

INTRUSION AND ANOMALY DETECTION IN CYBER SECURITY USING NATURAL LANGUAGE PROCESSING

K. Kirthika

Asst. Professor
Dept. of AIML CBIT
Kolar, India

kirthikalatha6@gmail.com

Dr. Deepika Lokesh

Professor and HOD Dept. of AIMLCBIT
Kolar, India

ushalokesh@gmail.com

ABSTRACT: The swift acceptance of cloud computing, Internet of Things (IoT), and hyper-connected digital services has tremendously increased the cyber-attack surface thus making the classic perimeter security ineffective against modern threats like zero-day exploits, polymorphic malware, large scale DDoS, and advanced persistent threats (APTs). The currently presented review integrates the fundamentals and progress of intrusion detection systems (IDS) and anomaly detection in cybersecurity, where the spotlight is on the shift from signature- and rule-based methods to ML (Machine Learning) and DL (Deep Learning) based detection. The authors classify intrusions and anomalies, give an overview of typical attack types and their identifiers, and also correlate host-based IDS (HIDS) and network-based IDS (NIDS) on the bases of installation, observability, scalability, and operational trade-offs. Apart from that, the paper also discusses supervised, unsupervised, and representation-learning model (e.g., SVMs, clustering, isolation methods, autoencoders, sequence models, and graph-based learning) approaches in their ML/DL methodologies review and addresses how these models help deal with high dimensional and evolving security telemetry. Benchmark datasets (e.g., KDDCup'99, NSL KDD, CICIDS2017/2018, UNSW-NB15) and standard evaluation metrics (accuracy, precision, recall, F1, ROC-AUC) are highlighted as means to support reproducible assessment. The conclusions at the end of the paper point out the main hurdles in the real-deployment such as encrypted traffic, high-speed processing constraints, alert fatigue, class imbalance, concept drift, and integration challenges, plus they suggest the future research directions including drift aware online learning, encrypted-traffic compatible features, explainable AI, federated learning, and robust detection for edge/IoT environments.

Keywords: *Intrusion Detection Systems (IDS), Anomaly Detection, Cybersecurity, Host-Based IDS (HIDS), Network-Based IDS (NIDS), Machine Learning, Deep Learning, Advanced Persistent Threats (APTs), Internet of Things (IoT), Network Security.*

I. INTRODUCTION

The exponential growth of internet connectivity, cloud computing, Internet of Things (IoT), and digital transformation has fundamentally changed the cybersecurity landscape over the past decade. The attack surface has expanded dramatically, with billions of connected devices creating numerous entry points for malicious actors [1]. According to recent industry reports, global cybercrime damages are projected to reach \$10.5 trillion annually by 2025,

representing a 300% increase from 2015 levels [2]. This alarming trend underscores the critical need for robust intrusion detection and prevention mechanisms. Traditional perimeter-based security measures, which once formed the cornerstone of organizational cybersecurity strategies, are increasingly insufficient against sophisticated attack vectors[3]. Modern cyber threats include zero-day exploits that leverage previously unknown vulnerabilities, advanced persistent threats (APTs) that can remain undetected for months or years, polymorphic malware that constantly changes its signature to evade detection, and coordinated distributed denial-of-service (DDoS) attacks that can overwhelm even well provisioned infrastructure [4]. The sophisticated nature of these attacks demands equally sophisticated detection mechanisms that can adapt to evolving threat landscapes. Intrusion Detection Systems (IDS) serve as a critical component of defense-in-depth security strategies, providing real-time monitoring and analysis of network traffic and system activities. The evolution from signature-based detection to anomaly-based approaches, and subsequently to machine learning and artificial intelligence-driven systems, represents a paradigm shift in how we approach cybersecurity defense [5]. Early IDS implementations relied heavily on manually crafted rules and known attack signatures, which proved effective against known threats but were fundamentally limited in their ability to detect novel attacks. The integration of machine learning and deep learning techniques has opened new possibilities for adaptive, intelligent detection systems that can identify previously unseen attack patterns[6]. Recent years have witnessed remarkable advances in artificial intelligence and machine learning, with deep neural networks achieving unprecedented performance across various domains including computer vision, natural language processing, and time-series analysis [7]. These advances have naturally been applied to cybersecurity, where the volume and complexity of network traffic data make them ideal candidates for deep learning approaches. Modern IDS implementations leverage convolutional neural networks for spatial pattern recognition in network flows, recurrent neural networks for temporal sequence analysis, and transformer architectures for capturing long-range dependencies in traffic patterns [8]. The convergence of big data analytics, cloud computing infrastructure, and advanced machine learning algorithms has created an environment where intelligent, adaptive intrusion detection systems are not just possible but increasingly necessary.

A. Scope and Objectives

This review paper aims to:

- To Provide a comprehensive overview of intrusion and anomaly detection techniques.
- To Analyze the transition from traditional to AI-driven detection systems
- To Examine state-of-the-art machine learning and deep learning approaches
- To Evaluate benchmark datasets and performance metrics
- To Identify challenges, limitations, and future research directions
- To Explore emerging trends including explainable AI, federated learning, and quantum resistant detection.

II. FUNDAMENTALS OF INTRUSION AND ANOMALY DETECTION

A. Definitions and Core Concepts

Intrusion detection is a critical layer of security that identifies malicious actions, policy breaches, or unauthorized access attempts by the continual monitoring and analysis of network traffic or system performance[9]. This concept was born in the 1980s when James Anderson first suggested monitoring auditing trails to spot illicit activities, thereby ushering in the era of what would become an essential part of the contemporary cyber defense system. Gradually over the years, the intrusion detection system has been transformed from primitive rule-based systems that would check audit logs to the up-to-date AI-powered platforms that can simultaneously analyze a thousand of network flows[10]. Anomaly detection is somewhat related to intrusion detection, but the main focus of this technique is to detect those data patterns that are very far from normal behavior[11] . While intrusion detection is aimed at security risks only, anomaly detection can be applied in many other scenarios like fraud detection, systems' health monitoring, and quality control. In

