ISSN 2581 – 3781

AJCSE
ASIAN JOURNAL OF COMPUTER SCIENCE ENGINEERING

Available Online at www.ajcse.info
Asian Journal of Computer Science Engineering 2025;10(2):1-14

REVIEW ARTICLE

Hybrid Intrusion Detection Alert System using a Highly Scalable Framework on Commodity Hardware Server

Gaurav Kulkarni[1]*, Pankaj Dharkar[2]

[1]*Department of Computer Science, ITM (SLS) Baroda University, Vadodara, Gujarat, India*
[2]*Principal Project Analyst CoreCARD Software India Private Ltd, Bhopal, Madhya Pradesh, India*

Received on: 05-04-2025; Revised on: 18-05-2025; Accepted on: 22-06-2025

ABSTRACT

This abstract presents a novel approach to developing a Hybrid Intrusion Detection Alert System (HIDS) designed for deployment on commodity hardware servers, while maintaining high scalability and real-time performance. Traditional Intrusion Detection Systems (IDS) often struggle with the increasing volume and sophistication of cyber threats, requiring significant computational resources and often leading to scalability issues or high false positive rates. Our proposed HIDS integrates both signature-based and anomaly-based detection mechanisms to provide a comprehensive and robust security solution. The system leverages a highly scalable framework, utilizing distributed processing paradigms to efficiently analyze large volumes of network traffic and system logs. By distributing the detection workload across multiple commodity hardware nodes, the system achieves linear scalability, allowing for cost-effective expansion as network size and data throughput increase. The anomaly detection component utilizes machine learning algorithms trained on normal system behavior, enabling the identification of previously unknown threats. Furthermore, the system incorporates an intelligent alert correlation engine to reduce alert fatigue and provide actionable insights to security analysts. This framework aims to offer a practical, efficient, and economically viable solution for organizations seeking enhanced network security without incurring prohibitive infrastructure costs.

Key words: Anomaly-based detection, Commodity hardware, Distributed processing, Hybrid intrusion detection system (HIDS), Real-time performance, Signature-based detection

INTRODUCTION

The digital realm is currently experiencing an unprecedented surge in both the frequency and sophistication of network intrusions. This escalation of cyber threats necessitates the development and deployment of increasingly advanced security mechanisms to safeguard critical infrastructures and sensitive data.[1] Traditional security measures, while foundational, often prove inadequate against the nuanced tactics employed by modern adversaries, thereby underscoring the imperative for robust Intrusion Detection Systems (IDS) capable of identifying and mitigating these evolving threats.[3] The proliferation of Internet of Things devices and the widespread adoption of cloud computing paradigms have further expanded the attack surface, creating a more intricate and challenging security landscape that demands more efficacious and adaptable security solutions.[4] The escalating interconnectedness and the sheer volume of data traversing contemporary networks establish a dynamic and demanding environment for cybersecurity, necessitating the deployment of adaptive and high-performance IDS.

Traditional IDS encounter significant challenges when confronted with the task of processing the massive quantities of network traffic generated in modern environments. The sheer volume often leads to delays in identifying potential threats, thereby compromising the system's effectiveness.[6] Furthermore, many traditional systems rely on signature-based detection methodologies, which, while effective against known attack patterns, are inherently limited in their ability to identify novel or 0-day exploits.[7] Conversely, anomaly-based IDS, designed to detect deviations from normal network behavior and thus capable of

Address for correspondence:
Gaurav Kulkarni
E-mail: gaurav.kulkarni@itmbu.ac.in

identifying new threats, often suffer from elevated false positive rates and necessitate substantial computational resources for their operation.[7] Moreover, a considerable number of existing IDS lack the necessary scalability to effectively adapt to the ever-changing network conditions and the continuous evolution of attack strategies.[5] The inherent limitations associated with relying solely on either signature-based or anomaly-based detection techniques underscore the necessity for a hybrid approach that can leverage the strengths of both while concurrently mitigating their respective weaknesses.

The motivation for developing a hybrid and highly scalable intrusion detection system capable of operating efficiently on commodity hardware stems from several critical factors. The need for real-time or near real-time responsiveness to network intrusions is paramount for ensuring effective network protection and minimizing potential damage.[2] Scalability is equally essential to accommodate the continuously increasing volumes of network traffic without experiencing a detrimental impact on performance.[12] The utilization of commodity hardware offers a compelling advantage in terms of cost-effectiveness and operational flexibility when compared to solutions that necessitate specialized and often expensive hardware.[12] A well-designed hybrid approach has the potential to significantly enhance detection accuracy and reduce the occurrence of false positives by intelligently integrating different analytical techniques.[9] Constructing a high-performance IDS on readily available and economically viable hardware is therefore crucial for facilitating its widespread adoption and practical implementation across a diverse range of network environments.

This research proposes a novel hybrid intrusion detection alert system built on a highly scalable framework specifically designed for deployment on commodity hardware servers. The system aims to combine the strengths of both anomaly-based and signature-based detection methodologies to achieve enhanced accuracy and a reduced false positive rate. The framework is engineered for scalability, allowing for distributed deployment across a cluster of commodity servers to handle high volumes of network traffic. The system is designed to generate timely alerts on the detection of suspicious or malicious activities. Furthermore, the research explores potential future enhancements to the proposed framework, notably the integration of modules for monitoring domain name system (DNS) and border gateway protocol (BGP) events within the network. The report also discusses how the execution time of the proposed system can be further improved by the straightforward addition of more nodes to the existing server cluster. While the current iteration of the proposed system focuses on network traffic analysis for intrusion detection, it does not provide detailed information regarding the structure and characteristics of identified malware. Recognizing this limitation, the research outlines a significant direction for future work: The integration of advanced malware analysis capabilities through the training of complex deep neural networks (DNNs) on advanced hardware using a distributed approach. Due to the substantial computational resources required for training such complex DNN architectures, this task was not undertaken within the scope of the current research but is considered a critical next step. This research contributes a novel hybrid intrusion detection approach with inherent scalability on readily available hardware. It also highlights the potential for significant improvements in threat detection through the incorporation of DNS and BGP monitoring and lays the groundwork for future advancements in malware analysis using distributed deep learning techniques.

