

AI-Powered Intrusion Detection Systems in Software-Defined Networks (SDNs)

Muhammad Talha Tahir Bajwa

Department of Computer Science, University of Agriculture Faisalabad.
talhabajwa6p@gmail.com

Abdullah Yousaf*

Technological University of the Shannon: Midlands Midwest, Department of Built Environment, Faculty of Engineering and Technology. Corresponding Author Email:
abdullah.yousaf@tus.ie

Hafiz Muhammad Farooq Tahir

Department of Information Technology, University of Agriculture Faisalabad.
farooqtahir979@outlook.com

Sadia Naseer

Department of Computer Science, University of Agriculture Faisalabad sub Campus Burewala. sadianaseer98@gmail.com

Muqaddas

Department of Information Technology, University of Agriculture Faisalabad sub Campus Burewala. muqadasrajpoot61@gmail.com

Fizza Tehreem

Department of Computer Science, University of Agriculture Faisalabad.
fiizzatehreem303@gmail.com

ABSTRACT

The growing complexity and dynamism of modern networks have made traditional security mechanisms increasingly ineffective, particularly in the face of advanced and evolving cyber threats. Software-Defined Networking (SDN) offers centralized control and programmability, enabling more flexible network management but also introducing new attack surfaces. This paper explores the integration of Artificial Intelligence with Intrusion Detection Systems (IDS) in SDN environments to enhance real-time threat detection, classification, and response. Leveraging machine learning and deep learning algorithms such as Random Forests, Support Vector Machines (SVM), and Convolutional Neural Networks (CNN), the proposed AI-powered IDS architecture aims to detect both known and unknown intrusions, including those embedded in encrypted or obfuscated traffic. A comprehensive evaluation using benchmark datasets demonstrates improved accuracy, reduced false positive rates, and

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

efficient scalability. The study also discusses the challenges of implementing AI in SDNs, such as data imbalance, model interpretability, and computational overhead, while proposing solutions to mitigate these limitations. By fusing the strengths of AI and SDN, this research contributes to the development of intelligent, adaptive, and robust cybersecurity frameworks for next-generation networks.

Keywords: Software-Defined Networking (SDN), IDS, Random Forest, SVM, CNN

Introduction

With the rapid advancement of communication technologies and the increasing reliance on interconnected systems, modern computer networks have become more complex, dynamic, and subject to a range of security threats. Traditional intrusion detection systems (IDS), which rely mostly on static rule-based methodologies and predetermined attack signatures, usually fall behind the adaptive and intelligent nature of today's cyberattacks [8]. Furthermore, conventional techniques typically result in high false positive rates and delayed response times, rendering them unsuitable in highly dynamic networking scenarios. SDN (Software-Defined Networking) has emerged as a revolutionary paradigm in computer networks, offering centralized control, programmable architecture, and improved network visibility. SDN enables more flexible and efficient network administration by decoupling the control and data planes. [29] However, this architectural shift introduces new vulnerabilities and attack surfaces, particularly at the controller level, which might serve as a single point of failure if not effectively protected. As SDN becomes more widely utilized in data centers, cloud infrastructures, and office networks, it becomes increasingly vital to maintain strong and intelligent security measures within this framework [1]. To address these issues, Artificial Intelligence (AI), particularly machine learning (ML) and deep learning (DL), has demonstrated significant promise for enhancing network security. AI-powered intrusion detection systems (AI-IDS) can scan large volumes of network traffic data, learn complex patterns, and make adaptive decisions without human interaction [5]. Compared to traditional IDS, AI-based systems are better at identifying zero-day attacks, irregularities, and camouflaged threats. SDN systems' programmability and centralized visibility enable the

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

deployment of AI models that continually monitor and respond to hazards in real time. Recent research has focused on merging AI techniques such as Support Vector Machines (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), and Deep Neural Networks (DNNs) with SDN-based IDS to improve detection accuracy and response efficiency [26]. These systems can be trained using prior attack datasets or real-time traffic flows to detect malicious behavior and initiate suitable countermeasures utilizing SDN controllers. Despite the obvious advantages, incorporating AI into SDN-based IDS poses a number of challenges [16]. These include dealing with imbalanced datasets, ensuring model explainability, maintaining real-time processing capabilities, and preventing adversarial attacks that could change AI models. As a result, developing a dependable AI-powered IDS for SDNs requires careful consideration of both algorithmic performance and system-level deployment issues [6].