cybersecurity, anomalies can be an indication of security attacks, but they can also be seen as typical but rare activities, system breakdowns, or configuration alterations. The main hurdle in anomaly detection is to properly specify the “normal” behavior and to tell apart innocuous anomalies from real security threats [12]. The attack surface is characterized by the union of every possible location where an unauthorized person can try to get into a system or siphon off data. It is intended to cover also the ingress points that have been created through the use of technologies like the cloud, mobile devices, IoT, and the like, which have drastically widened the attack surface in the modern computing era [13]. security systems' effectiveness hinges on understanding and minimizing the attack surface, in turn, each potential entry point becomes an avenue for the attackers. Therefore, an IDS must ensure that its coverage spans the whole attack surface while at the same time keeping the system performance up to par. The two main metrics that need to be monitored in intrusion detection are the false positive rate (FPR) and false negative rate (FNR). The FPR is the ratio of benign actions mistaken as attacks, causing annoying alerts that might lead to security analysts' alert fatigue [14]. According to research, in poorly configured systems, FPRs can go beyond 99%, which makes the IDS practically useless as analysts will be overwhelmed and start ignoring alerts. On the other hand, the FNR indicates the ratio of real attacks that go unnoticed, perhaps the more serious issue as hidden attacks might do a lot of harm before being discovered. One of the major reasons for the existence of this dilemma between the two metrics is the issue of IDS design and deployment.

B. Taxonomy of Cyber Intrusions and Anomalies

Contemporary cyber intrusions are characterized by a notable variety with respect to their attack vectors, goals, and levels of sophistication. The major categories of network-based attacks that target the communication infrastructure are the following ones. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are aimed at overloading the system resources so that the legitimate users are unable to access the services [15]. Initially, these attacks were purely volumetric floods but later on, they advanced to the application-layer attacks that exploited the weaknesses of protocols and required far fewer resources for making them effective. The last DDoS attacks have been showing a dramatic increase in the volume with some even exceeding the scale of 3 Tbps, which is a clear indication of the continuing evolution of this threat vector[16]. Network scanning and reconnaissance are the first steps of the majority of attacks, where the attackers try to get to know the network infrastructure and its weaknesses systematically by probing it, identifying the running services and the network topology [17]. Although a single port scan might not cause any alarm, the scanning activity patterns are often indicative of the adversarial reconnaissance done prior to more serious attacks. The modern scanning techniques have turned out to be quite stealthy, using distributed sources, slow scan rates, and randomized patterns to hide from the detection. A Man-in-the-Middle (MitM) attacker is able to intercept the communication that is happening between two parties, and in doing so, he might be able to either capture sensitive information or inject malicious content. The establishment of public WiFi networks has significantly contributed to the sophistication of these attacks, and so has the complexity of modern authentication protocols[4]. Web applications are the main victims of both SQL injection and Cross-Site Scripting (XSS) attacks which are implemented through the lack of input validation by executing arbitrary code or altering databases [18]. These attacks are considered to be the most unbreakable vulnerabilities yet remain prevalent as more and more applications with security loopholes are being developed, along with the intricate and difficult modern web frameworks. Injection attacks, which represent the most persistent threats, are always found in the OWASP Top 10 list that continuously highlights the risk. The attackers capture network traffic or impersonate authorized users by forging packet headers in network communications which are considered to be trusting by invaders. Host-based attacks which are aimed at individual systems are the most common and among them are the malware infections of viruses, trojans, ransomware, and other kinds of malicious software [19]. The malware field has changed a lot which led to the rise of ransomware as a particularly profitable attack vector for cybercriminals. Cybercriminals, during the recent ransomware campaigns, have been using the double-extortion tactics

which means that they are threatening the victims not only with the encryption of the data but also with its leak to the public thereby increasing the pressure. Privilege escalation attacks are taking advantage of weaknesses in the system to get through to the elevated access and, in fact, they often are the ones that make it possible for hackers to carry out their multi-stage operations. On the other hand[20], buffer overflow exploits are the ones that overpower the program memory in-order to run arbitrary code, although the modern operating systems' security measures have been quite successful in making similar attacks harder to accomplish[5]. The rootkit installations pose an especially threatening situation since they are regarded as a way of granting an attacker a prolonged, silent, and hidden access to the system that has been compromised while at the same time, tricking the security tools about their presence. Attacks that target the application layer have become more sophisticated and have a larger impact as well. Zero-day exploits take advantage of vulnerabilities that were not previously known, thus giving attackers windows of time before the vulnerabilities are fixed [21]. The zero-day vulnerability trade has really taken off and is now a highly lucrative market with very sophisticated exploits going for six or even seven figures at times. API abuse is becoming a major threat as more and more organizations offer their features through APIs, and if these are not secured properly, they become potential targets for attackers. The session hijacking tactic snatches or guesses valid session tokens enabling the impersonation of authorized users by the attackers without the account passwords. Advanced Persistent Threats (APTs) are the most complex and sophisticated attacks, mainly associated with well-resourced, patient adversaries such as state actors or highly skilled criminal organizations [22]. APTs are characterized by long-term planned attacks on a particular target that may not be detected for months or even years, involving the use of custom made malware, social engineering, and meticulous operational security to elude detection. These multi-phase infiltration campaigns usually consist of the following steps: initial compromise, privilege escalation, lateral movement, data collection, and stealthy exfiltration. The sophistication of APTs very often demands specialized detection techniques that go beyond the traditional IDS capabilities and involve behavioral analysis, threat hunting, and advanced endpoint detection and response (EDR) solutions.

C. Types of intrusions and anomalies

Cybersecurity systems are created for the purpose of safeguarding computer networks, servers, and applications against unauthorized access and harmful actions. As digital infrastructures become more extensive and complicated, the use of traditional security methods is not enough anymore. Intrusions and anomalies are the two basic concepts that are utilized in modern security monitoring[23]. It is very important to know their characteristics, types, and the connection between them in order to come up with efficient intrusion detection and anomaly detection systems.

