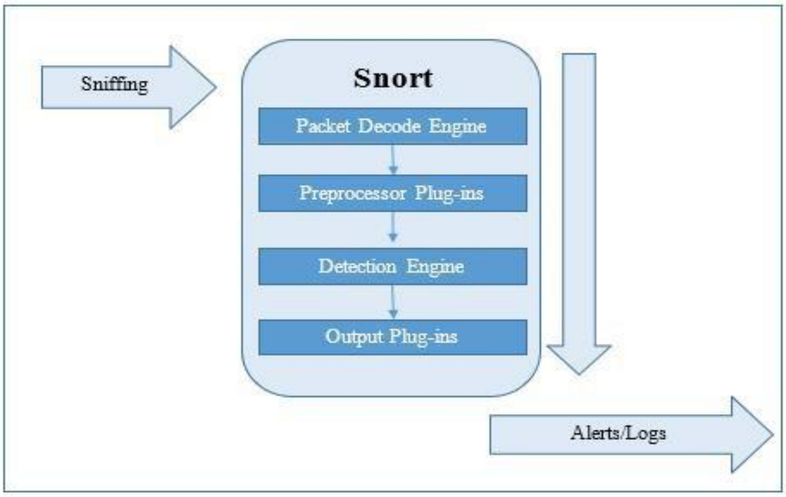

Figure 2. Depicts the basic operating principle

B. IDS and IPS Working Principle

A network-based system that monitors for suspicious behavior and records security-related events is called an intrusion detection system (IDS). A control panel allows users to alter IDS settings, monitor security reports, and evaluate incidents. A sensory subsystem collects events relating to security. Intrusion detection systems may be either host-based (HIDS), protocol-based (PIDS), application protocol-based (APIDS), or network-based (NIDS). In order to identify malicious behavior, NIDS analyzes network traffic and keeps an eye on a certain set of hosts. PIDS keeps tabs on both HTTP and HTTPS traffic, while APDDS analyzes data depending on application protocols. Host intrusion detection systems scour several sources for signs of malicious activity on a host, including application logs, file changes, system calls, and host statements. By combining several intrusion detection and prevention systems, hybrid intrusion detection systems provide a more comprehensive view of a network's security.

1) Passive and active systems of intrusion detection

Alerts are recorded in application log files and sent to consoles or system administrators by passive intrusion detection systems. When an active Intrusion Prevention System (IPS) identifies suspicious behavior, it may terminate the offending connection or reroute traffic via the firewall to stop the attack. When an intrusion detection system (IDS) detects an attack, the intrusion prevention system (IPS) immediately responds to stop the threat. The IPS systems may be categorized in a wide variety of ways. The following is the categorization used by Scarfone and Mell (2007): Network intrusion prevention systems (NIPS) scan network traffic for malicious activity and then shut it down.

Actions in wireless networks are monitored by Wireless Intrusion Prevention Systems (WIPS). Misconfigured wireless access points, man-in-the-middle attacks, and MAC address spoofing are some of the most common security issues it uncovers.
Network behavior analysis (NBA) examines data flows in a network to detect anomalies like denial-of-service (DoS) and distributed denial-of-service (DDoS) assaults.
Host-based intrusion prevention systems (HIPS) are locally installed programs that monitor the host computer for malicious activity.

2) Strategies for Countering Attacks

After an attack has begun, it can be stopped in several ways: by blocking the connection, the user account, the computer network host, the attack itself, or by using a firewall; by altering the settings of communication devices; or by actively suppressing the attack's source. After an attack has been discovered, these techniques may be put into place utilizing network sensors to stop the attackers from continuing their assault. These techniques, however, may fail if the security system has already been breached. In addition, sensor host systems allow for the simultaneous disabling of many user accounts, a process that might be time-consuming and/or need the intervention of a security administrator. The best way to counter an assault is conditional on its details and the strength of its perpetrator.

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

3) Sensors

Host sensors detect remote and local attacks, analyzing packets on different interaction layers to prevent crypto-secure connections and intercepting system calls to applications to block risks, thereby enhancing system security.

