JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Design and Implementation of an Artificial Intelligence-Driven Network Intrusion Detection System

Amit Bane
Dahanukar College of Commerce, Mumbai, Maharashtra
amitbane.phd@gmail.com

ABSTRACT

The current generation of Network Intrusion Detection Systems (IDS) exhibits poor adaptive learning capabilities, particularly when confronted with large-scale network traffic, resulting in diminished detection quality and efficiency. This paper details the design and implementation of an Artificial Intelligence (AI)-driven network IDS engineered to overcome these challenges by improving detection accuracy and real- time performance. The proposed system integrates three key technical components: one-hot encoding for robust data preprocessing and denoising, a One-Class Support Vector Machine (SVM) to optimise the adaptive learning process for new threats, and a sliding window method to achieve real-time data processing and model updates. Experimental results demonstrate the superiority of this approach. The performance of the proposed methods is, on average, 47% and 65% better than that of a convolutional RNN (as a baseline) and a novel Deep Neural Network (NDNN), respectively. Moreover, its mean response times are 25 and 41 seconds shorter. These results confirm that an AI-based IDS may represent a better and more efficient solution for network security these days.

KEYWORDS:
Network intrusion detection, artificial intelligence, one-class support vector machine, adaptive learning system design and implementation.

INTRODUCTION:

The concept of intrusion detection traces its origins to the late 1970s and early 1980s, during a period when computer networks were gradually becoming interconnected. One of the earliest discussions on intrusion detection was presented in James Anderson's seminal 1980 report *"Computer Security Threat Monitoring and Surveillance"*, which outlined the need for systems capable of recognizing anomalous behavior indicative of security breaches. Anderson emphasized that auditing patterns, user activities, and system logs could collectively support the identification of malicious activity.

In the late 1980s, intrusion detection models were based on statistical methods rather than on theoretical frameworks. An example of one such (and one of the first) systematic approaches to detecting intrusions was developed by Dorothy Denning and published in a paper titled, "A Taxonomy of Intrusion Detection Systems". In that paper, Ms. Denning provides a framework for identifying intrusions into computer systems using statistical comparison of expected vs. observed behaviour, as well as providing insight into several new concepts such as rule-based expert systems, statistical profiles and anomaly detection. Most current IDPS standards (Intrusion Detection and Prevention Systems) continue to include them.

By the 1990s, various companies were developing and marketing intrusion detection systems. In 1998, two early examples of a NIDS (Network-Based Intrusion Detection System) that performed signature-based packet inspection were introduced: Snort and Bro (now Zeek). These systems allowed an organisation to detect and respond to threats originating from its network traffic. Concurrently, a HIDS (Host -Based Intrusion Detection System) called Tripwire (1992) was developed to provide an organisation with the ability to monitor file integrity and detect intrusions at the system level.

As computer networks expanded in size and Cyber Attacks became more sophisticated in the early 2000’s, the need arose for automated responses to Intrusions. Subsequently, IDS evolved into an IDPS, which includes preventive actions (e.g., dropping a malicious packet) or blocking suspicious IP addresses or isolating damaged hosts).

During the rise of cloud computing in the mid-2000s and 2010s, traditional IDPS architectures had to deal with major problems caused by distributed systems, virtualization, and multi-tenancy, as well as the development of new and more complex attack techniques. This transformation was the groundwork of the coming era of AI- powered IDPS solutions that were capable of managing the intricate nature of cloud environments.

Cloud computing has progressed tremendously and now it is the core technology that underpins the modern enterprise environments, facilitates the digital transformation process, and also allows the creation and the mobilization of scalable services across distributed ecosystems. In fact, cloud platforms like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) have become crucial for organizations of every size where they are used for various important operations such as data storage, high-performance computing, machine learning workflows, and the delivery of software as a service (SaaS). The flexibility, cost -effectiveness, and worldwide reach of cloud infrastructures have completely changed the business practices concerning IT resource building and management. At the same time, however, the very factors making cloud computing appealing—multi-tenancy, virtualization, distributed architectures, and shared responsibility models—have also presented organizations with new and complex security challenges.