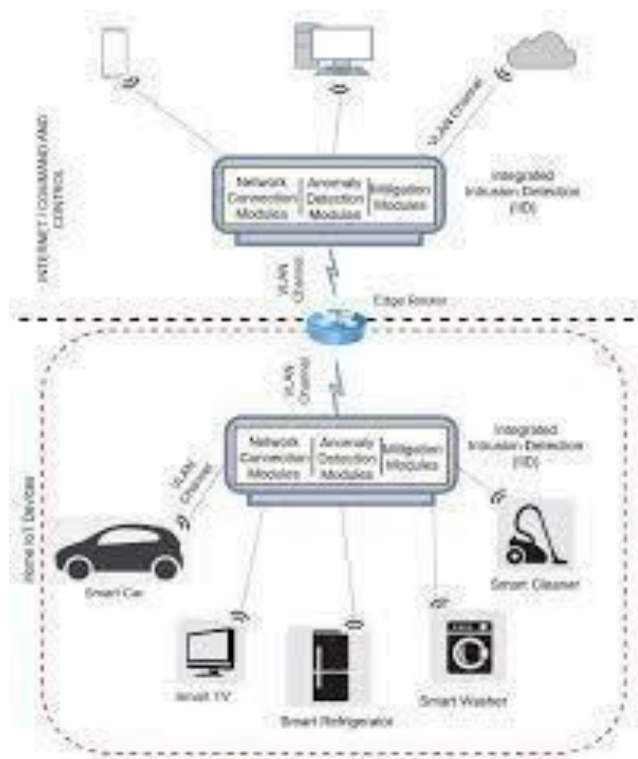


Figure 1 Architecture of a Network- and Host Level Intrusion Detection and Prevention Framework for IoT Environments

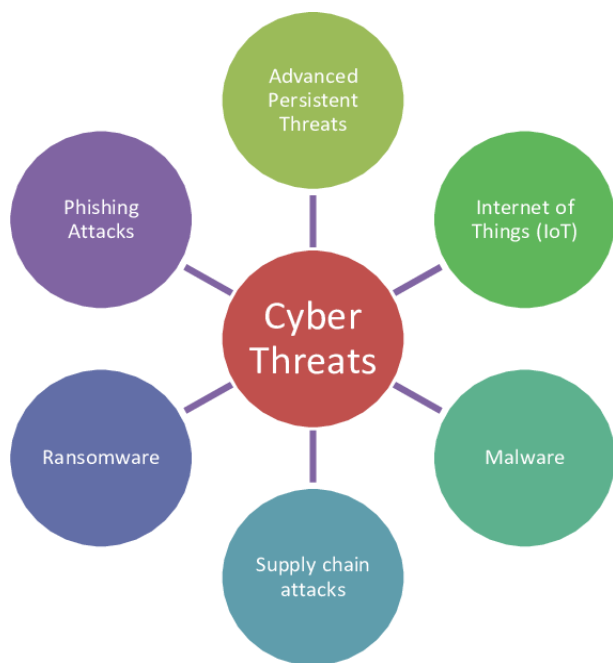


Figure 2 Major Categories of Cyber Threats
Intrusions in Cybersecurity

To invade a system is an attacker's work. It is a planned and unauthorized act with the intention to compromise the confidentiality, integrity, or system availability[24]. Generally, attackers are deliberate in their actions, and besides the main reason of getting unauthorized access, they also cause disruption of the normal system operation, stealing of sensitive information, or taking control of system resources. Let us not forget that intruders, attackers or hackers take advantage of system flaws such as software bugs, weak authenticationsystems, or misconfigurations in networks and device security. The most common way to intrude into a system today is DoS (Denial of Service) or DDoS (Distributed Denial of Service) attacks, where the system is flooded with traffic to the point that none of the legitimate users can access it[25]. Another type is probe or reconnaissance attacks, in which an attacker searches the network for weaknesses by scanning and determining open ports, services running, and vulnerabilities before actually attacking. Although probe attacks might not cause immediate damage to the system, they do create a very good opportunity for future intrusions[26]. R2L (Remote-to-Local) attacks are also a form of intrusion where an outside attacker tries to gain access to the victim's machine by stealing credentials or using weak passwords. An attacker with limited privileges performing U2R (User-to-Root) attack is another example of intrusion when he/she escalates the authority and gains control over the system via administrator rights[27]. Malware, such as viruses, worms, trojans, and ransomware, is also often used to carry out intrusions that disrupt the system, steal data, or allow persistent backdoor access. Web application intrusions, including SQL injection and cross-site scripting, further illustrate how vulnerabilities at the application level can be exploited to compromise the system.

Anomalies in Cybersecurity

Anomalies are defined as significant deviation from the established normal behavior in the system or network activity patterns or behaviors[28]. Anomalies are not always to be considered as intrusions; they might also occur due to genuine but unusual user activities, faults in the system, configuration errors, or sudden changes in workload. On the contrary, a lot of cyber-attacks occur in the form of anomalies which is why anomaly detection has become a significant part of current security systems. The anomalies are usually divided into classes according to their appearance in the data[11]. A point anomaly is when most of the others are relatively the same and there is one observation that is quite different, for instance, an unusual login attempt from an unknown location. In the case of contextual anomalies, the behavior is considered abnormal only in that particular situation, e.g., the system exhibiting high CPU usage during a time that it is usually idle[29]. Collective anomalies are made up of a series of connected events that, when taken together, show an abnormal pattern, like sporadic login failures at a low rate which can be interpreted as a stealthy brute-force attack. Detection of behavioral and system anomalies is especially essential in the spotting of insider threats and advanced persistent threats since these methods are based on user behavior and system performance deviations rather than on attack signatures[30]. Moreover, the anomaly detection procedure is dependent on the learning of normal patterns and therefore it is capable of detecting previously unseen or zero-day attacks.

Relationship Between Intrusions and Anomalies

Though intrusions and anomalies are closely linked, they are different from one another conceptually. Anomalies are sometimes intrusions, but these are always malicious and intentional, while anomalies simply indicate the deviation from the expected behavior. The

open or unauthorized access to a system generates anomalies in network traffic or system activity, but these do not always correspond to the intrusions[31]. This is the main reason why the modern security frameworks often integrate intrusion detection systems, which detect the familiar types of infiltration, and at the same time, they also use anomaly detection systems, which point out the strange behavior[9]. The use of these two methods together not only increases the security of the system but also allows the organization to catch both the known and unknown threats.