RELATED WORK

The domain of network security has witnessed considerableresearchintothedevelopmentofhybrid IDS, aiming to overcome the inherent limitations of relying on single detection methodologies. These systems typically integrate signature-based detection, which excels at identifying known threats through pattern matching, with anomaly-based detection, which can identify novel attacks by detecting deviations from established normal behavior.[7] Existing hybrid systems often employ a combination of machine learning and deep learning techniques to enhance their detection capabilities.[19] For instance, one proposed system utilizes k-means and Random Forest algorithms for initial binary classification, followed by convolutional neural networks (CNNs) and long short-term memory (LSTM) networks for further classifying abnormal events into specific attack

types.[21] Another approach combines anomaly detection based on cross-entropy with a signature-based attack detection framework utilizing genetic algorithms.[7] While these hybrid systems offer improved detection accuracy compared to their individual counterparts, they can still suffer from potential computational overhead and complexities in effectively integrating disparate detection engines, potentially impacting their real-time performance and scalability.[3]

To address the ever-increasing volumes of network traffic, significant research has focused on developing scalable intrusion detection frameworks designed for deployment on commodity hardware. These frameworks often employ distributed architectures to distribute the computational load across multiple machines.[6] Techniques such as distributing network traffic evenly across an extensible set of analysis nodes, parallel processing of traffic segments, and managing analysis state across distributed nodes are commonly utilized.[12] The NIDS cluster, for example, is a system built on commodity PCs that collaboratively analyze a traffic stream by transparently exchanging low-level analysis state, effectively performing the same analysis as a single instance of the NIDS would if it could handle the full network load.[12] Similarly, Kargus is a highly scalable software-based IDS that exploits the full potential of commodity computing hardware by batch processing incoming packets at network cards and balancing pattern matching workloads across multicore central processing units (CPUs) and heterogeneous graphics processing units (GPUs).[14] The University of Glasgow successfully scaled its IDS using Gigamon technology to filter and load-balance traffic across multiple cost-effective Linux servers, demonstrating the feasibility of using commodity hardware for high-throughput intrusion detection.[13] These examples illustrate that building scalable IDSs on readily available hardware is not only possible but also a practical approach to managing the increasing demands of network security.

Deep learning techniques have emerged as a powerful paradigm in the field of intrusion detection, largely due to their inherent ability to automatically learn intricate patterns from complex and high-dimensional data such as network traffic.[1] Models like CNNs have proven effective at pattern recognition in structured data, while recurrent neural networks, particularly LSTMs, excel at capturing temporal dependencies within sequential data streams.[1] The application of deep learning to intrusion detection often involves training these models on large datasets of network traffic to distinguish between benign and malicious activities.[1] Studies have demonstrated the superior performance of deep learning-based IDSs on various benchmark datasets, achieving high accuracy and low false-positive rates in detecting a wide range of network intrusions, including distributed denial-of-service (DDoS), Botnet, and Brute Force attacks.[1] The ability of deep learning models to automatically extract relevant features from raw network traffic eliminates the need for manual feature engineering, a significant advantage in dealing with the dynamic and evolving nature of cyber threats.[1] This growing body of research underscores the significant potential of deep learning to enhance the accuracy and overall effectiveness of IDS.

To handle the massive datasets and computational demands associated with modern network security analysis, distributed computing frameworks have become increasingly relevant. Platforms such as Apache Spark and Apache Hadoop provide the infrastructure necessary for processing large-scale data in a parallel and distributed manner.[5] These frameworks enable the distribution of network traffic data across a cluster of machines, allowing for parallel analysis and the training of complex machine learning and deep learning models on distributed clusters.[5] The inherent scalability and fault-tolerance offered by these frameworks make them well-suited for building high-performance IDS capable of handling the immense data volumes characteristic of today's network environments.[23] For instance, Spark has been utilized to implement distributed versions of machine learning algorithms such as k-means and Random Forest for intrusion detection.[19] This capability to leverage distributed resources is crucial for developing IDSs that can keep pace with the ever-increasing scale and complexity of network traffic.

Monitoring DNS and BGP events within network traffic has emerged as a valuable strategy for enhancing intrusion detection capabilities. Analyzing DNS traffic can reveal various malicious activities, including network scanning, botnet command and control communications, and data exfiltration attempts.[3] Techniques

Table 1: Comparison of existing intrusion detection systems

System name	Detection method	Scalability	Hardware requirements	DNS/BGP monitoring	Malware analysis detail	Performance metrics (if available)
DL-NIDS-ZTN	Deep Learning (CNN)	High	Commodity	No	No	Accuracy: 99.80%[6]
Hybrid IDS (Qazi)	ML and Microservices	High	Commodity	No	No	Accuracy: Not specified
Hybrid IDS (Bronte)	Anomaly and Signature-based	Medium	Commodity	No	No	Accuracy: Not specified
Hybrid IDS (Liu)	ML and Deep Learning	High	Commodity	No	No	TPR: Better for most attacks[21]
Hybrid IDS (Grace)	Spark ML and Conv-LSTM	High	Commodity	No	No	Accuracy: 97.29%, F1-score: 97.29%[19]
NIDS Cluster	Signature-based (Bro)	High	Commodity	No	No	Accuracy: Not specified
Kargus	Signature-based	High	Commodity (central processing unit/graphics processing unit)	No	No	Throughput: Up to 33 Gbps[14]

for DNS monitoring often involve correlating network flows with preceding DNS queries and responses to detect scanning probes or identifying unusual patterns in DNS requests that might indicate malicious behavior.[3] Similarly, monitoring BGP updates can provide critical insights into the integrity and stability of network routing infrastructure. Analyzing BGP events can help detect routing anomalies, such as route hijacks where traffic is redirected to malicious destinations, and other security threats targeting the fundamental protocols that govern internet traffic.[29] While various approaches exist for DNS and BGP monitoring, their integration into a comprehensive intrusion detection system can significantly improve the ability to detect sophisticated attacks that operate at different layers of the network.[25]

Table 1 provides a comparison of existing IDS based on the reviewed literature.

PROPOSED HYBRID INTRUSION DETECTION ALERT SYSTEM

The proposed hybrid intrusion detection alert system is designed to provide a robust and scalable solution for detecting malicious activities within network traffic. Its architecture and operational mechanisms are detailed in the following subsections.