C. Comparing IDS with firewall and IPS development

IPS emerged as a combination of firewalls and IDS. Firewalls limit access to host or subnetwork traffic for intrusion prevention but do not monitor inside the network. Modern IPS systems evolved in four directions: inline-IDS development, evolution of firewalls, development of antivirus solutions, and creation from scratch. IPS faced problems such as false positives, reaction automation, and administrative tasks. However, these problems were solved through event correlation systems and next-generation IPS (NGIPS). NGIPS must work in real-time without affecting network activity and act as a single platform that combines previous IPS advantages with new features, such as control and monitoring applications, use of third-party information, and analysis of file contents.

D. Snort (IDS/IPS vendor) alternatives

Top five free enterprise network IDS/IPS tools listed by TechTarget (2016) are:

- Security Onion is a flexible system with collaboration tools, while Suricata is an advanced IPS/IDS system with multitasking, 10Gbit traffic handling, and Snort rules support.
- Suricata creates HTTP traffic inspection tools using the HTP Library, enabling file recovery, content parsing, and identifying URIs, cookies, and user agents, while also decoding IPv6.
- Suricata's IPS utilizes OS batch filters and unified output, enabling analysis using various backends and outputs in PCAP, Syslog, and files.
- Suricata's 7th OSI processing enhances malware detection, parsing protocols even on non-standard ports. Native rules resemble Snort's, but supply is limited and some disabled.
- Suricata utilizes flow bits for rule tracking, detecting malicious traffic across multiple connections. Version 2.1 uses IP Reputation subsystem for quick search and comparison with IP addresses.

E. Bro IDS

Bro is a flexible framework for network intrusion detection and intrusion prevention (IDS/IPS) that uses a scripting language to set monitoring rules for protected objects. It is designed for large traffic volumes and supports separated architecture. Bro is capable of doing in-depth analyses of traffic and enables the use of different protocol analysers in addition to high-level semantic analysis that can be performed even at the application level. Packet capture, the Event Mechanism, and the Policy Script Interpreter are all components of its multi-level, modular framework. Bro is capable of being used for IDS creation as well as traffic analysis. It has the ability to replace Wireshark and provides support for the full-text search engine Elasticsearch. Support for Elasticsearch, although in an experimental form, has been added to more current versions.

F. Snort Rules

The rule header and the rule options are the only two parts of a Snort rule, both of which are contained on a single line. The rule's action, protocol, source and destination IP addresses, netmasks, and source and destination ports may all be found in the rule's header. Warnings and details on which portions of the packet need to be examined may be found in the rule options section. There are just five predefined actions for each rule, but inline mode enables many more, such as alerting, logging, passing, activating, dropping, rejecting, and dropping. You may choose from protocols like TCP, UDP, ICMP, and IP. The direction operator signals the flow of traffic.

The rule's choices are the second portion, and they fall into four broad classes as indicated in Figure 1: general, payload, non-payload, and post-detection.

To stop assaults in the cloud, the IDPS keeps tabs on networks, compiles and analyses data, and recognizes unusual packet behaviour. Known and unknown assault stages are included, as well as a hybrid strategy. Some methods, however, suffer from low precision, lack real-time performance, or are computationally expensive and slow.

G. Network Attack Detection and Prevention

The Next Generation Network (NGN) is a packet-based network that offers various services, including telecommunications, using high-bandwidth transport technologies. It allows consumers to choose from various service providers and supports universal mobility. NGN operates regardless of access technology, including wireless, mobile, and fixed access. Intrusion detection and prevention are key security mechanisms in NGN [17-21]. Intrusion detection is a multi-step process involving hardware, software, and human analysts. Intrusion

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

prevention is a proactive method of network defense that detects and counters attacks before they occur. Customer Access Networks connect users to their service providers, and many network parts use tried- and-true algorithms. However, new wireless technologies like 3G, 4G, 5G, and WiMAX have led to widespread hacking.