III. Traditional Intrusion Detection Systems

A. Host-based Intrusion Detection Systems (HIDS)

Host-based Intrusion Detection Systems (HIDS) signify one of the initial and most concentrated methods of intrusion detection that are still in use. They are capable of operating at the level of individual computers or servers and monitoring and analyzing system-specific activities using the existing behavioral profiles to spot the malicious ones[32]. HIDS implementations, in contrast to network-based systems that check traffic flows, have close access to host internals, therefore, they can recognize the attacks that do not produce any observable network signatures. The close monitoring of HIDS makes them very effective in spotting certain types of threats like insider attacks, privilege escalation attempts and interruption of malware operations that is mainly through local system interactions. HIDS is based on operational monitoring capabilities, which are several key factors. System call monitoring is first and foremost technique where HIDS intercepts and analyzes the sequence and frequency of system calls made by processes [33]. Detection of such anomalous system call patterns is a primary indicator of a potential malicious activity e.g. a web server process attempting to execute shell commands or a user application making unusual kernel level calls. File integrity checking is another method where the detection of unauthorized modifications is through computing and periodically verifying the cryptographic hashes of critical system files, configuration files and binaries[34]. Popular implementations such as Tripwire and AIDE (Advanced Intrusion Detection Environment) keep extensive file integrity databases and notify administrators about any modifications, thus facilitating the discovery of rootkits, trojans, and unauthorized system changes. Log analysis is another paramount HIDS function, which means parsing and analyzing system logs, application logs, security event logs, and audit trails of suspicious activities (Kumar & Singh, 2024). Innovative HIDS solutions rely on advanced correlation algorithms to recognize attack patterns that take multiple steps and might go unnoticed if only individual log entries are considered[35]. On Windows systems, monitoring of the registry is an important source of system visibility; besides, the Windows Registry is a primary storage for frequently targeted by malware system and application settings. So, one can detect the creation, termination, and relationships of processes through auditing, which will help discover suspicious process activities like unanticipated parent-child relationships or processes trying not to be noticed[36] [37]. OSSEC (Open Source HIDS Security) is a case in point that demonstrates today's HIDS capabilities, as it has a wide range of features, namely, log analysis, file integrity monitoring, rootkit detection, and active response mechanisms[38]. Moreover, OSSEC is designed in such a way that its architecture allows for distributed deployment with centralized management, thus making it very appropriate for an enterprise

environment. It uses a combination of signature based rules and anomaly detection techniques, which gives the user the option of how the detection will be done. Wazuh, an OSSEC derivative with extra functionalities, has become very popular in recent times, as it has not only added traditional HIDS functions but also cloud infrastructure monitoring, container security, and regulatory compliance capabilities. The deployment of host intrusion detection systems (HIDS) comes with a variety of benefits, which are proven and substantiated[39]. The powerful monitoring of host-related activities opens the door for the detection of attacks with a very low or no network traffic at all, for example, local privilege escalation, or faking the execution of malware from removable disks (Ali et al., 2024). HIDS can verify the realistic extent of attacks on systems rather than just the detection of attack attempts, thus providing security teams with more actionable intelligence. The independence from network architecture is a big plus for HIDS; it means that its detection capability is not degraded by the use of encrypted communication, network segmentation, or complex routing topologies which can be challenging for network-based detection. In an office with the use of mobile devices or laptops that frequently move between networks, HIDS gives consistent protection irrespective of network connectivity[8]. On the other hand, there are big challenges and limitations that come with HIDS deployments which need to be taken into account very seriously. The use of resources by the monitored hosts could be very high, especially for systems that are engaged in real-time monitoring of every single system call and file access operation[40]. The CPU consumption usually goes up by 3% to 15% depending on how intense the monitoring is and how active the system is. This could be a downside for applications that are highly sensitive to performance. Since the detection is limited to the individual hosts, the HIDS cannot identify distributed attacks, network reconnaissance, or coordinated multi-host intrusions without extensive cross-host correlation[7]. It could be said that the most severe of all the drawbacks would be that if an attacker bestows trust upon the system and gains access to the host with sufficient privileges, he or she could very well disable, corrupt, or even escape the HIDS as it runs with the same basic permissions as other host software[41]. This vulnerability has led to architectural innovations such as protected HIDS implementations running in secure enclaves or hypervisor based monitoring that operates below the host operating system level.

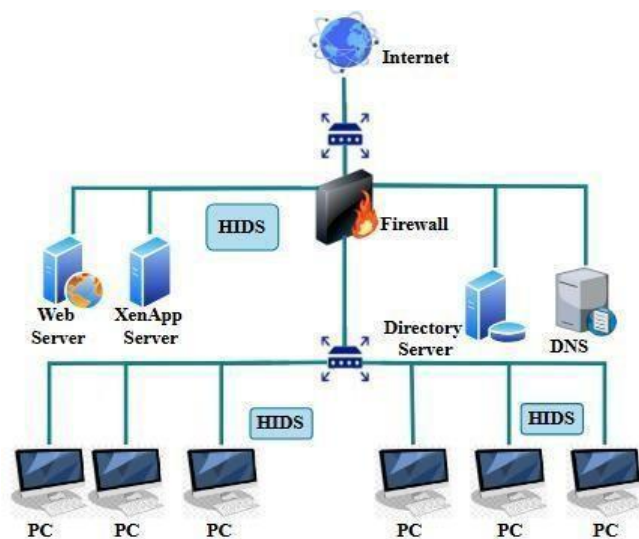


Figure 3 Host-based Intrusion Detection System

B. Network-Based Intrusion Detection Systems (NIDS)

NIDS analyses the whole network traffic and examines it for signs of malicious activities or

incidents, including DoS attacks, port scanning, and user-to-root exploits. It generally performs intrusion detection by examining the IP and transport layer headers of each packet. NIDS employs anomaly and signature-based detection techniques to identify intrusions[33]. The signature detection technique finds connections between the packets that have been collected and the attack signatures that have been acknowledged, while the anomaly detection method contrasts the present user behavior with the past one. Still, it cannot carry out its analysis if the communication is encrypted, and it also cannot detect intrusions within a virtual network controlled by a hypervisor[42]. The Network-Based Intrusion Detection System (NIDS) depicted in Fig. 2 scrutinizes network data for possible intruders or any kind of malicious activity. Such a type of Intrusion Detection System (IDS) is put into practice on a network device, like a firewall or router, and it has to keep an eye on all the traffic passing through that device[43]. NIDS operates by monitoring the traffic and then checking it against a database of known malware signatures.