Detailed Architecture of the Proposed System

The proposed system employs a modular architecture to facilitate flexibility and scalability. The core components include a Network Traffic Capture Module, a Traffic Distribution Module, a Hybrid Detection Engine comprising both Anomaly Detection and Signature-Based Detection sub-modules, an Alert Generation and Management Module, and a Data Storage and Analysis Module. Network traffic is initially captured by the Network Traffic Capture Module, which can interface with various network monitoring tools and technologies. The captured traffic is then passed to the Traffic Distribution Module. This module is responsible for efficiently distributing the incoming traffic stream across a cluster of commodity hardware servers. Load balancing algorithms are employed to ensure an even distribution of the workload, preventing any single server from becoming a bottleneck.

Each server in the cluster hosts an instance of the Hybrid Detection Engine. This engine is the core of the intrusion detection system and consists of two primary sub-modules: The Anomaly Detection Module and the Signature-Based Detection Module. The Anomaly Detection Module analyzes network traffic for deviations from established baseline behavior. It employs statistical methods and machine learning algorithms to identify unusual patterns that may indicate malicious activity. The Signature-Based Detection Module, on the other hand, utilizes a database of known attack signatures and patterns to identify traffic that matches previously identified threats. The results from both the Anomaly Detection and Signature-Based Detection Modules are then fed into a Decision Engine. This engine combines the outputs from the two detection modules, potentially using techniques such as weighted scoring or ensemble methods, to make a final determination

on whether the traffic is benign or malicious.

On identifying malicious traffic, the Alert generation and management module is triggered. This module creates detailed alerts containing relevant information about the detected intrusion, such as the type of attack, source and destination IP addresses, timestamps, and severity level. These alerts are then logged and can be prioritized based on their severity. The Alert Management component also provides mechanisms for integrating with other security tools, such as security information and event management (SIEM) systems, allowing for centralized monitoring and incident response. Finally, the data storage and analysis module is responsible for storing network traffic data, detection logs, and generated alerts. This data can be used for further analysis, threat intelligence gathering, and refining the detection models over time.

Explanation of the Hybrid Detection Approach

The proposed system leverages a hybrid detection approach that strategically combines the strengths of both anomaly-based and signature-based intrusion detection techniques. The anomaly detection module is designed to identify novel or previously unseen attacks by establishing a baseline of normal network behavior and detecting any significant deviations from this baseline. This module can employ a range of statistical methods, such as monitoring traffic volume, protocol usage, and connection patterns, to identify unusual activities. In addition, machine learning algorithms, including clustering algorithms like k-means or classification algorithms like support vector machines or random forests, can be trained on historical network traffic data to learn normal behavior patterns and subsequently flag any anomalous traffic.[21]

Complementing the anomaly detection module is the signature-based detection module. This module relies on a regularly updated database of known attack signatures and patterns. It examines incoming network traffic and compares it against these signatures to identify instances of known malicious activities. Signature-based detection is particularly effective at accurately and efficiently identifying well-established threats with a low rate of false positives.[20] The signatures can include specific byte sequences, protocol anomalies, or known malicious IP addresses and domains.

The integration of these two detection approaches is crucial for the overall effectiveness of the hybrid system. The decision engine plays a key role in combining the outputs from the anomaly detection and signature-based detection modules. For instance, if the anomaly detection module flags a traffic pattern as potentially suspicious but does not definitively match a known attack, the signature-based detection module can provide further analysis. Conversely, if the signature-based detection module identifies a known attack, the anomaly detection module can provide contextual information about the surrounding network behavior. The decision engine can use a weighted scoring system, where the confidence level of each detection module's output contributes to a final score. If this score exceeds a predefined threshold, an alert is generated. Ensemble methods, where the results of multiple anomaly detection algorithms and the signature-based detection are combined using techniques such as voting or stacking, can also be employed to improve the accuracy and robustness of the final intrusion determination.[6] This synergistic combination allows the system to detect both known and novel threats while minimizing the limitations inherent in each individual approach.

Design of the Scalable Framework for Commodity Hardware

The scalability of the proposed intrusion detection system is achieved through a distributed framework designed to operate efficiently on commodity hardware servers. The Traffic Distribution Module is a key component of this framework, responsible for ensuring that incoming network traffic is effectively distributed across the available servers in the cluster. This can be implemented using various load balancing techniques. For example, software-defined networking principles can be leveraged to intelligently route traffic to different servers based on factors such as current load and server availability.[13] Traditional load balancers, operating at either the network or application layer, can also be employed to distribute traffic based on algorithms such as round-robin or least connections.[13]

Once the traffic reaches a server, parallel processing is crucial for maximizing throughput. Each commodity server typically features multi-core CPUs, which can be effectively utilized to process different segments of the network traffic

concurrently. Techniques such as multi-threading and multi-processing can be implemented within the hybrid detection engine to divide the workload across multiple CPU cores, allowing for parallel analysis of different traffic flows or packets.[14] Furthermore, many commodity servers are now equipped with powerful GPUs, which can be leveraged for accelerating certain aspects of the intrusion detection process, particularly the computationally intensive tasks associated with anomaly detection algorithms and future deep learning-based malware analysis.[14]

Maintaining system state and coordinating analysis across the distributed nodes is essential for detecting attacks that may span multiple traffic flows or time intervals. A distributed state management mechanism can be implemented where relevant information about network connections and ongoing traffic patterns is shared among the servers in the cluster.[12] This can be achieved through a distributed in-memory data store or by utilizing the capabilities of distributed computing frameworks like Apache Spark. Spark's ability to distribute data and computations across a cluster of commodity machines makes it a suitable platform for building the scalable framework for the proposed IDS.[19] By leveraging these traffic distribution, parallel processing, and distributed state management techniques, the proposed system can achieve high scalability and efficiently analyze large volumes of network traffic using readily available commodity hardware.

Mechanisms for Alert Generation and Management

The Alert Generation and Management Module is responsible for creating and handling notifications when the Hybrid Detection Engine identifies malicious activity. Alerts are generated based on the final determination made by the Decision Engine. If the combined analysis of the Anomaly Detection and Signature-Based Detection Modules indicates a high probability of an intrusion, an alert is triggered. The conditions for alert generation can be configured based on the severity of the detected anomaly or the confidence level of the signature match. For instance, a definitive match with a critical attack signature may immediately trigger a high-priority alert, while a less severe anomaly might generate a lower-priority warning for further investigation.