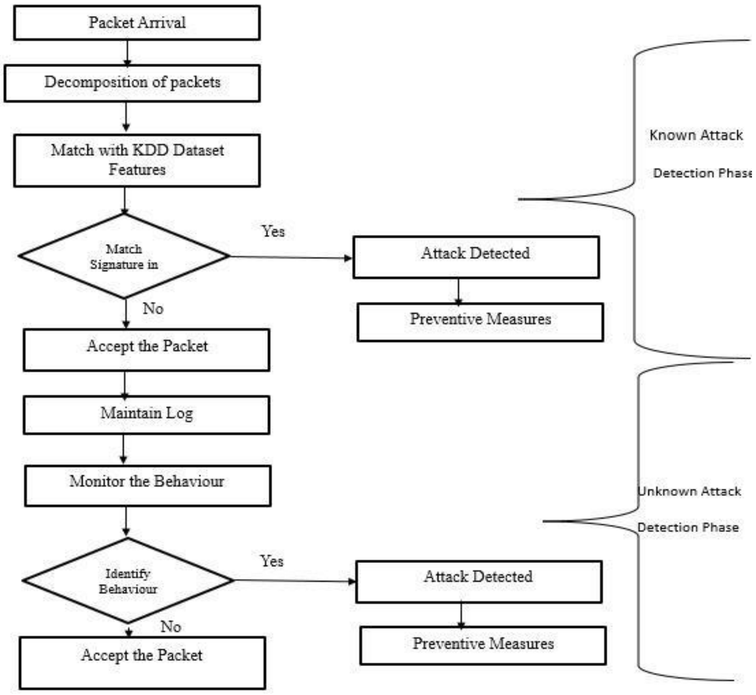

Figure 3. Snort operations

4. Proposed Methodology

A. IDS_IPS

In the implementation part, we proposed three approaches. In all approaches, collect efficient features, optimize features, and use efficient learning approaches. In this document, step by step, give a brief summarization of all approaches and steps.

B. Datasets

1) The KDD Cup 1999 dataset was enhanced to produce the NSL-KDD dataset. Many researchers have used the
2) NSL-KDD dataset to build and test the NIDS problem. All potential forms of attack are represented in the dataset. Due to the issues with the KDD CUP 99 dataset, the NSL-KDD dataset no longer has information that was already there. For NIDS,
3) The evaluation is more effective and accurate because the number of test and training data is more realistic, and the normal-to-abnormal data and UNSBW2.

C. AI based Feature optimization with Random Forest

This first approach is the primary reason for choosing this approach as the next. The main concern improves the features by efficient AI based optimization. In this proposed approach use AI BASED OPTIMIZATION like particle swarm optimization and Genetic algorithm. It optimizes on the basis of information gain and entropy.

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

D. Procedure of the proposed method

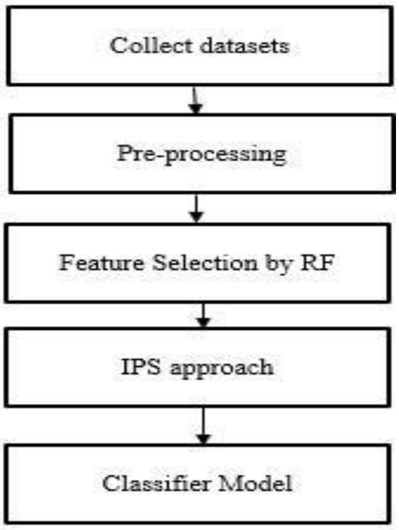

Figure 4. Proposed Flow

Step 1: Collect datasets. In all approaches, use three datasets, which are detailed above.
Step 2: After collecting the dataset, preprocess the features and optimize by AI approaches
Step 3: After optimize weighting of features learn by random forest
Step 4: Random Forest also optimize by boosting approach
Step 5: After detection intrusion apply IPS approach and ignore that instances
Step 6: Make the classifier model and test it, then analyze it for precision, recall, and accuracy.