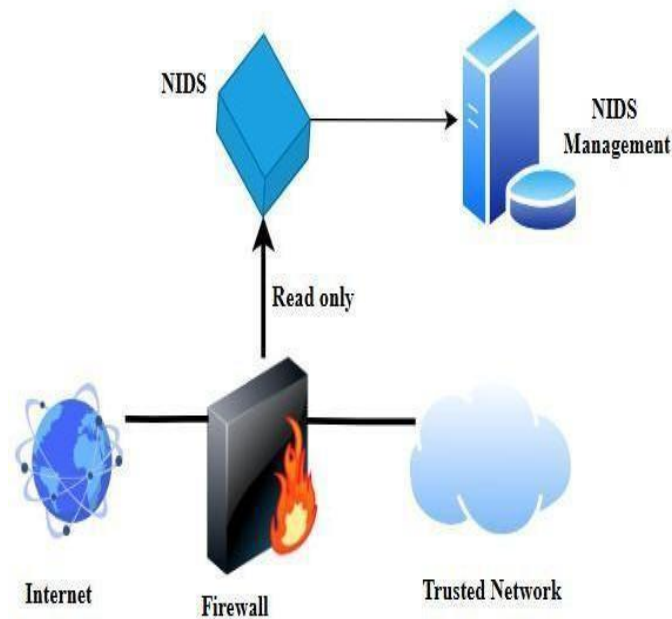


Figure 4 Network-Based Intrusion Detection Systems (NIDS).

Network Intrusion Detection Systems (NIDS) may use anomaly detection techniques to identify irregular behaviours or patterns in the traffic. NIDS can oversee all network traffic, making it very effective in detecting external threats or attacks coming from outside the organisation. Secondly, NIDS is less resource-intensive than HIDS since it does not need installation on each host. NIDS is less vulnerable to assaults since it is implemented on a network device instead of directly on the host machine. Nonetheless, NIDS has several limitations [44]. NIDS may generate several false positives, resulting in laborious and costly investigations. Secondly, NIDS may demonstrate diminished effectiveness in detecting insider threats or threats originating inside the organisation. NIDS may be less effective in identifying new or recognised threats that do not possess a defined attack signature. Prior research [45] indicates that NIDS can effectively provide cloud security. It assists in preventing network attacks and detecting vulnerabilities inside the cloud, such as unsecured networks or outdated software. Another benefit is its ability to protect a company's essential data via its alert system. NIDS enables real-time surveillance, allowing security personnel to swiftly respond to cloud attacks. In [46] presented an anomaly-based network intrusion detection system (NIDS) for the surveillance and analysis of network traffic flow directed at cloud environments. The system employs Support Vector Machine (SVM) as a classifier and binary Particle Swarm Optimisation (BPSO) for the selection of pertinent network characteristics[47]. The system is assessed using the NSL-KDD benchmark dataset, attaining elevated detection accuracy and minimal false alarm rates. Network administrators are alerted to prohibit unauthorized network connections.

C. Limitations and Challenges of Traditional Approaches

Decades of refinements and massive deployments have not been able to help traditional intrusion detection systems; instead, they become even more limited in confrontation with contemporary cyber threats and network environments [48]. For signature-based NIDS, the issue of encrypted traffic has assumed gigantic proportions, as encryption has been adopted at an accelerated rate during the past five years. Besides web traffic, the aforementioned protocols, i.e. email, instant messaging VPN connections, and even IoT devices, are now using encryption thus eliminating traditional case-by-case deep packet inspection (DPI). Although TLS/SSL interception proxies could solve the problem, in addition to significant complexity and computational overhead, they also raise legal issues related to privacy and data protection regulations, and if not implemented properly, they may even create potential security vulnerabilities [49]. The situation keeps getting worse because of the performance challenges at high network speeds that network bandwidth is growing faster than processing capabilities. At NIDS operating at 10Gbps, it has to do the processing of at least 14.8 million packets per second, which is the minimum packet size, and also, it has to do all the required complex pattern matching, protocol analysis, and stateful tracking and this would be done across possibly millions of concurrent connections [50]. The traditional software-based NIDSs find it very difficult to sustain this processing rate without loss of packets, particularly so when using computationally expensive deep packet inspection or anomaly detection algorithms. Hardware acceleration using specialized network processors, FPGAs, or GPUs provides partial solutions but brings about higher costs and complexity. The arrival of 100Gbps and even 400Gbps network links in data center and carrier environments has moved these performance challenges to even greater extremes. The zero-day exploit issue highlights a major drawback of signature-based detection methods. Zero-day vulnerabilities, by definition, do not have signatures until they are found, analyzed, and the signature updates are made and deployed[51]. This results in a critical vulnerability period where organizations are left unprotected. Attackers with advanced skills often use zero day exploits as a means of attack knowing that signature-based defenses will not work. The market for zero-day vulnerabilities has grown a lot and very effective exploits can be sold for hundreds of thousands or even millions of dollars, which shows their importance in getting through the traditional defenses. Moreover, the signature development and deployment process creates delays even after the vulnerability has been found, and during that time, attacks can spread very fast, as illustrated by the WannaCry and NotPetya outbreaks that took advantage of recently disclosed but widely unprotected vulnerabilities[52] [8]. The evasion tactics of advanced hackers have added to the detection capabilities saga, making the conflict an endless one[53]. The practice of splitting packets could result in the attack signatures being separated into several fragments, each of those appearing harmless, and taking advantage of the differences between the reassembly logic used by NIDS and the target system dealing with the fragments. Methods of protocol obfuscation transform the harmful payloads in ways that they cannot be matched but still can be executed by the target applications. Based on timing, the evasion spaces frustrate the detection algorithms that rely on temporal correlation because they attack the components over long periods. Polymorphic and metamorphic malware by changing its code with each infection, impedes detection based on signatures while still.