The generated alerts contain comprehensive information designed to assist network administrators in understanding and responding to the security incident. This information typically includes a description of the detected attack type (if identified), the timestamp of the detection, the source and destination IP addresses and ports involved in the suspicious traffic, the protocol used, and a severity level indicating the potential impact of the intrusion. The format of the alerts can be standardized, for example, using the common event format, to ensure compatibility with various security management tools.

The alert management component of the module handles the logging, prioritization, and integration of alerts. All generated alerts are stored in a centralized log for auditing and historical analysis. The system can prioritize alerts based on their severity level, allowing security teams to focus on the most critical incidents first. Furthermore, the alert management module is designed to integrate seamlessly with other security infrastructure, such as SIEM systems, intrusion prevention systems, and network management platforms. This integration enables a coordinated and automated response to detected threats. For example, an alert generated by the proposed system could automatically trigger a blocking rule in a firewall or initiate an investigation workflow within a SIEM system. This comprehensive approach to alert generation and management ensures that detected intrusions are not only identified but also effectively communicated and acted on, enhancing the overall security posture of the network.

SCALABILITY AND PERFORMANCE ENHANCEMENTS

Achieving high scalability on commodity hardware necessitates a multi-faceted approach that optimizes resource utilization and distributes the workload effectively. The proposed system employs several strategies to meet these demands.

Strategies for Achieving High Scalability on Commodity Hardware

Load balancing is a cornerstone of the proposed system's scalability strategy. The traffic distribution

module utilizes sophisticated load balancing techniques to ensure that the incoming network traffic is distributed evenly across all the servers in the cluster.[12] This prevents any single server from being overwhelmed by a disproportionate amount of traffic, thereby maintaining overall system performance. Various load balancing algorithms can be employed, including round-robin, which distributes traffic sequentially across the servers, and more intelligent algorithms that consider server load and capacity.[13]

Parallel processing is another critical aspect of achieving high scalability. Each commodity server in the cluster is capable of processing multiple tasks concurrently by leveraging its multi-core CPU architecture.[14] The hybrid detection engine is designed to take advantage of this parallelism using techniques such as multi-threading, where multiple threads within a single process execute different parts of the detection logic simultaneously, and multi-processing, where multiple independent processes handle different traffic flows.[14] Furthermore, the potential of utilizing GPUs, which are increasingly common in commodity servers, for accelerating specific computational tasks, such as cryptographic operations or pattern matching in anomaly detection, will be explored in future optimizations. The proposed system can also leverage distributed computing frameworks like Apache Spark for scalable data analysis.[19] Spark's ability to process large datasets in parallel across a cluster makes it well-suited for tasks such as training and updating the anomaly detection models based on historical network traffic data. By distributing the data and the computational workload across multiple nodes, Spark can significantly reduce the time required for these intensive tasks, contributing to the overall scalability and responsiveness of the intrusion detection system. This combination of efficient load balancing, parallel processing within each server, and the potential use of distributed computing frameworks ensures that the proposed system can effectively scale to handle high volumes of network traffic on commodity hardware.

Discussion on How Adding Nodes to the Cluster Enhances Execution Time

One of the key advantages of the proposed system's scalable framework is its ability to enhance execution time by simply adding more nodes to the server cluster. This concept, known as horizontal scaling, allows the system to distribute the processing workload across a larger pool of resources.[12] When a new server is added to the cluster, the traffic distribution module automatically incorporates it into the load balancing scheme, thereby reducing the amount of traffic that each individual server needs to process.

Theoretically, with an efficient load balancing mechanism and minimal coordination overhead, the system's performance should scale near-linearly with the number of nodes. For instance, if the network traffic volume doubles, adding a similar number of servers to the cluster should roughly maintain the original execution time per unit of traffic.[12] However, it is important to acknowledge that there may be practical limitations to this scalability. Factors such as network bandwidth limitations between the servers, the overhead associated with coordinating the analysis across a large number of nodes, and potential bottlenecks in shared resources could eventually impact the linear scalability.[15] Nevertheless, the ability to improve the system's capacity and reduce processing time by adding more commodity servers provides a cost-effective and flexible way to adapt to growing network demands. This inherent scalability is a significant advantage over systems that require vertical scaling, which involves upgrading to more powerful and expensive hardware.

Potential Performance Metrics and Evaluation Methods

To rigorously evaluate the effectiveness and efficiency of the proposed hybrid intrusion detection system, several key performance metrics will be considered. Detection accuracy, which measures the system's ability to correctly identify malicious traffic, is a primary metric. This is often expressed as the percentage of correctly classified instances out of the total number of instances. The false-positive rate (FPR), which indicates the proportion of benign traffic incorrectly flagged as malicious, is equally important as a high FPR can lead to alert fatigue and operational inefficiencies. Conversely, the false-negative rate, which measures the proportion of malicious traffic

that goes undetected, needs to be minimized to ensure effective security. Detection latency, the time taken by the system to identify an intrusion after it occurs, is critical for real-time protection. Throughput, which measures the volume of network traffic that the system can process per unit of time without significant performance degradation, will also be evaluated to assess the system's scalability.

The evaluation methodology will involve testing the proposed system using standard benchmark datasets commonly employed in intrusion detection research. Datasets such as CICIDS2017 and NSL-KDD, which contain a diverse range of both benign and malicious network traffic patterns, will be utilized for training and testing the system.[1] The experimental setup will involve deploying the proposed system on a cluster of commodity hardware servers and subjecting it to various traffic scenarios, including both normal traffic and different types of simulated attacks. The performance metrics mentioned above will be measured under different traffic loads and cluster sizes to assess the system's scalability and its ability to maintain performance under varying conditions. Furthermore, the performance of the proposed system will be compared against existing state-of-the-art IDS, including both hybrid and non-hybrid approaches, to demonstrate its effectiveness and potential advantages.[19] This comprehensive evaluation using standard metrics and benchmark datasets will provide a robust assessment of the proposed system's capabilities.