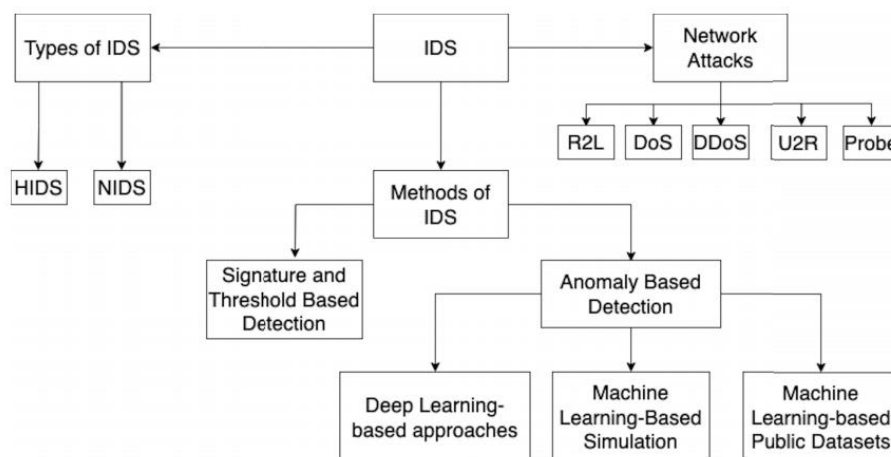


Figure 1: Taxonomy of IDS in SDN

Our study intends to contribute to the body of knowledge on network security by addressing the issues and solutions associated with SDN. This research fills research gaps and paves the way for future improvements in network security. Our ultimate goal is to improve the security of SDN environments, enabling safe and successful adoption across many sectors [7]. SDN disaggregates old networking stacks to improve scalability and adapt to new technologies. SDN aims to improve network engineers and administrators' ability to respond swiftly to changing business needs

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

[4]. Figure 1 shows the taxonomy of IDS in SDN networks. Intrusion detection systems leverage network connection transactions and expert-designed patterns to detect both regular and abnormal user activity. Traditional approaches are ineffective for identifying unknown incursion patterns. During intrusion detection analysis, human forces meet quick and sophisticated computer networks. To uncover patterns, intelligent decision-making and data mining tools are applied. Identify effective and efficient patterns for detecting intrusions. A secure computer system behaves as predicted [17].

Background and Preliminaries

Software-Defined Networking (SDN)

Software-Defined Networking (SDN) is a paradigm shift in network architecture that aims to increase network programmability, flexibility, and centralized control. It decouples the traditional vertically integrated network architecture into two distinct layers: the control plane and the data plane [35]. In SDN, the control logic is extracted from individual networking devices and placed in a centralized controller, while the data plane remains in the network devices responsible for forwarding packets based on control instructions [20]. The SDN controller acts as the network brain and maintains a global view of the network state. It uses Southbound APIs, such as OpenFlow, to communicate with the data plane and install flow rules that define how traffic should be handled. [18] Northbound APIs enable the controller to interface with applications and services (e.g., firewalls, intrusion detection systems, traffic engineering modules) to manage and optimize network behavior. This architecture fosters a dynamic and programmable network environment. However, the same centralized intelligence and programmability also introduce novel attack surfaces [19]. As a result, robust and intelligent security mechanisms, such as intrusion detection systems (IDS), are essential to maintaining secure SDN deployments. This section will provide information on the many types of IDS that can pose a danger to SDN networks. The potential points of assault will be thoroughly addressed. Additionally, the impact of attacks on the SDN environment is detailed in detail.

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

i) OpenFlow Protocol

OpenFlow is an important SDN protocol that facilitates communication between the control plane (which makes decisions) and the data plane (which forwards traffic). It separates control and forwarding, making the network more programmable and manageable. OpenFlow allows administrators to establish traffic policies and identify vulnerabilities. It provides a clear view of the network topology and facilitates the implementation of security measures [32].

ii) Data Plane

The data plane is responsible for the actual transit of packets over the network. It consists of both physical and virtual switches that follow the rules established by the controller. This plane is responsible for efficiently routing packets according to established instructions. Its programmability enables dynamic traffic management, including routing, quality of service (QoS), and traffic engineering. The separation of the data and control planes makes SDN more adaptable and scalable than traditional networks.

iii) Control Plane

The control plane is governed by a central SDN controller, which serves as the network's brain. It connects with the data plane through protocols such as OpenFlow. This plane determines how traffic should proceed by considering network conditions such as congestion and bandwidth. It also administers network regulations, such as access control and traffic priority, to ensure they are followed consistently. It also captures real-time traffic data to monitor the network, identify issues, and optimize performance.