5. Experiment Results and Analysis

A. Grey wolves Optimization with Random

Grey Wolves Optimization (GWO): GWO is a metaheuristic algorithm that simulates grey wolf hunting behavior and hierarchy, proving effective in finding optimal or near-optimal solutions in optimization tasks. Additionally, Random Forest Classification: It is a robust, accurate, and complex dataset-handling ensemble learning method that combines multiple decision trees to perform classification tasks.

NSL-KDD: NSL-KDD is a benchmark dataset used for evaluating intrusion detection systems, updated from the original KDD Cup 1999, providing a realistic and standardized environment for algorithm development and testing.

Class wise results on NSL-KDD: The NSL-KDD dataset uses random forest classification with IDS/IPS using Grey Wolves Optimization, evaluating performance metrics like accuracy, precision, recall, F1 score, and confusion matrix for each intrusion class, providing insights for improvement.

Table 2: Comparison of Proposed Approach.

Intrusion Type	Precision	Recall	F1 Score
DOS	0.85	0.92	0.88
Probe	0.76	0.81	0.78
R2L	0.92	0.89	0.90
U2R	0.68	0.75	0.71
normal	0.95	0.96	0.95

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

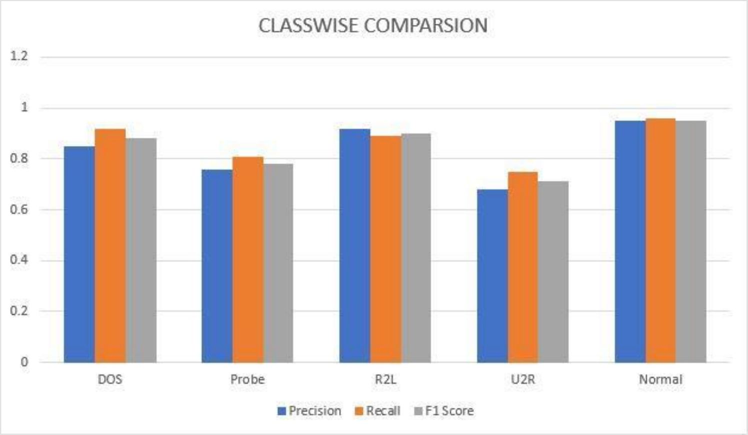

Figure 5. Comparison of proposed approach (GWO-RF)

Table 2 and Figure 5 display precision, recall, and F1 score for intrusion types, providing insights into the IDS/IPS system's performance for each class, enabling a detailed evaluation of its effectiveness in detecting specific intrusion types.

Table 3: comparison of machine learning approach on NSL-KDD dataset

Classifier	Accuracy	Precision	Recall	F1 Score
Random Forest	0.92	0.93	0.91	0.92
Decision Tree	0.88	0.89	0.87	0.88
ANN	0.94	0.95	0.93	0.94
Naive Bayes	0.80	0.82	0.78	0.80

B. Performance of Random Forest, Decision Tree, Artificial Neural Network (ANN), and Naive Bayes classifiers on the NSL-KDD dataset.

Table.3 and figure 6 display the NSL-KDD dataset is compared using various classifiers, with RF and ANN showing the highest accuracy, precision, recall, and F1 scores. DT performs slightly lower but still shows reasonable performance.