FUTURE ENHANCEMENTS

The proposed hybrid intrusion detection alert system offers a solid foundation for network security. However, its capabilities can be further enhanced by integrating additional modules for monitoring specific network events. This section details the proposals for incorporating DNS and BGP monitoring into the framework.

Detailed Proposal for Integrating a Module for Monitoring DNS Events

Integrating a module for monitoring DNS events can significantly enhance the threat detection capabilities of the proposed system. DNS, the protocol responsible for translating human-readable domain names into IP addresses, plays a critical role in almost all internet communications. Monitoring DNS traffic can reveal various malicious activities, including command and control (C2) communications established by malware, data exfiltration attempts, and techniques used to evade security controls.[3]

The proposed DNS monitoring module will involve capturing DNS queries and responses traversing the network. This can be achieved by passively sniffing network traffic or by querying dedicated DNS servers. The captured DNS data will then be analyzed for suspicious patterns. For instance, a high volume of DNS queries to newly registered or known malicious domains could indicate botnet activity or malware attempting to establish communication with its C2 server.[26] The module will also look for indicators of DNS tunneling, a technique where malicious data is encoded within DNS queries and responses to bypass firewalls and other security measures.[26] In addition, the module can track DNS fast flux techniques, where attackers rapidly change the IP addresses associated with a domain to make it difficult to track and block malicious websites.[26] The DNS monitoring module will be designed to integrate seamlessly with the existing hybrid detection framework. The analysis results from the DNS module can be fed into the decision engine, where they can be correlated with the outputs from the anomaly detection and signature-based detection modules. For example, if the DNS module detects queries to a known malware domain originating from an internal host that was also flagged as exhibiting anomalous network behavior by the anomaly detection module, the confidence level of a potential intrusion would be significantly increased, leading to the generation of a high-priority alert. This integration will provide a more comprehensive view of network security threats by leveraging the information contained within DNS traffic.

Detailed Proposal for Integrating a Module for Monitoring BGP Events

Integrating a module for monitoring BGP events offers another valuable enhancement to the proposed intrusion detection system. BGP, the Border Gateway Protocol, is the fundamental

routing protocol that enables communication between different autonomous systems (AS) on the internet. Monitoring BGP updates can provide critical insights into the stability and security of network routing and can help detect attacks that manipulate routing information for malicious purposes.[29]

The proposed BGP monitoring module will involve capturing and analyzing BGP update messages exchanged between routers. This can be achieved by establishing peering sessions with network routers or by utilizing publicly available BGP monitoring data. The module will focus on detecting various types of BGP anomalies, such as route hijacks, where an attacker announces a route to a prefix that they do not own, potentially redirecting traffic destined for legitimate networks to malicious destinations.[34] The module will also monitor for unusual changes in BGP routing paths, which could indicate misconfigurations or malicious activity. In addition, the module can track changes in prefix origin AS numbers, which might signal attempts to impersonate legitimate network entities.[34]

Similar to the DNS monitoring module, the BGP monitoring module will be designed to integrate with the existing hybrid detection framework. The analysis results from the BGP module can be fed into the Decision Engine and correlated with the outputs from the other detection modules. For instance, if the BGP module detects a route hijack attempt originating from an internal network that is also exhibiting anomalous traffic patterns, the system can generate a high-priority alert, indicating a potential coordinated attack targeting both the network infrastructure and end hosts. This integration will provide a broader security perspective by monitoring the routing layer of the network, which is often overlooked by traditional IDS that primarily focus on network traffic content.

Discussion on the Benefits of These Enhancements for Improved Threat Detection

The integration of both DNS and BGP monitoring modules offers significant benefits for improving the overall threat detection capabilities of the proposed hybrid intrusion detection system. DNS monitoring can provide valuable insights into the communication patterns of malware and other malicious software. By detecting queries to known malicious domains or unusual DNS traffic patterns, the system can identify infected hosts and potentially block communication with command and control servers, thereby mitigating the impact of malware infections.[9] Furthermore, DNS monitoring can help in detecting phishing attacks by identifying access attempts to fraudulent websites that may be using domain names similar to legitimate ones.[26]

BGP monitoring, on the other hand, enhances the system's ability to detect attacks that target the network infrastructure itself. By identifying route hijacks and other BGP anomalies, the system can alert network administrators to potential attempts to disrupt network availability or redirect traffic to malicious destinations for eavesdropping or other malicious purposes.[34] This is particularly important for maintaining the overall stability and integrity of the network.

The combination of these enhancements with the existing hybrid detection approach will lead to a more robust and comprehensive security solution. The DNS monitoring module adds an application-layer perspective to threat detection, while the BGP monitoring module provides a network infrastructure-level view. These modules complement the traffic content analysis performed by the anomaly detection and signature-based detection modules, resulting in a multi-layered security approach that can detect a wider range of sophisticated attacks across different layers of the network. This integrated approach will significantly strengthen the overall security posture of the network by providing enhanced visibility into potentially malicious activities that might otherwise go undetected.

LIMITATIONS AND FUTURE WORK

While the proposed hybrid intrusion detection alert system offers a promising approach to enhancing network security, it is important to acknowledge its current limitations and outline directions for future research and development.

Acknowledgement of the Current System's Limitations in Providing Detailed Malware Information

The current research focuses primarily on the

Table 2: Potential benchmark datasets for evaluation

Dataset name	Key characteristics	Relevance to proposed system/future work
CICIDS2017	Realistic traffic, diverse attacks (including Botnet, distributed denial-of-service, and Web Attacks)	Baseline for intrusion detection performance; can be augmented with malware traffic.
NSL-KDD	Widely used for intrusion detection research, includes various attack categories	Useful for comparing performance with existing studies; may need to be supplemented for detailed malware analysis.
Malware executable detection dataset	Contains features of malware executables	Directly relevant for training and evaluating the deep neural network-based malware analysis module.
Bot-IoT	Specifically designed for IoT intrusion detection, includes botnet attacks	Relevant for evaluating the system's performance in IoT environments and its ability to detect botnet malware.
UNSW-NB15	Modern dataset with diverse attack types	Provides a contemporary dataset for evaluating intrusion detection capabilities.
BCCC-CSE-CIC-IDS2018	Includes benign and various attack traffic, with a large number of features	Offers a comprehensive dataset for evaluating both intrusion detection and potentially for identifying network patterns of malware.