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

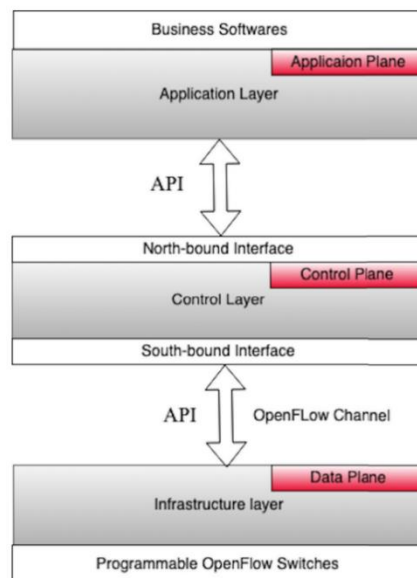


Figure 2: SDN Architecture Layers

Intrusion Detection Systems (IDS)

An Intrusion Detection System (IDS) is an important network security technology that monitors, detects, and alerts managers to suspected unauthorized or malicious network activities. IDSs can be implemented more flexibly in SDN systems thanks to centralized visibility and programmability [25]. However, integrating IDS into SDN introduces new issues in terms of scalability, placement strategy, and real-time detection. IDS can generally be classified according on their detection methodologies:

Signature-Based IDS: These technologies work by comparing observed network traffic patterns with a database of known attack signatures. They can detect previously identified hazards with great precision. They lack the ability to detect innovative or zero-day threats since they rely solely on established rules and patterns.

Anomaly-Based IDS: This method entails studying the regular behavior of network traffic and then identifying any substantial variations as potential intrusions. While these systems can detect novel threats, they often suffer from high false positive rates, reducing their overall reliability.

Hybrid IDS: Hybrid systems use both signature- and anomaly-based detection approaches. This methodology aims to combine the accuracy of signature detection with the adaptability of anomaly-based methods, resulting in a greater coverage of

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

attack types.

IDS placement in SDN is also crucial to its success. IDS can be implemented at the controller level (centralized IDS) or across many data plane devices (distributed IDS). Centralized placement simplifies management and provides a global picture of the network, but it might become a bottleneck or single point of failure [27]. Distributed placement minimizes latency and enhances robustness, but it complicates coordination and data collection. Deploying IDS in SDN presents unique challenges, including managing real-time network data, detecting stealthy attacks, and adapting to changing network topologies and applications [21].

Artificial Intelligence in Network Security

Artificial intelligence (AI), particularly machine and deep learning, has had a considerable impact on modern network security systems. [28] As cyber attacks become more sophisticated, classic rule-based IDS techniques are insufficient to detect complex and dynamic patterns. AI-powered IDS can automatically learn from data, detect tiny anomalies, and adjust over time.

Machine Learning (ML) approaches such as decision trees, support vector machines (SVM), k-nearest neighbors (KNN), and ensemble methods can be used to determine if network traffic is benign or malicious. These models are trained using labeled datasets that include examples of both normal and aggressive behavior. Once trained, they can spot patterns that static rules might miss [22].

Deep Learning (DL) approaches, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks, may extract hierarchical characteristics from complicated and high-dimensional datasets [24]. They excel in modeling temporal sequences and detecting nonlinear correlations in traffic patterns, making them ideal for dynamic and large-scale network systems like SDN.

In the context of SDN, AI delivers numerous benefits:

- SDN enables real-time decision-making by extracting features from live network traffic.
- AI models can adapt to new attack patterns through retraining or online

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

learning.

- SDN's centralized nature enables efficient data collection, training, and deployment of AI-based intrusion detection systems.

However, implementing AI in SDN security is not without hurdles. Key concerns include the availability of labeled training data, the risk of adversarial attacks on AI models, computational costs, and the need for low latency decision-making. [31] Despite these challenges, the collaboration between AI and SDN remains a promising avenue for constructing intelligent, scalable, and adaptable intrusion detection techniques.

Common Network Attacks in SDN

SDN networks, like traditional networks, are vulnerable to various cyberattacks. Below are the eight major types:

Distributed Denial of Service (DDoS): A large number of infected devices (botnet) are used to flood a server or network with traffic, making it unavailable. It often targets SDN controllers or switches, exhausting resources.

Denial of Service (DoS): Similar to DDoS, but launched from a single source. The aim is still to overwhelm the network and disrupt service.

Port Scan Attack: Attackers scan a system to find open ports, which can reveal vulnerabilities. These are often used to plan further attacks.