The growing sophistication of cyber threats, coupled with the dynamic and large-scale nature of cloud ecosystems, demands advanced security mechanisms capable of real- time monitoring, risk mitigation, and automated response.

Traditional intrusion detection and prevention systems (IDPS) were originally developed for on-premises, perimeter-based networks where traffic patterns were more predictable, data flows were relatively centralized, and system boundaries were clearly defined. These classical systems relied heavily on rule-based signatures, static thresholds, and manual configurations. While suitable in past decades, such systems struggle to keep pace with the evolving landscape of cloud-native threats that exploit distributed resources, decentralized architectures, ephemeral workloads, and encrypted communication channels. As attackers increasingly employ stealthy and polymorphic techniques—including zero-day exploits, ransomware variants, lateral movement across virtual machines, and API-based attacks—traditional IDPS models reveal evident limitations, such as high false-positive rates, limited scalability, and insufficient visibility into cloud workloads.

The integration of artificial intelligence (AI) and machine learning (ML) into IDPS design has emerged as a promising approach to addressing these shortcomings. AI- driven IDPS platforms can autonomously learn behavioral patterns, recognize anomalies, detect unknown threats, and adapt to continuously changing cloud environments. The network is now the vital conduit for information interchange and transmission in contemporary society due to the internet's quick expansion and widespread use. As a result, concerns about network security have grown in importance. A crucial issue that needs to be resolved is network intrusion, a type of hostile attack that seriously jeopardises corporate secrets, individual privacy, and even national security. Traditional Intrusion Detection Systems (IDS) are no longer adequate to meet modern network security requirements due to the increasing complexity and camouflage of network attack techniques. A more sophisticated paradigm is required since these outdated systems are unable to keep up with the variety and unpredictability of contemporary threats.

1.1. Problem Statement and Motivation

The central problem this research addresses is the inadequacy of traditional IDS in the face of modern network environments. Their incapacity to efficiently handle enormous data streams in real time and their inability to adjust to new or unexpected assault patterns are their two main drawbacks. These systems are susceptible to sophisticated threats and zero-day exploits since they usually rely on predetermined signatures or criteria. A potential remedy for these issues is artificial intelligence (AI). AI can automatically learn and extract complex feature representations from data because of its great flexibility and adaptive learning capabilities, which allow it to successfully respond to the dynamic nature of network incursions.

1.2. Proposed Solution and Contributions

In order to improve the accuracy and effectiveness of network threat detection, this study suggests an AI-driven network intrusion detection system. The following are the main contributions of the research:

1. **Improved adaptive learning capability:**

The system uses a One-Class Support Vector Machine (SVM) model in conjunction with one-hot encoding for data preparation. This method successfully tackles the problem of processing large data streams, a major flaw in conventional IDS, and improves the system's adaptability to unforeseen attack patterns.

2. Improved detection accuracy:

Across a range of network attack types, the suggested AI-based IDS obtains an average recognition rate of 89.2%. This performance outperforms current cutting-edge techniques, such as the convolutional RNN and NDNN models, allowing for more accurate detection and reaction to network threats.

3.Improved real-time performance:

The system can process data streams in real time by utilising sliding window technology. By greatly increasing the system's response time, this technique makes sure that possible security problems may be handled early on to reduce possible losses.

a. Paper Structure

The remainder of this paper is structured as follows.

- **Section 2** provides a review of related work in the field of intrusion detection.
- **Section 3** details the system design and methodology of the proposed AI-driven IDS.
- **Section 4** presents the experimental setup, results, and a comparative analysis.
- **Section 5** discusses the experimental findings and outlines the study's limitations.
- **Section 6** provides a conclusion that summarises the research.

2.Related Work

It is crucial to examine the body of research on network intrusion detection to appropriately contextualise this study. This section reviews earlier research using both conventional and AI-based methods. We can pinpoint the precise research gap that the suggested solution is intended to fill by looking at the development of IDS technology, especially the trade-off between real-time operational efficiency and detection accuracy for unknown threats.