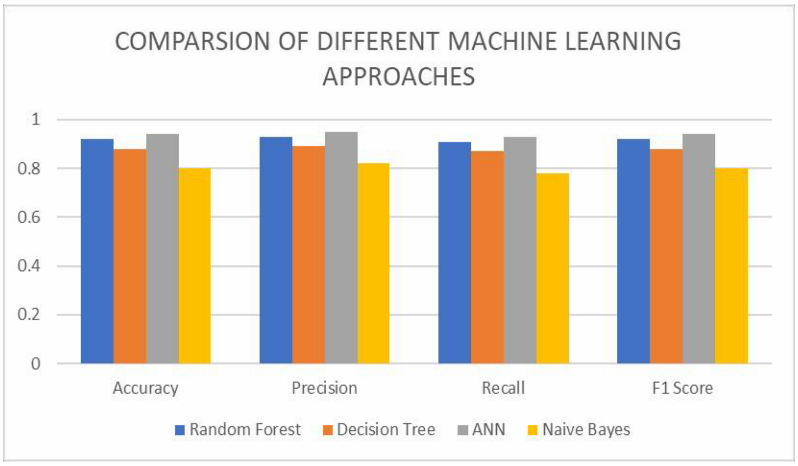

Figure 6. Comparison of Machine Learning Approach on NSL-KDD Dataset

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

The NB classifier has lower accuracy, precision, recall, and F1 score, suggesting weaker performance. However, actual performance may vary based on factors like dataset preprocessing, feature selection, hyperparameter tuning, and cross-validation techniques, and the suitability of each classifier depends on the dataset's characteristics and the classification task.

Table 4: comparison of proposed and machine learning approach

Classifier	Accuracy	Precision	Recall	F1 Score
Random Forest	0.92	0.93	0.91	0.92
GWO + Random Forest	0.94	0.95	0.93	0.94

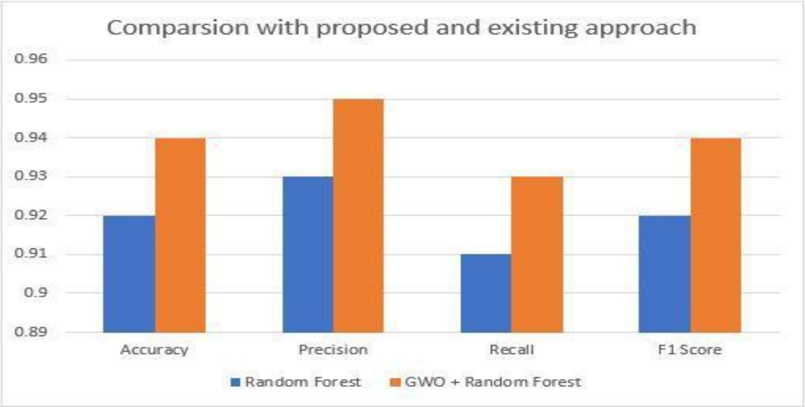

Figure 7. Comparison of Proposed and Machine Learning Approach

The table 4 compares the performance of various classifiers on the NSL-KDD dataset. Random Forest achieves good accuracy, precision, recall, and F1 score on its own. Combining Grey Wolves Optimization (GWO) with Random Forest slightly improves performance across all metrics. GWO optimizes hyperparameters, features, ensemble combination, and fine-tuning parameters, enhancing classification accuracy and overall performance.

C. RESNET50 With SVM

GWO's optimization capabilities combined with Random Forest can overcome biases in parameter settings, improving classification performance. ResNet-50 with SVM results in table format for intrusion detection on NSL-KDD, KDD99, and UNSW-NB15 datasets.