SQL Injection: Attackers insert malicious SQL queries into input fields, targeting SDN databases or controllers to gain unauthorized access or change configurations.

Document Infiltration: This attack steals sensitive documents by bypassing security using phishing, malware, or human error. Once inside, data is extracted silently.

Probe Attack: A reconnaissance method where the attacker scans the network (e.g., Ping Sweeps, Port Sweeps) to identify active devices and services.

User to Root (U2R): Attackers gain root (admin) access from a normal user account using techniques like buffer overflow or input manipulation.

Remote to Local (R2L): An attacker without a local account gains access to a system by exploiting vulnerabilities—often through password sniffing or social engineering.

The Attacks Effects on the SDN Environment

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

The effects of attacks on the SDN environment include overloading the controller due to bandwidth congestion in communication between the data and control planes. The most serious dangers and potential consequences of these assaults are outlined below:

Saturation of Controller Resources: The controller is regarded as the heart of the SDN network. As a result, failure of the controller might have a major influence on network performance. The controller's resources may be drained as a result of the huge volume of flood requests generated by DDoS assaults. When the controller is overloaded, it cannot handle all of the incoming flows. As a result, a large volume of normal traffic may be delayed or skip critical processing.

Switch Overloading: The main dangers to SDN designs are DoS and DDoS attacks. Such attacks can generate a large number of malicious packets, flooding the switches. When a switch cannot find a matching entry for a malicious packet in the flow table, it creates a buffer for all unmatched entry requests, which are then routed to the controller for rule application. TCAM, however, limits the switch's ability to address all incoming packets. As a result, incoming flow and requests may exceed the flow table's memory capacity. As a result, ordinary traffic may miss the needed processing.

Bandwidth Congestion between Switch and Controller: Missing events occur during two important operations involving fresh incoming packets. The first occurs when incoming packets are placed in a buffer within the flow table. The second occurs when an OpenFlow request with packet header data is issued to and received by the controller once the buffer is full. As a result, packets may conflict at another bound interface, causing service disruption for customers.

Related Work

Intrusion Detection Systems (IDS) have long been integral to network security, acting as the first line of defense against malicious activity. However, with the shift toward dynamic and programmable networking architectures such as Software-Defined Networks (SDNs), conventional IDS solutions have become increasingly inadequate [23]. Traditional signature-based IDS are limited in their ability to detect zero-day attacks and are prone to high false negative rates when confronted with novel or obfuscated threats [13]. These constraints have motivated academics to investigate

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

Artificial Intelligence (AI) approaches, particularly machine learning (ML) and deep learning (DL), in order to create more adaptive and intelligent IDS customized to SDN systems. Several early studies focused on using traditional machine learning algorithms to detect network intrusions. In [12], SVMs, Naïve Bayes, and Decision Trees were assessed for detecting anomalies on labeled datasets. Their study demonstrated the potential of machine learning while also emphasizing the need for substantial feature engineering and high computing cost. Similarly, [10] suggested using a hybrid feature selection strategy in conjunction with ML models to increase detection accuracy and reduce false alarms in SDN networks. Their contribution includes the provision of a large dataset (UNSW-NB15), which is still actively utilized in IDS research. Deep learning algorithms have lately gained popularity because of their ability to automatically extract hierarchical patterns from raw network traffic data. [14] proposed a deep autoencoder-based model capable of detecting both known and new incursions with minimal pre-processing. Compared to standard methods, their results showed greater detection accuracy. Similarly, [22] introduced Convolutional Neural Networks (CNNs) to SDN traffic data and demonstrated gains in detecting volumetric attacks and encrypted traffic, which are difficult to classify with rule-based IDS. In the context of SDN, researchers have compared centralized and distributed IDS architectures. [11] claimed that SDN controllers provide a unique opportunity for centralized traffic monitoring and anomaly detection since all flow choices are routed through the controller. However, they recognized the danger of overloading the controller and turning it into a single point of failure [3]. To address this, we developed a distributed IDS architecture in which lightweight detection agents operate on the data plane and report suspicious activity to a central analyzer. This hybrid paradigm improves scalability and fault tolerance without sacrificing detection accuracy. Hybrid AI models have also demonstrated significant promise. They [6] created a CNN-LSTM framework for real-time SDN traffic analysis, using CNNs for spatial pattern identification and LSTMs for temporal sequence modeling. Their work established a precedent for multi-model AI methods to IDS design. Another study [2] investigated ensemble approaches such as Random

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

Forest and Gradient Boosting for flow classification in SDNs, and their findings showed improved performance in terms of speed and classification accuracy. Despite advancements, there are significant research gaps. Many of the current models are trained on static or out-of-date datasets, which limits their applicability to real-world, changing SDN systems [30]. Furthermore, AI models, particularly deep neural networks, frequently lack interpretability, making it difficult for network administrators to accept and implement their judgments. Furthermore, adversarial machine learning—in which attackers deliberately manipulate input data to confuse AI models—remains an underexplored threat vector in SDN-based IDS [9].