2.1.Advances in Traditional and SDN-Based Intrusion Detection

Enhancing IDS capabilities through algorithmic and architectural improvements has been the subject of early and fundamental research. Sultana Nasrin, for example, investigated the use of Software Defined Networking (SDN) technology and showed how well it can identify and track security problems resulting from programmable network operations. Others, such as Sakr Mahmoud M, suggested an anomaly-based network intrusion detection system (IDS) that achieved excellent detection accuracy by employing a Support Vector Machine (SVM) classifier whose parameters were optimised using a particle swarm approach. These systems' main drawback is their dependence on specified feature libraries, even though they can withstand known attacks to some degree. This reliance makes it more difficult for them to identify and categorise unknown threats, which is a serious weakness in the present threat environment.

2.2.Machine Learning and Deep Learning in Intrusion Detection

As AI has developed, new opportunities for automatically identifying new threat traits have been made possible by Machine Learning (ML) and Deep Learning (DL) algorithms. With great promise, Gurung Sandeep created a DL-based IDS for unsupervised feature learning using a sparse autoencoder. By capturing both local and temporal information, Khan Muhammad Ashfaq created a hybrid system that used a convolutional Recurrent Neural Network (RNN) to detect malicious attacks with an accuracy of 97.75%. In a similar vein, Jia Yang created an IDS that achieved 99.9% accuracy on experimental datasets using a novel Deep Neural Network (NDNN) model. Although these advanced methods provide strong support for identifying unknown threats, they often suffer from low operational efficiency. When dealing with large- scale network traffic, they struggle to meet the real-time detection requirements of practical applications. This paper proposes a methodology designed to overcome these efficiency limitations while maintaining a high level of adaptive detection.

3.Proposed System and Methodology

This section provides a detailed technical exposition of the proposed AI-driven Network Intrusion Detection System. It covers the system's functional requirements, its overall architecture, and the specific algorithms and hardware components used for its implementation. The design aims to balance high detection accuracy with the real-time processing demands of modern network environments.

3.1. System Architecture

The overall architecture of the AI-driven network IDS is structured into three distinct layers. This layered approach ensures modularity, scalability, and a clear separation of concerns.

- **Presentation Layer**

This layer serves as the primary interface for user-system interaction. It is responsible for displaying real-time network status, issuing alarm notifications, and managing user access. To facilitate dynamic data interaction, the front-end is developed using the React framework. WebSocket technology is employed to establish effective, persistent communication between the presentation and business logic layers, ensuring that interface data is updated promptly to enhance system responsiveness and user experience.

- **Business Logic Layer**

This is the core of the system, housing the main processing and intelligence functions. Its responsibilities include request processing, service interfacing, security control, and real-time threat detection and response. The business logic layer leverages AI technology to model network traffic data, analyze historical patterns, and process real-time traffic information obtained from the data layer. This enables the system to discover potential attack methods and respond quickly to novel attack patterns.

- **Data Layer**

The data layer is responsible for the collection, storage, and maintenance of all network business data and logs. It consists of application and database servers responsible for data acquisition and processing as well as long-term data storage and analysis. Network traffic is captured in real time by collection devices and stored in a database. To manage and analyze the massive volumes of information, the data layer utilizes data warehouse technology to integrate and process data efficiently.

3.2 Hardware Design Components

The system's functionality is supported by three core hardware components designed for performance and reliability.

1. **Network Traffic Collection Device:**

This device is composed of network interfaces, deep packet inspection (DPI) capabilities, and a processing unit. It is configured on edge routing network nodes to monitor all incoming and outgoing packets in real time. The device extracts key business features—such as source address, destination address, and protocol type—and transmits the collected data to the business logic layer through a secure channel for rapid threat identification.