IoT: Internet of Things

detection of network intrusions and the generation of alerts based on the analysis of network traffic patterns. The proposed system, in its present form, does not incorporate detailed malware analysis capabilities that would provide specific information about the structure, behavior, and characteristics of any detected malware [User Query]. This limitation is primarily due to the significant computational resources and the specialized techniques required for in-depth malware analysis, which were beyond the scope of this initial research effort. While the system can identify potentially malicious network activity that might be associated with malware, it does not delve into the intricacies of the malware itself, such as its specific functionalities, the extent of its potential damage, or methods for its removal. The primary goal of the current system is to provide timely alerts about potential intrusions based on network traffic anomalies and known attack signatures, enabling network administrators to take immediate action to investigate and mitigate the detected threats.

Comprehensive Plan for Future Work Involving Training Complex DNN Architectures

A significant direction for future work involves enhancing the proposed system with advanced malware analysis capabilities. This will entail integrating mechanisms for not only detecting intrusions but also for providing detailed insights into the malware responsible for the attack. To achieve this, future research will focus on training complex DNN architectures specifically designed for malware classification and analysis.[1] This will involve collecting and curating a comprehensive dataset of malware samples and their associated network traffic patterns. The DNN models will then be trained to learn the distinguishing features and characteristics of different types of malware, enabling the system to not only detect malicious activity but also to classify the type of malware involved. This will provide valuable information for incident response and remediation efforts. Training such complex DNN models requires significant computational resources, including advanced hardware such as high-performance GPUs and substantial memory capacity. Future work will therefore necessitate access to such advanced hardware infrastructure.

Proposed Distributed Approach for Training DNNs on Advanced Hardware

Given the substantial computational demands of training complex DNN architectures for malware analysis, a distributed approach will be essential. Future research will explore the use of distributed deep learning frameworks to train these models across multiple high-performance machines.[4] Distributed training offers several key benefits, including the ability to significantly reduce the training time for large and complex models by distributing the computational workload across multiple processing units. It also enables the handling of very large datasets that might not fit into the memory of a single machine. However, distributed training also presents challenges, such as the overhead associated with communication and synchronization between the distributed nodes.

Future work will investigate efficient strategies for data parallelism and model parallelism to optimize the distributed training process for the specific task of malware analysis within the context of the proposed intrusion detection system.

Potential Benchmark Datasets for Evaluating the Enhanced System

To evaluate the performance of the enhanced system with integrated DNN-based malware analysis, appropriate benchmark datasets that include both network traffic data and malware samples will be required. Several publicly available datasets, such as those containing network traffic from various attack scenarios and repositories of malware samples, could be utilized.[1] The evaluation metrics for the malware analysis module will include classification accuracy, precision, and recall, which will measure the system's ability to correctly identify and classify different types of malware. In addition, the overall performance of the enhanced intrusion detection system, including its detection accuracy for network intrusions and its ability to provide detailed malware information, will be assessed to demonstrate the effectiveness of the integrated approach.

Table 2 outlines potential benchmark datasets for evaluating the enhanced system.

CONCLUSION

This research proposes a hybrid intrusion detection alert system built on a highly scalable framework designed for operation on commodity hardware servers. The system strategically combines anomaly-based and signature-based detection techniques to enhance the accuracy and reduce the FPR associated with traditional IDS. The framework's architecture is engineered for scalability, enabling deployment across a cluster of commodity servers to effectively handle the increasing volumes of network traffic characteristic of modern network environments. Furthermore, the research explores the potential for significant future enhancements, notably the integration of modules for monitoring DNS and BGP events, which promise to provide deeper insights into network behavior and infrastructure-level attacks. The ability to improve the system's performance by simply adding more commodity servers to the cluster offers a cost-effective and flexible approach to adapting to evolving network demands. While the current system focuses on network traffic analysis for intrusion detection and alerting, acknowledging its limitations in providing detailed malware information, the research lays out a comprehensive plan for future work. This future direction involves leveraging the power of complex DNNs trained on advanced hardware using a distributed approach to integrate advanced malware analysis capabilities into the system. This will enable the system to not only detect intrusions but also to provide valuable information about the structure and characteristics of the malware involved. The proposed system and the outlined future work hold significant potential for enhancing network security by providing a cost-effective, scalable, and comprehensive solution for detecting and understanding a wide range of cyber threats.

REFERENCES AQ5

1. HPAC-IDS: A Hierarchical Packet Attention Convolution for Intrusion Detection System. Available from: https://arxiv.org/html/2501.06264v1 [Last accessed on 2025 May 17].
2. Deep Learning Algorithms Used in Intrusion Detection Systems - Dedan Kimathi University of Technology. Available from: https://repository.dkut.ac.ke:8080/xmlui/bitstream/handle/123456789/8473/2402.17020v1/pdf?sequence=1&isallowed=y [Last accessed on 2025 May 17].
3. (PDF) A Comprehensive Systematic Literature Review on Intrusion Detection Systems. Available from: https://www.researchgate.net/publication/356368631_a_comprehensive_systematic_literature_review_on_intrusion_detection_systems [Last accessed on 2025 May 17].
4. IEEE Standards and Deep Learning Techniques for Securing. Available from: https://eudoxuspress.com/index.php/pub/article/view/1210 [Last accessed on 2025 May 17].
5. Abid A, Jemili F, Korbaa O. Distributed deep learning approach for intrusion detection system in industrial control systems based on big data technique and transfer learning. J Inf Telecom 2023;7:513-41.
6. Balaji P, Pradeeswari R, Rathimalar G, Thenmozhi A, Varshini J. Real time intrusion detection with latency optimization. Int J Res Public Rev 2025;6:7060-4.
7. (PDF) A Framework for Hybrid Intrusion Detection Systems - Research Gate. Available from: https://www.researchgate.net/publication/311575310_a_framework_for_hybrid_intrusion_detection_systems [Last accessed on 2025 May 17].
8. Enhancing Network Intrusion Detection Performance