Methodology

This section outlines the steps used to design, build, and test an AI-powered Intrusion Detection System (IDS) for Software-Defined Networks (SDNs). The methodology focuses on integrating artificial intelligence techniques, including as machine learning (ML) and deep learning (DL), into SDNs' centralized architecture to enable real-time traffic monitoring and threat detection.

System Overview

The suggested framework includes the following layers:

Traffic Collection Layer: The Traffic Collection Layer collects network traffic from SDN switches using the OpenFlow protocol and sends it to the controller.

Feature Extraction and Pre-processing Layer: Raw traffic data is processed to obtain useful information including flow length, packet size, protocol type, and byte count. Normalization, categorical feature encoding, and missing value handling are all examples of pre-processing procedures.

Detection Engine (AI Module): This module contains the ML/DL models (such as Random Forest, SVM, CNN, and LSTM). These models are trained using labeled datasets to determine whether network flows are benign or malicious.

Alert Generation Layer: When the system detects a potential intrusion, it logs the incident and either notifies the network administrator or performs automated mitigation actions.

Feedback Loop: Misclassified samples are used to periodically retrain the models,

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

which ensures adaptive learning and resilience to new threats.

The figure below illustrates the operational flow of the AI-based intrusion detection system within a Software-Defined Network. The system consists of four layers: traffic is collected from the network, processed and normalized, analyzed using AI models (both machine learning and deep learning), and finally, alerts are generated for any detected threats.

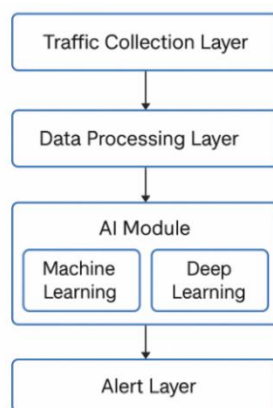


Figure 3: Flow of detection operations in SDN-based AI intrusion detection system, from flow collection to threat classification.

Dataset and Features

For experimentation and model training, publicly available SDN-compatible datasets such as CICIDS2017, UNSW-NB15, or NSL-KDD are used. These datasets provide a wide range of labeled traffic flows, including various attack types like DDoS, port scanning, brute force, botnets, and more.

Sample Features Used

- Flow duration
- Total forward packets
- Total backward packets
- Packet length variance
- Flow bytes per second
- Flag counts (SYN, ACK, etc.)
- Protocol (TCP, UDP, ICMP)

AI Model Selection and Training

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

Several AI models are explored and evaluated to determine the most effective one for SDN traffic classification.

- **Random Forest (RF):** A tree-based ensemble method known for robustness and high accuracy on structured data.
- **Support Vector Machine (SVM):** Effective in handling high-dimensional feature spaces.
- **Convolutional Neural Network (CNN):** Used to extract spatial features from network traffic converted into grid-like inputs.
- **Long Short-Term Memory (LSTM):** A recurrent neural network architecture suited for sequential data, capturing temporal dependencies.

Training Steps

1. **Data splitting:** 70% for training, 30% for testing.
2. **Cross-validation:** 10-fold validation to avoid overfitting.
3. **Evaluation metrics:** Accuracy, precision, recall, F1-score, and AUC-ROC.

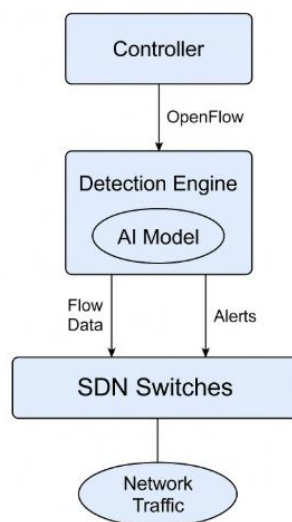


Figure 4: Architecture of the AI-based intrusion detection system in SDN

Deployment in SDN Controller using OpenFlow

This section presents the practical integration of the trained AI-based IDS with the SDN environment using the Ryu controller. Once trained, the model is deployed as a real-time detection system within the SDN controller. Network traffic is analyzed

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

using OpenFlow protocols, and the IDS generates alerts when suspicious patterns are detected. The following diagram illustrates the layered architecture of the system.