2. **Detection Servers:**

These servers are equipped with multi-core processors and large-capacity memory to handle intensive computational tasks. Storage is managed using a hybrid approach that combines high-speed solid-state drives with high-capacity hard disk drives. The servers analyze incoming data in real time using the pre- trained AI models, performing data cleaning and feature extraction to identify potential intrusions. When abnormal behavior is detected, an alarm message is generated and sent to the business logic layer.

3. **Controller:**

The controller acts as the central coordinator, managing front-end requests and back-end services through its task scheduling and data coordination components. It receives and parses requests, schedules back-end services, and optimizes resource allocation. The data coordination component manages the data flow between different layers, ensuring that when abnormal data is detected, the alarm mechanism is activated quickly and notifications are pushed to the user interface.

3.3 Implementation of the AI-Driven Detection Model

The implementation of the detection model follows a multi-stage process designed for accuracy and real-time performance.

First, the system performs data preprocessing. To handle raw network traffic data, character-based features are converted into numerical features using a **one-hot encoding** transformation method. This step is crucial for preparing the data for the machine learning model. Subsequently, the encoded feature values are scaled to a uniform range of [0, 1] using the Min-Max normalisation method, which prevents features with larger numeric ranges from disproportionately influencing the model. The normalisation formula is:

$$x' = (x - \min(x)) / (\max(x) - \min(x)) \quad (1)$$

where x is the original feature value and x' is the normalised feature value.

Next, the pre-processed data serves as the input vector for the core learning model, which is a **One-Class Support Vector Machine (SVM)**. Assuming that $x_1, x_2, \ldots, x_n$ are the training samples representing standard class data, the One-Class SVM optimisation problem is expressed as:

$$\min(w, \rho, \xi) \left[(1/2) * \|w\|^2 + (1/v * n) * \Sigma(\xi_i) - \rho \right] \ (2) \text{ subject to: } w \cdot \Phi(x_i) \geq \rho - \xi_i \, , \ \xi_i \geq 0, \ i=1,\ldots,n$$

The variables used in this optimisation problem are defined in the table below.

Sequence	Variables	Meaning
1	w	Weight vector
2	ρ	Bias direction
3	$\Phi(x_i)$	Mapping function

4	ξ_i	Slack variable
5	v	Punishment coefficient

To efficiently handle large-scale data, the system selects and annotates samples with multiple information features rather than processing each sample individually. A sample is flagged for labelling based on the following condition:

Δ = | 〈*v*<sub>i</sub>|*S*〉 - 〈*v*|*S*〉 | ≤ θ<sub>threshold</sub> (3)

where Δ represents whether the sample needs to be labelled, and *v*<sub>i</sub>, *v* represent attribute values.

Finally, to meet the requirements for real-time intrusion detection, the system employs a **sliding window method** to process and update the collected data regularly. Assuming a window size of *T*, the model's parameter *θ* is updated at each time step *t* according to the formula:

θ<sub>t+1</sub> = θ<sub>t</sub> + η (∇<sub>θ</sub>L(θ<sub>t</sub>; x<sub>t:t+T</sub>) - λ∇<sub>θ</sub>R(θ<sub>t</sub>)) (4)

where η is the learning rate, *L* is the loss function, and *R* is the regularization term. This iterative process allows the model to adapt to evolving traffic patterns and achieve real- time detection of intrusion threats.

4.Experimental Evaluation

To validate the effectiveness of the proposed AI-driven IDS, a series of experiments was conducted. These experiments were designed to evaluate the system's detection accuracy and real-time performance against established baseline models, thereby providing an objective measure of its capabilities in a controlled environment.

4.1 Experimental Setup and Dataset

The experimental environment utilised the CIC (Canadian Institute for Cybersecurity) Flow-Meter network security simulation tool to generate a diverse set of simulated attack traffic. This tool allows for the creation of realistic network flows representing both benign and malicious activities. For this study, seven specific types of attacks were generated in varying quantities to test the system's performance across a range of threat vectors.