using Generative Adversarial Networks. Available from: https://arxiv.org/html/2404.07464v1 [Last accessed on 2025 May 17].
9. Dini P, Elhanashi A, Begni A, Saponara S, Zheng Q, Gasmi K. Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity. Appl Sci 2023;13:7507.
10. What is Malware Detection? Importance and Techniques - Sentinel One. Available from: https://www.sentinelone.com/cybersecurity/101/threat/intelligence/what/is/malware/detection/[Last accessed on 2025 May 17].
11. Malware Detection: Techniques and Technologies - Check Point. Available from: https://www.checkpoint.com/cyber/hub/threat/prevention/what/is/malware/malware/detection-techniques-and-technologies [Last accessed on 2025 May 17].
12. Available from: https://www.icir.org/robin/papers/raid07.pdf [Last accessed on 2025 May 17].
13. Available from: https://www.gigamon.com/content/dam/resource-library/english/case/study---use-cases/cs-university-of-glasgow.pdf [Last accessed on 2025 May 17].
14. Kargus: A Highly-Scalable Software-Based Intrusion Detection System. Available from: https://www.researchgate.net/publication/262208695_kargus_a_highly/scalable_software/based_intrusion_detection_system [Last accessed on 2025 May 17].
15. The NIDS Cluster: Scalable, Stateful Network Intrusion Detection on Commodity Hardware. Available from: https://www.researchgate.net/publication/221427491_the_nids_cluster_scalable_stateful_network_intrusion_detection_on_commodity_hardware [Last accessed on 2025 May 17].
16. Kargus: A Highly-Scalable Software-Based Intrusion Detection System - GitHub Pages. Available from: https://yung/web.github.io/home/publication/conference/kargus: a_highly/scalable_sof/ware-based_intrusion_detection_system.pdf [Last accessed on 2025 May 17].
17. The NIDS Cluster: Scalable, Stateful Network Intrusion Detection on Commodity Hardware. Available from: https://www.osti.gov/biblio/935341 [Last accessed on 2025 May 17].
18. The NIDS Cluster: Scalable, Stateful Network Intrusion Detection on Commodity Hardware. Available from: https://colab.ws/articles/10.1007%2F978-3-540-74320-0_6 [Last accessed on 2025 May 17].
19. Khan MA, Karim R, Kim. A scalable and hybrid intrusion detection system based on the convolutional-LSTM network. Symmetry 2019;11:583.
20. Top Malware Detection Techniques - Key Methods Explained. Available from: https://amatas.com/blog/top-malware-detection-techniques-key-methods-explained [Last accessed on 2025 May 17].
21. Liu C, Gu Z, Wang AJ. A hybrid intrusion detection system based on scalable K-means+ random forest and deep learning. IEEE Access 2021;9:75729-40.
22. Optimizing Intrusion Detection System Performance Through Synergistic Hyperparameter Tuning and Advanced Data Processing - arXiv. Available from: http://arxiv.org/abs/2408.01792 [Last accessed on 2025 May 17].
23. Apache Hadoop. Available from: https://hadoop.apache.org [Last accessed on 2025 May 17].
24. Goyal R, Ansari K, Shrivastava V, Pandey A, Phathak V. AI in meal planning unlocking the benefits of organized nutrition. Int J Res Public Rev 2025;6:5118-25.
25. Detecting Network Scanning Through Monitoring and Manipulation. Available from: https://doaj.org/article/9a5ffa95a5cd4662bb4a22640918d753 [Last accessed on 2025 May 17].
26. CMC. Free Full-Text. The Impact of Domain Name Server (DNS) over Hypertext Transfer Protocol Secure (HTTPS) on Cyber Security: Limitations, Challenges, and Detection Techniques - Tech Science Press. Available from: https://www.techscience.com/cmc/v80n3/57854/html [Last accessed on 2025 May 17].
27. Gezer A. Identification of abnormal DNS traffic with Hurst parameter. Balkan J Electr Comput Eng 2018;6:191-7.
28. Accepted Papers - IEEE Symposium on Security and Privacy; 2024. Available from: https://sp2024.ieee/security.org/accepted-papers.html [Last accessed on 2025 May 17].
29. Scott BA, Johnstone MN, Szewczyk P. A survey of advanced border gateway protocol attack detection techniques. Sensors (Basel) 2024;24:6414.
30. Scott BA, Johnstone MN, Szewczyk P. A survey of advanced border gateway protocol attack detection techniques. Sensors (Basel) 2024;24:6414.
31. Using External Security Monitors to Secure BGP - Computer Science Cornell. Available from: https:// www.cs.cornell.edu/ fbs/ publications/ nexusbgptr.pdf [Last accessed on 2025 May 17].
32. Silva RS, De Assis FM, Macodo EL, De Moraes LF. Inferring the confidence level of BGP-based distributed intrusion detection systems alarms. Ann Telecomm 2024;79:901-12.
33. Analysis of Intrusion Detection System (IDS) in Border Gateway Protocol - OPUS at UTS. Available from: https://opus.lib.uts.edu.au/bitstream/10453/21852/2/02whole.pdf [Last accessed on 2025 May 17].
34. IP Prefix Hijack Detection using BGP Connectivity Monitoring - PEARL. Available from: https://pearl.plymouth.ac.uk/cgi/viewcontent.cgi?article=2346&context=secam-research [Last accessed on 2025 May 17].
35. BGP Anomaly Detection Techniques: A Survey - GMU CS Department. Available from: https://cs.gmu.edu/~eoster/2019/795/2019/795/papers/bgp%20anomaly%20detection%20techniques%20survey.pdf [Last accessed on 2025 May 17].
36. ELISHA: A Visual-Based Anomaly Detection System for the BGP Routing Protocol - UCLA Computer Science Department. Available from: https://web.cs.ucla.edu/~lixia/papers/02raid.pdf [Last accessed on 2025 May 17].
37. Benchmark Intrusion Detection Datasets. Download Scientific Diagram. Available from: https://www.