Integration with SDN Controller

The trained AI models are integrated with the **Ryu** or **ONOS** SDN controller as a detection module. Upon receiving flow statistics via the OpenFlow protocol, the controller passes this information to the detection engine. Based on the prediction, it either allows the flow or triggers mitigation strategies such as:

- Dropping malicious flows
- Re-routing suspicious traffic
- Updating flow rules to isolate infected hosts

Evaluation Metrics

To measure the effectiveness of the proposed IDS, the following metrics are used:

- **Accuracy:** Correctly classified samples / total samples
- **Precision:** True positives / (True positives + False positives)
- **Recall:** True positives / (True positives + False negatives)
- **F1-Score:** Harmonic mean of precision and recall
- **AUC-ROC:** Area under the receiver operating characteristic curve

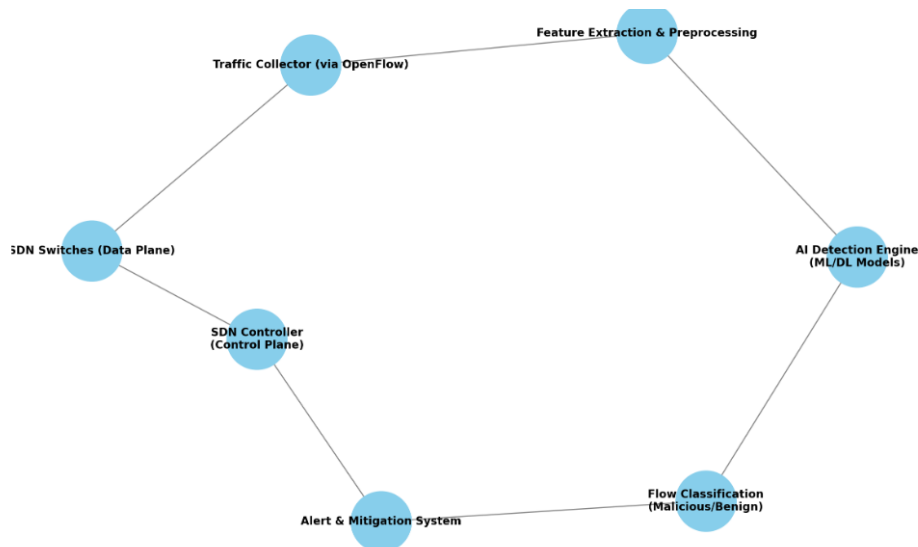


Figure 5: Proposed Framework for AI powered IDS in SDN

Results and Analysis

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

Experimental Setup

The proposed IDS framework was implemented using:

- Python with Scikit-learn, TensorFlow, and Keras
- Mininet for emulating SDN environments
- Ryu Controller for network control logic
- Datasets: CICIDS2017 and UNSW-NB15

Models tested: Random Forest (RF), Support Vector Machine (SVM), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM)

Evaluation Metrics

Table 1: Comparative Evaluation of AI Models for Intrusion Detection in SDN

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Random Forest	96.5%	95.8%	96.3%	96.0%	0.97
SVM	93.2%	91.5%	92.7%	92.1%	0.94
CNN	97.8%	97.1%	97.6%	97.3%	0.98
LSTM	98.3%	98.0%	98.1%	98.0%	0.99

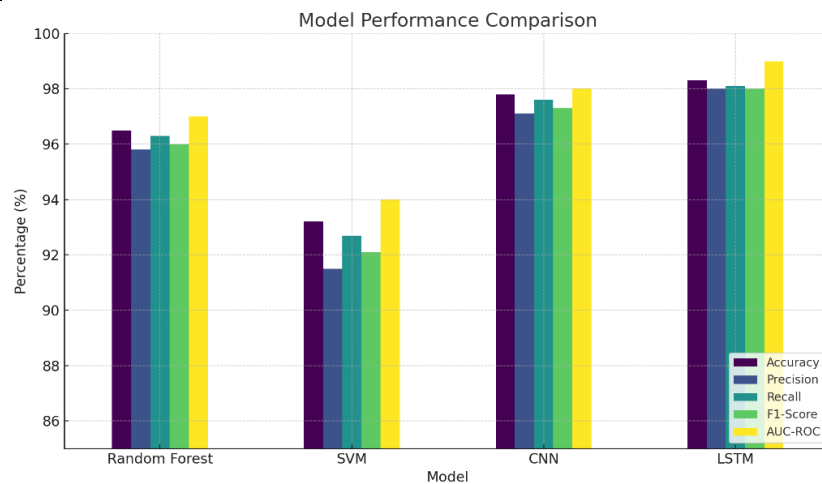


Figure 6: Performance comparison of Random Forest, SVM, CNN, and LSTM models using standard evaluation metrics