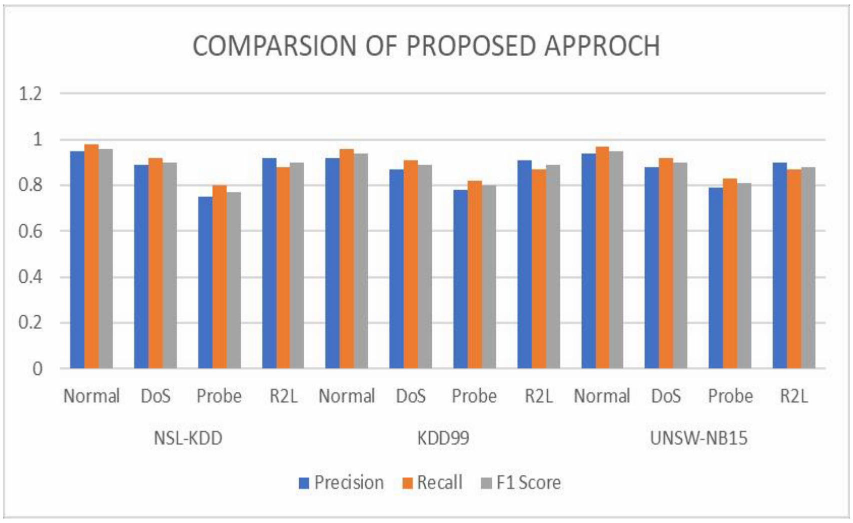

Figure 8. Comparison of Proposed Approach Performance

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

Tables 4 and 8 present performance metrics like precision, recall, and F1 score for intrusion detection using ResNet-50 with SVM on the NSL-KDD dataset. The table includes four main classes: Normal, DoS, Probe, and R2L. Precision measures the proportion of correctly identified intrusion types, with higher precision values indicating fewer instances.

6. Conclusion

In conclusion, this research provides a comprehensive analysis of various methodologies for enhancing Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) through advanced AI-based feature optimization and a spectrum of machine learning techniques. The use of the NSL-KDD dataset proves crucial for its realistic representation of network threats. The study's innovative approach, integrating Particle Swarm Optimization and Genetic Algorithm with Random Forest, demonstrates significant improvements in reducing class overlapping and optimizing feature selection. Particularly noteworthy is the application of Grey Wolves Optimization (GWO) with Random Forest, which shows superior performance, achieving an accuracy of 0.94, precision of 0.95, recall of 0.93, and F1 score of 0.94. Additionally, the comparison of machine learning classifiers like Random Forest, Decision Tree, ANN, and Naive Bayes on the NSL-KDD dataset reveals varied performances, with ANN slightly outperforming others in terms of accuracy and F1 score. The exploration of ResNet-50 combined with SVM also adds a promising dimension to the field of intrusion detection. Overall, this study not only contributes significantly to the realm of network security by enhancing detection and prevention mechanisms but also provides a detailed comparative analysis that aids in selecting the most appropriate methods for specific security scenarios.

Funding: “This research received no external funding”

Conflicts of Interest: “The authors declare no conflict of interest.”

References

[1] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, “A cooperative and hybrid network intrusion detection framework in cloud computing based on Snort and optimized back propagation neural network,” *Procedia Computer Science*, vol. 83, pp. 1200–1206, 2016.

[2] M. A. Hatef, V. Shaker, M. R. Jabbarpour, J. Jung, and H. Zarrabi, “HIDCC: A hybrid intrusion detection approach in cloud computing,” *Concurrency and Computation: Practice and Experience*, vol. 30, no. 3, 2018.

[3] S. Raja and S. Ramaiah, “An efficient fuzzy-based hybrid system to cloud intrusion detection,” *Int. J. Fuzzy Syst.*, vol. 19, no. 1, pp. 62–77, 2017.

[4] J. K. Samriya and N. Kumar, “A novel intrusion detection system using hybrid clustering-optimization approach in cloud computing,” *Materials Today: Proceedings*, 2020.

[5] S. A. R. Shah and B. Issac, “Performance comparison of intrusion detection systems and application of machine learning to Snort system,” *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018.

[6] P. Singh and V. Ranga, “Attack and intrusion detection in cloud computing using an ensemble learning approach,” *Int. J. Inf. Technol. (Singapore)*, 2021.

[7] E. Albin and N. C. Rowe, “A realistic experimental comparison of the Suricata and Snort intrusion-detection systems,” in *Proc. 26th IEEE Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, 2012, pp. 122–127.

[8] A. Alhomoud, R. Munir, J. P. Disso, I. Awan, and A. Al-Dhelaan, “Performance evaluation study of intrusion detection systems,” *Procedia Computer Science*, vol. 5, pp. 173–180, 2011.

[9] G. K. Bada, W. K. Nabare, and D. K. K. Quansah, “Comparative analysis of the performance of network intrusion detection systems: Snort, Suricata and Bro intrusion detection systems in perspective,” *Int. J. Comput. Appl.*, vol. 176, no. 40, pp. 39–44, 2020.