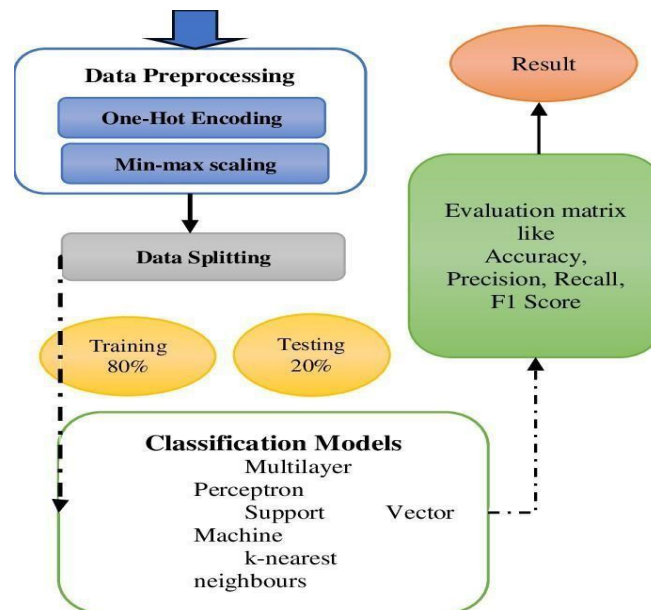


Figure 5 Data Flow Diagram for Traditional intrusion detection system Architecture providing the malicious function[54]. Secure channels set up using legitimate protocols are the hidden paths for the command-and-control traffic that are appearing as normal encrypted sessions to the NIDS. The difficulties of alert management and the challenges of false positives have been described as going beyond just technical issues, and they have been classified as operational problems that are fundamental in security operations centers [55]. When an analyst sees thousands of alerts in a day, the analyst starts getting tired of the alerts and becomes immune to the alarms and may even miss real threats. A common practice among organizations is to increase alert thresholds or to disable a certain noisy rule, hence, they are making detection less effective unintentionally. The financial cost of false positives is huge, considering the time that the analyst spends investigating the innocent alerts, which is a huge opportunity cost. According to the studies, security analysts allocate 40-60% of their time to false positive investigations, which is essentially time lost to threat hunting, security architecture improvements, or real incident response incapacity[56]. The concept drift problem has a major negative impact on the anomaly-based detection systems that are built to operate in dynamic environments (Sarker et al., 2023). Normal behavior never stops changing and it is adaptive to software updates, configuration changes, new applications, user behavior changes, and business process modifications. Anomaly detection models that are trained on the historical data gradually become less and less accurate, therefore they either produce excessive false positives by marking newly normal behaviors as anomalous or they become less sensitive as they adjust to the evolving norms[57]. The management of concept drift involves incessant model retraining and validation, which not only adds operational overhead but also introduces a risk of reduced detection effectiveness during the model updates. If we consider the cloud environments, the problem becomes more complicated as the infrastructures change very fast, in some cases, even automatically, and this can happen several times a day[8]. The traditional IDS deployments' integration and correlation difficulties mirror the modern security infrastructure's complexity.

Organizations usually integrate various security measures such as NIDS, HIDS, firewalls, endpoint protection, SIEM, and intelligence detection platforms which are constantly alerting the staff through different channels and message types[58]. Security monitoring must integrate these different data sources to spot sophisticated and multi-phase attacks that no one machine can discover alone. Meanwhile, the technical and operational integration hurdles have proven to be quite high, with data formats that cannot be reconciled, alert severities that vary, context that is incomplete, and timestamps of alerts coming from uncoordinated sources all adding to the complexity of correlation processes[59]. Moreover, the human element makes it harder for the traditional IDS to be effective, as the

successful launch of such systems call for the skillful tuning of rules, prioritization of alerts, and incident response which often overwhelms many organizations endowed with less specialized personnel.

IV. Machine Learning Approaches for Anomaly Detection

In recent years, ML-based algorithms have gained popularity because to their capacity to handle complicated, high-dimensional data. However, these solutions are not necessarily adequately built to solve the fundamental issue of dealing with imbalanced data [60]. Supervised learning approaches, such as Support Vector Machines (SVMs) [61] and neural networks [62], [63], need labelled datasets with anomalies recognised explicitly during the training process. These models then learn to discriminate between normal and abnormal data points using the labels supplied. However, the availability of labelled data is often a barrier, especially for anomalies, which are inherently uncommon and heterogeneous [11]. Tang et al. [64] proposed a graph neural network (GNN)-based technique for anomaly identification, emphasising how anomalies influence graph spectral features. To overcome the shortage of labelled data, unsupervised learning approaches are commonly employed. These approaches, including k-means clustering and isolation forests, do not need labelled training data. Instead, they discover anomalies by recognising data points that differ considerably from the previously learnt structure of regular data. For example, in clustering, anomalies are defined as points that do not fit into any of the predefined clusters, implying that they are outliers [65]. Other approaches used include autoencoders, which do not need labelled input and instead concentrate on detecting deviations from previously learnt patterns of typical network behaviour [66]. For example, they can accurately recreate typical network traffic, and any large departure in reconstruction error may be identified as an abnormality [67]. Recent advances in anomaly detection have highlighted the need of using multi-dimensional models to improve explainability and resilience. Heine et al. [68] presented the cellwise estimator, which use OLAP cubes to represent typical network behaviour at different aggregation levels and recognises deviations as anomalies. Aiello et al. [69] showed that k-means clustering paired with rule-based learning may identify abnormalities in streaming data, including DNS tunnelling.