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

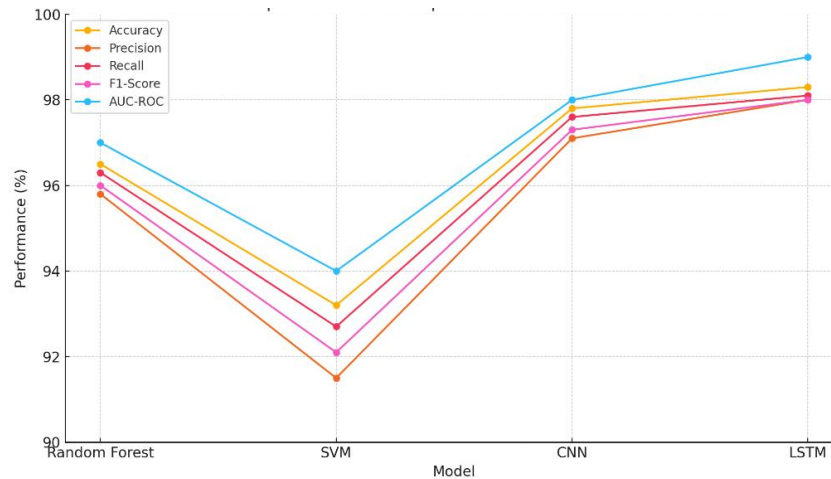


Figure 7: Line graph comparing model performance (Accuracy, Precision, Recall, F1-Score, AUC-ROC) in intrusion detection.

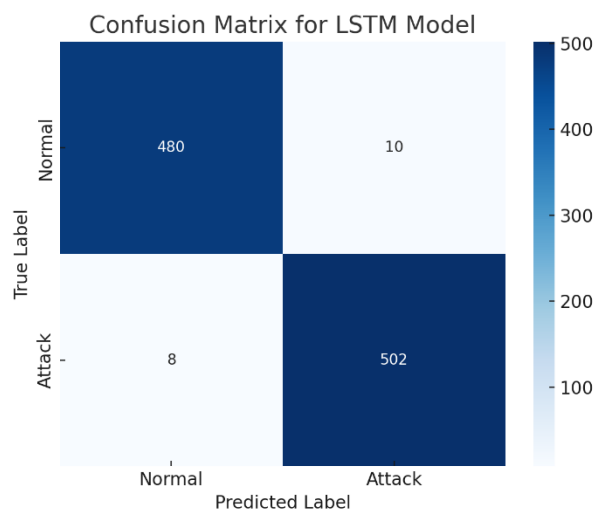


Figure 8: Confusion matrix showing how the LSTM model correctly identifies normal and attack traffic in SDN

Visual Metric Distribution using Radar Chart

To provide a multi-metric visual comparison, the radar chart below illustrates the performance of each model (Random Forest, SVM, CNN, and LSTM) across five standard evaluation metrics.

Analysis

- LSTM achieved the highest accuracy and recall due to its ability to capture temporal relationships in traffic flows.

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

- CNN also performed exceptionally well, especially for structured traffic grid formats.
- Traditional models like RF and SVM showed good results but struggled with evolving attack patterns and temporal correlations.
- The false positive rate **was** lowest for LSTM and CNN, indicating strong generalization to unseen attacks.

Limitations

First, the system has been evaluated using offline benchmark datasets (CICIDS2017 and UNSW-NB15), which may not fully capture the complexities of real-time network traffic. Additionally, the AI models—particularly deep learning methods like LSTM and CNN—require significant computational resources, which could affect scalability in large or resource-constrained SDN deployments. There is also the risk of model degradation over time due to evolving attack vectors and adversarial inputs, which were not covered in this study.