[10] V. Balamurugan and R. Saravanan, “Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation,” *Cluster Computing*, vol. 22, pp. 13027–13039, 2019.

[11] B. M. Beigh and M. A. Peer, “Performance evaluation of different intrusion detection system: An empirical approach,” in *Proc. Int. Conf. Comput. Commun. Informatics (ICCCI)*, 2014.

[12] K. Scarfone and P. Mell, “Guide to intrusion detection and prevention systems (IDPS),” *NIST Special Publication*, vol. 800, p. 94, 2007.

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025

[13] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "Newest collaborative and hybrid network intrusion detection framework based on Suricata and isolation forest algorithm," in *Proc. ACM Int. Conf.*, 2019.

[14] P. Ghosh, S. Shakti, and S. Phadikar, "A cloud intrusion detection system using novel PRFCM clustering and KNN-based Dempster-Shafer rule," *Int. J. Cloud Appl. Comput.*, vol. 6, no. 4, pp. 18–35, 2016.

[15] A. N. Jaber and S. U. Rehman, "FCM–SVM based intrusion detection system for cloud computing environment," *Cluster Computing*, 2020.

[16] R. F. Olanrewaju, B. U. Islam Khan, A. R. Najeeb, K. A. Ku Zahir, and S. Hussain, "Snort-based smart and swift intrusion detection system," *Indian J. Sci. Technol.*, vol. 11, no. 4, pp. 1–9, 2018.

[17] S. Raja and S. Ramaiah, "An efficient fuzzy-based hybrid system to cloud intrusion detection," *Int. J. Fuzzy Syst.*, vol. 19, no. 1, pp. 62–77, 2017.

[18] J. K. Samriya and N. Kumar, "A novel intrusion detection system using hybrid clustering-optimization approach in cloud computing," *Materials Today: Proceedings*, 2020.

[19] K. Sengaphay, S. Saiyod, and N. Benjamas, "Creating Snort-IDS rules for detection behavior using multi-sensors in private cloud," *Lecture Notes in Electrical Engineering*, vol. 376, pp. 589–601, 2016.

[20] P. Singh and V. Ranga, "Attack and intrusion detection in cloud computing using an ensemble learning approach," *Int. J. Inf. Technol. (Singapore)*, 2021.

[21] D. Srilatha and G. K. Shyam, "Cloud-based intrusion detection using kernel fuzzy clustering and optimal type-2 fuzzy neural network," *Cluster Computing*, 2021.

[22] T. Thilagam and R. Aruna, "Intrusion detection for network-based cloud computing by custom RC-NN and optimization," *ICT Express*, 2021.

[23] S. R. K. Tummalapalli and A. S. N. Chakravarthy, "Intrusion detection system for cloud forensics using Bayesian fuzzy clustering and optimization-based SVNN," *Evol. Intell.*, vol. 14, no. 2, pp. 699–709, 2021.

[24] M. A. Jumaah, Y. H. Ali, T. A. Rashid, and S. Vimal, "FOXANN: A method for boosting neural network performance," *J. Soft Comput. Comput. Appl.*, vol. 1, no. 1, Art. no. 1001, 2024.

[25] T. Nsabimana, C. I. Bimenyimana, V. Odumuyiwa, and J. T. Hounsou, "Detection and prevention of criminal attacks in cloud computing using a hybrid intrusion detection system," in *Proc. 3rd Int. Conf. Intell. Human Syst. Integration (IHSI)*, Modena, Italy, 2020, pp. 667–676.

[26] N. Pandeeswari and G. Kumar, "Anomaly detection system in cloud environment using fuzzy clustering based ANN," *Mobile Netw. Appl.*, vol. 21, no. 3, pp. 494–505, 2016.

DOI: https://doi.org/10.54216/JCIM.160101
Received: November 25, 2024 Revised: January 05, 2025 Accepted: February 02, 2025