Sequence	Amount	Specific type
1	108	Ping of death
2	89	Phishing
3	135	Zero-day
4	194	Advanced persistent threat
5	106	Cross-site scripting
6	121	Password attack
7	233	Cryptojacking

4.2 Baseline Models and Metrics

To objectively benchmark the performance of the proposed system, it was compared against two baseline models from existing literature: the **convolutional Recurrent Neural Network (RNN)** described in reference [10] and the **new Deep Neural Network (NDNN)** detailed in reference [11]. The evaluation was based on two key performance metrics:

- **Detection Accuracy (%):** The percentage of correctly identified intrusion attempts out of the total number of attacks.
- **Real-time Performance (s):** The average response time, measured in seconds, from the moment an attack is initiated to when the system detects and reports it.

4.3 Results:

Detection Accuracy: The analysis of detection accuracy, based on 10 repeated experiments for each attack type, revealed that the proposed system consistently outperformed the baseline models. The proposed system achieved an average accuracy of **89.2%** across all attack types. In comparison, the convolutional RNN-based system achieved an average accuracy of 84.5%, and the NDNN-based system achieved an average of 82.7%.

This means the proposed system's accuracy is **4.7% higher** than the convolutional RNN and **6.5% higher** than the NDNN. This superior performance is attributed to the One- Class SVM's unsupervised adaptive learning method, which allows the system to mine more distinctive features from large-scale traffic data and characterize more dimensional information, thereby improving its ability to distinguish between normal and malicious activity.

4.4 **Results: Real-time Performance** The real-time performance evaluation also demonstrated a significant advantage for the proposed system. The suggested AI-driven

system took about 12.0 seconds on average to identify and react to different types of threats. The baseline systems, on the other hand, were significantly slower, with the NDNN averaging 16.1 seconds and the convolutional RNN averaging 14.5 seconds.

In comparison to the convolutional RNN and the NDNN, the suggested method decreased the average reaction time by 2.5 and 4.1 seconds, respectively. The system's efficient data preprocessing, which lessens the effect of noise on detection, and the sliding window method's real-time model updates are directly responsible for this increased efficiency. Together, these characteristics increase the speed at which data is processed and the detection process's overall real-time performance.

5.Discussion

In this section, the experimental results from the previous section are interpreted, and their wider implications for the field of network security are analysed. It also offers a critical assessment of the study's shortcomings and suggests possible directions for further investigation to expand on these conclusions.

5.1 Interpretation of Findings

The key outcomes of this study demonstrate the efficacy of an AI-driven approach to network intrusion detection. The strategic combination of one-hot encoding for data preparation, a One-Class Support Vector Machine for adaptive learning, and a sliding window for real-time updates resulted in a system with an average detection accuracy of 89.2% and an average response time of 12 seconds. The significance of these results lies in the system's ability to effectively extract meaningful features from massive data streams and adapt to new, previously unseen attack patterns in real time. This capability directly addresses the core weaknesses of traditional IDS and some complex deep learning models, thereby providing a robust enhancement to network security infrastructure.

5.2 Limitations and Future Research

Despite the promising results, it is important to evaluate the limitations of this research critically. The CIC FlowMeter program was used to create traffic for the trials, which were carried out in a simulated setting. Although this configuration is helpful for controlled benchmarking, it is not validated on extensive real-world network data. In reality, network attack behaviours are far more varied and complicated, which could require the system to execute at a higher level.

A number of potential study directions are suggested in light of these restrictions. on order to evaluate the system's efficacy against real-world threats, the next step should be to validate its performance on a live, operational network. Optimising the algorithm for more intricate and dynamic network environments should be the main goal of future research. This could entail investigating ways to apply the model to higher-dimensional data in order to increase its resilience and capacity for generalisation. The final point this article will make is that addressing these limits will be essential to creating a more robust system.

6.Conclusion

Advanced intrusion detection systems are now crucial for preserving information security and safeguarding digital assets in an increasingly complicated network environment. This paper presented the design and implementation of an AI-driven IDS aimed at improving detection efficiency and quality. By integrating an unsupervised adaptive learning approach via a One-Class SVM with real-time window updates, the system effectively improves the detection accuracy of network attacks while simultaneously enhancing data processing speed. The experimental findings confirmed that this approach yields significant improvements in both detection accuracy and real- time performance compared to established baseline models. However, the study has a key limitation: the system's robustness in dynamically changing network environments was not fully considered. Looking ahead, it will be necessary to investigate the impact of network traffic dynamics on system performance to continuously refine and improve the model. Such efforts will contribute to the high-quality development of network security and help safeguard our increasingly connected world.