researchgate.net/figure/benchmark/intrusion/detection/datasets_fig11_352725425 [Last accessed on 2025 May 17].
38. Network Intrusion Detection - Papers with Code. Available from: https://paperswithcode.com/task/network-intrusion-detection/codeless [Last accessed on 2025 May 17].
39. Ferriyan A, Thamrin AH, Takeda K, Murai J. Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic. Appl Sci 2021;11:7868.
40. CSE-CIC-IDS2018 on AWS - University of New Brunswick. Available from: https://www.unb.ca/cic/datasets/ids-2018.html [Last accessed on 2025 May 17].
41. IDS 2017. Datasets. Research. Canadian Institute for Cybersecurity. Available from: https://www.unb.ca/cic/datasets/ids-2017.html [Last accessed on 2025 May 17].
42. Expectations Versus Reality: Evaluating Intrusion Detection Systems in Practice - arXiv. Available from: https://arxiv.org/html/2403.17458v3 [Last accessed on 2025 May 17].
43. Ghurab M, Gaphari G, Alshami F, Alshamy R, Othman S. A detailed analysis of benchmark datasets for network intrusion detection system. Asian J Res Comput Sci 2021;7:14-33.
44. Benchmark Datasets for Network Intrusion Detection: A Review. Available from: https://ijns.jalaxy.com.tw/contents/ijns-v20-n4/ijns-2018-v20-n4-p645-654.pdf [Last accessed on 2025 May 17].
45. Ferrag MA, Shu L, Djallel H, Choo KK. Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. Electronics 2021;10:1257.
46. Deep Learning Approach for Intelligent Intrusion Detection System - Research Gate. Available from: https://www.researchgate.net/publication/332203589_deep_learning_approach_for_intelligent_intrusion_detection_system [Last accessed on 2025 May 17].
47. Ferrag MA, Shu L, Djallel H, Choo KK. Deep Learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. Electronics 2021;10:1257.
48. AN Analysis of Transferability in Network Intrusion Detection Using Distributed Deep Learning. Open Review. Available from: https://openreview.net/pdf?id=fpzbyci0yz1 [Last accessed on 2025 May 17].
49. HDLIDP: A Hybrid Deep Learning Intrusion Detection and Prevention Framework. Available from: https://www.techscience.com/cmc/v73n2/48352/html [Last accessed on 2025 May 17].
50. Federated Deep Learning for Intrusion Detection in IoT Networks - arXiv. Available from: https://arxiv.org/abs/2306.02715 [Last accessed on 2025 May 17].
51. Zhang Y, Muniyandi RC, Qamar F. A review of deep learning applications in intrusion detection systems: Overcoming challenges in spatiotemporal feature extraction and data imbalance. Appl Sci 2025;15:1552.
52. Li G, Cui Y, Su J. Adaptive mechanism-based grey wolf optimizer for feature selection in high-dimensional classification. PLoS One 2025;20:e0318903.
53. Jayalaxmi P, Saha R, Kumar G, Conti M, Kim TH. Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey. IEEE Access 2022;4:1
54. Ali ML, Thakur K, Schmeelk S, Debello J, Dragos D. Deep learning vs. Machine learning for intrusion detection in computer networks: A comparative study. Appl Sci 2025;15:1903.
55. Li Q, Tan L, Guo Y, Aziz A, Meijer E. D^3: Differential testing of distributed deep learning with model generation. IEEE Trans Softw Eng 2025;51:38-52.
56. Almohaimeed M, Alyoubi R, Aljohani A, Alhaidari M, Albalwy F, Ghabban F, *et al*. Use of machine learning and deep learning in intrusion detection for IoT. Adv Int Things 2025;15:17-32.
57. 6 Use Cases for Distributed Deep Learning - Spectral. Available from: https://spectralops.io/blog/distributed-deep-learning [Last accessed on 2025 May 17].
58. DLion: Decentralized Distributed Deep Learning in Micro-Clouds. Available from: https://par.nsf.gov/servlets/purl/10292672 [Last accessed on 2025 May 17].
59. On Studying Distributed Machine Learning - Liberty University.Available from: https://digitalcommons.liberty.edu/cgi/viewcontent.cgi?article=2190&context=honors [Last accessed on 2025 May 17].
60. Filho CP, Marques E Jr., Chang V, Dos Santos L, Bernardini F, Pires PF, *et al*. A systematic literature review on distributed machine learning in edge computing. Sensors (Basel) 2022;22:2665.
61. Dean J. A golden decade of deep learning: Computing systems & applications. Daedalus 2022;151:58-74.
62. A Survey on Distributed Machine Learning - arXiv. Available from: https://arxiv.org/pdf/1912.09789 [Last accessed on 2025 May 17].
63. Optimal In-Network Distribution of Learning Functions for a Secure-by-Design Programmable Data Plane of Next-Generation Networks - arXiv. Available from: https://arxiv.org/pdf/2411.18384 [Last accessed on 2025 May 17].
64. A Comparative Analysis of Distributed Training Strategies for GPT-2 - arXiv. Available from: https://arxiv.org/html/2405.15628v1 [Last accessed on 2025 May 17].
65. Communication-Efficient Large-Scale Distributed Deep Learning: A Comprehensive Survey. Available from: https://arxiv.org/html/2404.06114v1 [Last accessed on 2025 May 17].
66. Towards Domain-Specific Network Transport for Distributed DNN Training - USENIX. Available from: https://www.usenix.org/system/files/nsdi24-wang-hao.pdf [Last accessed on 2025 May 17].
67. Deep Neural Network for Cyber Security Use Cases Vinayakumar R1, Barathi Ganesh HB1,2, Prabaharan Poornachandran3, Ana - arXiv. Available from: https://arxiv.org/pdf/1812.03519 [Last accessed on 2025 May 17].
68. Priority-Based Parameter Propagation for Distributed DNN Training - arXiv. Available from: https://arxiv.org/abs/1905.03960 [Last accessed on 2025 May 17].
69. A Hitchhiker's Guide On Distributed Training of Deep Neural Networks - arXiv. Available from: https://arxiv.org/pdf/1810.11787 [Last accessed on 2025 May 17].

70. Security for Distributed Deep Neural Networks Towards Data Confidentiality & Intellectual Property Protection - arXiv. Available from: https://arxiv.org/abs/1907.04246 [Last accessed on 2025 May 17].
71. MIT Open Access Articles Distributed Learning of Deep Neural Network Over Multiple Agents. Available from: https://dspace.mit.edu/bitstream/handle/1721.1/121966/1810.06060.pdf [Last accessed on 2025 May 17].
72. Research Publications. ChainOpera® Documentation. Available from: https://docs.chainopera.ai/federate/tech/papers [Last accessed on 2025 May 17].