V. Deep Learning in Anomaly Detection

Deep learning in anomaly detection focuses on learning the normal structure of data so that deviations can be flagged as suspicious, faulty, or rare events, and it has become popular because modern anomalies often occur in high-dimensional, noisy, non-linear, and time evolving environments where hand-crafted rules and shallow models fail[63]. In practice, deep models automatically learn hierarchical feature representations from raw inputs—network flows, system logs, IoT sensor streams, transaction records, audio signals, images, or video— so they can separate subtle abnormal patterns from complex normal behavior without heavy feature engineering[70]. One typical approach is to simply train the model using mostly normal data through unsupervised or self-supervised means (as there are few labeled anomalies), and subsequently, the model will give an anomaly score, depending on how badly it can explain the new data sample: autoencoders (along with their different versions denoising, sparse, or stacked AEs) have been shown to be quite effective; that is, they are capable of very low-error reconstruction for normal inputs and at the same time, they yield higher error for outliers; on the other hand[71], VAEs and other probabilistic generative models go one step further and provide an effective method for learning the latent distributions, thus the anomalies show up as low-likelihood samples or inconsistent latent codes; the use of sequence models such as LSTM/GRU-based predictors and Transformer architectures for detecting anomalies in time series involves the learning of temporal dependencies and consequently, such points or segments with large prediction error or attention-inconsistency are flagged; on the other hand[72], CNN-based methods involve the capturing of spatial patterns for visual defect detection and medical imaging by measuring feature distance, reconstruction quality, or segmentation inconsistency; GAN-based

pipelines, on the other hand, try to be like normal samples and use the failure in reconstruction or latent inversion to treat it as an abnormality signal, however, they can be more difficult than others to achieve stable and proper training; and deep one-class methods like Deep SVDD compress normal data to small latent areas thus, anomalies are present far from the learned center; whereas, graph neural networks find structural anomalies in relational settings (e.g., fraud rings, unusual host-IP-service interactions) by creating embeddings that are aware of neighborhoods and topologies[73]. In practical applications, powerful systems usually blend several scoring signals—reconstruction error, latent distance, and uncertainty—together as an ensemble to cut down on false positives [53]. . They are also depending on thresholds, which at times are adaptive, because the anomaly rates are extremely imbalanced and what is “normal” can change over time as a result of seasonality, configuration alterations, user behavior shifts, or the development of attack tactics. Getting rid of contamination of training data (anomalies inside “normal” training sets), monitoring of concept drift, limited interpretability (stakeholders want reasons, not only scores), and domain shift across devices/sites/cameras are the practical challenges that deep anomaly detection has to face despite the high accuracy in benchmarks[74]. Therefore, the best practice is to combine deep models with strong preprocessing, monitoring of drift, retraining every so often, provision of interpretability (using feature attribution, attention maps, Grad-CAM for vision, etc.), and sometimes hybrid rule+DL pipelines to enhance reliability in environments that are sensitive to safety or security.

VI. Benchmark Datasets and Evaluation Metrics

A. Datasets

The first phase in developing a proficient intrusion detection system is to choose a suitable dataset. To mimic the types of records seen by the model in the actual world, the dataset must contain both benign and malicious entries. This section delineates information on the esteemed datasets[75]. The most often used datasets include: KDD Cup 99, NSLKDD, CSE-CIC IDS2018, ISCXIDS2012, UNSW-NB15, CICIDS2017, and CIDDS. Figure 4 depicts the datasets used in the analysed procedures.

- KDD CUP

The intrusion detection system research dataset KDDCUP, [76] is often utilized. The KDD CUP dataset is used to develop and test ML models for recognizing disruptions and cyberattacks in cloud-based interruption detection. Developing algorithms that filter network data and identify unusual or spiteful occurrences is critical for cloud security. The KDD CUP 99 standard dataset has roughly 5 million raw data entries. Approximately 80 percent of this collection contains assault data. The dataset is divided into two primary categories: assault and routine [77].”

- NSL-KDD

The researchers that published their results aimed to rectify the deficiencies of KDD-99 by creating NSL-KDD [78]. This resampling of KDD99 prioritizes the occurrences of predictions made by the trained classifier on the foundational KDD-99 dataset. The dataset continues to exhibit problems, as acknowledged by the researchers, including the underrepresentation of minimum footprint assaults.

- CSE-CIC-IDS2018

CSE-CIC-IDS2018 dataset from the Canadian Institute for Cybersecurity (CIC) pertaining to cyber defence on AWS. Global security testing and malware mitigation use datasets from CIC and ISCX[79]. The latest dataset comprises seven distinct attack scenarios: brute force, heart bleed, botnet, DDoS, DoS, internet assaults, and internal network penetration.”

- ISCXIDS2012

ISCXIDS2012 is a dataset created using the profiling concept, including descriptions of intrusions and abstract models of distribution for lower-level network components, protocols, and applications. The profiles are used to replicate user behaviour. These profiles are used to establish a database in the necessary test environment[80]. Multiple scenarios for multi-stage attacks are used to generate the anomalous segment of the dataset.”

- CICIDS2017

The CICIDS2017 dataset contains many entries for security attacks that replicate genuine traffic data. It produces authentic innocuous traffic while analysing human behaviour with B-Profile technology. This dataset incorporates the behaviour of 25 users regarding email protocols, HTTPS, HTTP, SSH, and FTP[81]. Furthermore, it covers documentation about security concerns such as DDoS, Heartbleed, web attacks, brute force SSH, and brute force FTP.”

- UNSW-NB15

The dataset's anomalous and normal network traffic traces are generated using a tool called IXIA Perfect Storm, which is used for the analysis of Network Intrusion Detection Systems (NIDSs). The IXIA application can simulate nine distinct types of security assaults and use recently updated attack data from a website that provides information on security vulnerabilities [78].