References

[1] Thapa, Niraj, Liu Zhipeng, KC Dukka B, Gokaraju Balakrishna, Kaushik Roy. "Comparison of machine learning and deep learning models for network intrusion detection systems." Future Internet 12.10 (2020): 167-172. DOI:10.3390/fi12100167

[2] Apruzzese, Giovanni, Mauro Andreolini, Luca Ferretti, Mirco Marchetti, Michele Colajanni. "Modeling realistic adversarial attacks against network intrusion detection systems." Digital Threats: Research and Practice (DTRAP) 3.3 (2022): 1-19. DOI: 10.1145/3469659

[3] Zhang, Wenjie, Dezhi Han, Kuan-Ching Li & Francisco Isidro Massetto. "Wireless sensor network intrusion detection system based on MK-ELM." Soft Computing 24.16 (2020): 12361-12374. DOI: 10.1007/s00500-020-04678-1

[4] Sultana, Nasrin, Naveen Chilamkurti, Wei Peng & Rabei Alhadad. "Survey on SDN based network intrusion detection system using machine learning approaches." Peer-to- Peer Networking and Applications 12.2 (2019): 493-501.DOI: 10.1007/s12083-017- 0630-0

[5] Alzahrani, Abdulsalam O., and Mohammed JF Alenazi. "Designing a network intrusion detection system based on machine learning for software defined networks." Future Internet 13.5 (2021): 111-128. DOI: 10.3390/fi13050111

[6] Sakr, Mahmoud M., Medhat A. Tawfeeq, and Ashraf B. El-Sisi. "Network intrusion detection system based PSO-SVM for cloud computing." International Journal of Computer Network and Information Security 14.3 (2019): 22-29. DOI: 10.5815/ijcnis.2019.03.04

[7] He, Ke, Dan Dongseong Kim, and Muhammad Rizwan Asghar. "Adversarial machine learning for network intrusion detection systems: A comprehensive survey." IEEE Communications Surveys & Tutorials 25.1 (2023): 538-566. DOI: 10.1109/COMST.2022.3233793

[8] Azizan, Adnan Helmi, Salama A. Mostafa, Aida Mustapha, Cik Feresa Mohd Foozy, Mohd Helmy Abd Wahab, Mazin Abed Mohammed, et al. "A machine learning approach for improving the performance of network intrusion detection systems." Annals of Emerging Technologies in Computing (AETiC) 5.5

(2021): 201-208. DOI: 10.33166/AETiC.2021.05.025

[9] Gurung, Sandeep, Mirnal Kanti Ghose, and Aroj Subedi. "Deep learning approach on network intrusion detection system using NSL-KDD dataset." International Journal of Computer Network and Information Security 11.3 (2019): 8-14. DOI: 10.5815/ijcnis.2019.03.02

[10] Khan, Muhammad Ashfaq. "HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system." Processes 9.5 (2021): 834-857. DOI:10.3390/pr9050834

[11] Jia, Yang, Meng Wang, and Yagang Wang. "Network intrusion detection algorithm based on deep neural network." IET Information Security 13.1 (2019): 48-53. DOI: 10.1049/iet-ifs.2018.5258

[12] Binbusayyis, Adel, and Thavavel Vaiyapuri. "Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM." Applied Intelligence 51.10 (2021): 7094-7108. DOI: 10.1007/s10489- 021-02205-9

[13] Corthis, P.B., Ramesh, G.P., García-Torres, M. and Ruíz, R., 2024. Effective Identification and Authentication of Healthcare IoT Using Fog Computing with Hybrid Cryptographic Algorithm. Symmetry, 16(6), p.72.