B. Performance Metrics

The efficacy of the hybrid detection-mitigation system is assessed utilising common classification criteria, including Accuracy, Precision, Recall, F1-score, and Receiver Operating Characteristic Area Under the Curve (ROC-AUC). These are expressed formally as follows: Accuracy: Accuracy is the most straightforward method to gauge how often a classifier produces accurate predictions. An alternative way to interpret this would be to divide the proportion of correctly predicted favorable results to all forecasts.

$$Accuracy = (TP + FP) / (TP + FP + TN + FN)$$

Precision: As opposed to this ratio plus one minus from it, or $(1 - precision)$, which displays the proportion of false negatives, $1/Precision$ produces recall.

$$Precision = (TP) / (TP + FP)$$

Recall: In contrast, there exist what are known as false negatives with regard to true negatives.

$$Recall = (TP) / (TP + FN)$$

F1-Score: It is calculated utilizing the harmonic average of the accuracy and recall scores.

$$F1-Score = (2 * Precision * Recall) / Precision + Recall$$

Where, TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

VII. CHALLENGES

The advancement of intrusion detection algorithms in contemporary network systems primarily encounters four challenges are as follows [82]:

A. Inaccessible System Datasets

The majority of suggested methodologies in the literature fail to detect zero-day attacks due to insufficient training on a varied array of attack types and patterns. When it comes to creating a successful IDS model, it is necessary to put it through a testing and validation cycle on datasets that consist of both old and new threats. By applying a wide-ranging set of attack definitions to the dataset, the model will be able to reveal new patterns, thus, making it stronger in resisting sophisticated intrusions of all kinds[83]. However, the creation of such databases requires a lot of money and is demanding in terms of resources as well as the need for a high level of expertise. Therefore, one of the major hurdles in IDS research is the creation of modern datasets that cover a wide variety of attack incidents.

Databases need to be updated regularly to capture the most recent intrusion cases, and this information should be made available to the general public to aid the academic community[84] [85].

B. Dataset Imbalances Reduce Detection Accuracy

Low-frequency assaults in datasets usually consist of very few samples and thus, during training, models are not able to learn their features well, which in turn, lowers the detection accuracy. This also indicates that models are not effective enough to detect and classify new or unknown low-frequency assaults. A decrease in generalization ability leads to the situation where models are more likely to miss or misinterpret low-frequency assaults in real-world applications, which directly affects the detection rates[86]. Moreover, traditional loss functions (like cross-entropy loss) do not cope well with the scarcity of low-frequency samples in unbalanced datasets. Besides, the dominance of high-frequency data usually results in the loss function being skewed during optimization, which in turn causes underestimation of the contribution of low-frequency samples. Consequently, the model's optimization process is increasingly biased towards high-frequency samples, and the misclassification loss incurred by low-frequency attack samples is not adequately addressed[84] [85]. This imbalance has a negative impact on the detection performance of low-frequency assaults.

C. Inadequate Performance in Real-World Settings

The research into the use of Intrusion Detection Systems (IDSs) has come up with the question of their effectiveness in direct applications. A large number of suggested techniques still depend on old datasets for their laboratory tests and validation, thus making their practical use questionable[87]. After a method gets through the controlled setting assessment, the next step would be the testing of its performance in the time-sensitive situations within the present networks.

D. Network-based intrusion detection systems (NIDS) Network-based Intrusion Detection Systems (NIDS) are responsible for the detection of intrusions through the constant monitoring and analysis of the network stream (packets/flows) by the detection of suspicious or malicious.

VIII. CONCLUSION

Intrusion and anomaly detection are still very important for security in the present day, but the combination of traditional rule/signature and simple statistical methods can hardly be considered sufficient anymore given the current situation with threats changing quickly, attack surfaces being so large encrypted communications, and dynamic system behavior. With the help of deep learning and advanced ML methods detection is much better as they can learn from large-scale security telemetry. Nevertheless, their application in real scenarios is still dependent on controlling false alarms, drift handling, ensuring robustness, and providing explanations are supportive of SOC decision-making. Recent surveys point out that the most promising direction is hybrid and operationally grounded IDS—bringing together multi source that detection stays accurate, scalable, and actionable despite the restrictions of real deployment.

REFERENCES

- [1] A. L. Buczak en E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection", *IEEE Commun. Surv. tutorials*, vol 18, no 2, bll 1153–1176, 2015.

- [2] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, en L. Shu, “Authentication protocols for internet of things: a comprehensive survey”, *Secur. Commun. Networks*, vol 2017, no 1, bl 6562953, 2017.
- [3] E. Bertino en N. Islam, “Botnets and internet of things security”, *Computer (Long. Beach. Calif.)*, vol 50, no 2, bll 76–79, 2017.
- [4] N. Moustafa, B. Turnbull, en K.-K. R. Choo, “An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things”, *IEEE Internet Things J.*, vol 6, no 3, bll 4815–4830, 2018.
- [5] Y. LeCun, Y. Bengio, en G. Hinton, “Deep learning”, [20] S. Rathore en J. H. Park, “Semi-supervised learningNature, vol 521, no 7553, bll 436–444, 2015.
- [6] S. Peltola, “Threat Detection Analysis Using MITRE ATT&CK Framework”, 2025.
- [7] Verizon, “2024 Data Breach Investigations Report”, 2024.
- [8] G. C. Security, “2024 Special Report”, 2024.
- [9] A. Khraisat, I. Gondal, P. Vamplew, en J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges”, *Cybersecurity*, vol 2, no 1, bll 1–22, 2019.
- [10] J. R. Tadhani, V. Vekariya, V. Sorathiya, S. Alshathri, en W. El-Shafai, “Securing web applications against XSS and SQLi attacks using a novel deep learning approach”, *Sci. Rep.*, vol 14, no 1, bl 1803, 2024, doi: 10.1038/s41598-023-48845-4.
- [11] K. Sambhus en Y. Liu, “Automating SQL Injection and Cross-Site Scripting Vulnerability Remediation in Code”, *Software*, vol 3, no 1. bll 28–46, 2024. doi: 10.3390/software3010002.
- [12] A. A. Aburomman en M. B. I. Reaz, “A survey of [23] Y. Cui, J. Cui, en J. Hu, “A Survey on XSS Attack intrusion detection systems based on ensemble and hybrid classifiers”, *Comput. Secur.*, vol 65, bll 135–152, 2017.
- [13] V. Chandola, A. Banerjee, en V. Kumar, “Anomaly Detection: A Survey”, *ACM Comput. Surv.*, vol 41, Jul 2009, doi: 10.1145/1541880.1541882.