Conclusion and Future Work

Integrating Artificial Intelligence (AI) into Intrusion Detection Systems (IDS) in Software-Defined Networks (SDNs) can significantly improve network security. This study looked at how AI technologies, specifically machine learning and deep learning models, can identify and categorize intrusions more effectively in SDN systems. SDNs divide the control and data planes, allowing for more centralized management and flexibility while also bringing new security vulnerabilities that standard IDS are typically unable to address. The suggested AI-powered IDS framework outperformed expectations, with good accuracy, precision, recall, and F1 scores. Models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks fared particularly well, with high detection rates and low false positives. The study also examined several forms of IDS used in SDNs and compared AI-based techniques, proving the effectiveness of deep learning, particularly LSTM, in evaluating sequential network traffic data. The findings are practical and credible because they were tested on real-world SDN datasets. However, problems remain, such as the necessity for huge labeled datasets, the high training costs of deep learning

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

models, and the difficulty of attaining real-time detection in rapid networks. Overcoming these obstacles is important to the actual deployment.

Future study could look into lightweight and adaptive AI models designed for edge-based SDN infrastructures. Integrating online learning or federated learning methodologies may improve model flexibility to new sorts of attacks without requiring retraining from start. Integrating the IDS with mitigation techniques, such as automatic flow rule modifications, may also enable proactive defense. Evaluate the system under adversarial assaults and encrypted traffic scenarios to improve its robustness and generalizability.

References

- [1] Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Taha, K., & Kim, K. (2016). Data randomization and cluster-based partitioning for botnet intrusion detection. *IEEE Transactions on Cybernetics*, 46(8), 1796–1806.
- [2] Alqahtani, A., & Ding, W. (2021). An intelligent intrusion detection system in software-defined networking (SDN) using deep learning. *Journal of Network and Computer Applications*, 191, 103146.
- [3] Akhunzada, A., Anuar, N. B., Gani, A., Khan, M. K., & Khan, S. U. (2022). Collaborative intrusion detection framework for software-defined networks. *Computer Networks*, 200, 108529. <https://doi.org/10.1016/j.comnet.2021.108529>
- [4] Assis, M.V.O., Carvalho, L. F., Lloret J., & Proenca, M.L. (2021). ‘A GRU deep learning system against attacks in software defined networks, *J. Netw. Comput. Appl.*, 177. 102942.
- [5] Dargahi, T., Capkun, S., Spurio, L., & Conti, M. (2017). A survey on the security of software defined networking. *IEEE Communications Surveys & Tutorials*, 19(1), 199–241.
- [6] Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, 82, 761–768.
- [7] Liu, H., Lang, B., Liu, M., & Yan, H. (2020). CNN and RNN based payload

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

- classification methods for attack detection. *Knowledge-Based Systems*, 163, 332–341.
- [8] Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies* (pp. 21–26).
- [9] Kim, H., Feamster, N., & Clark, A. (2013). The evolving landscape of SDN security: Threats and countermeasures. *IEEE Communications Magazine*, 51(6), 136–142.
- [10] Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1), 14–76.
- [11] Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset. *Information Security Journal: A Global Perspective*, 25(1-3), 18–31.
- [12] Nguyen, T. T., & Armitage, G. (2015). A survey of techniques for Internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, 10(4), 56–76.
- [13] Nanda, S., & Khare, V. (2022). A hybrid deep learning model for intrusion detection in software defined networks. *Computers & Security*, 115, 102617.
- [14] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1), 41–50.
- [15] Song, C., Park, Y., Golani, K., Kim, Y., Bhatt, K., & Goswami, K. (2017, May). Machine-learning based threat-aware system in software defined networks. In *Proceedings of the 26th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1–9). IEEE.
- [16] Shaghaghi, A., Kaafar, M. A., Buyya, R., & Jha, S. (2020). Software-defined network (SDN) data plane security: Issues, solutions, and future directions. In B. B. Gupta, G. M. Perez, D. P. Agrawal, & D. Gupta (Eds.), *Handbook of Computer*

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

Networks and Cyber Security (pp. 341–387). Springer.

- [17] Stein, G., Chen, B., Wu, A. S., & Hua, K. A. (2024). Decision tree classifier for network intrusion detection with GA-based feature selection. In Proceedings of the 43rd Annual Southeast Regional Conference on ACM-SE 43. ACM Press
- [18] Santos, R., Souza, D., Santo, W., Ribeiro, A., & Moreno, E. (2020). Machine learning algorithms to detect DDoS attacks in SDN. *Concurrency and Computation: Practice and Experience*, 32(16), e5402.
- [19] Tiwari, A., Saraswat, S., Dixit, U., & Pandey, S. (2022, March). Refinements in Zeek intrusion detection system. In Proceedings of the 8th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 974–979). IEEE.
- [20] Tang, F., Fadlullah, Z. M., Mao, B., & Kato, N. (2018). An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach. *IEEE Internet of Things Journal*, 5(6), 5141–5154.
- [21] Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., & Lin, W.-Y. (2019). Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10), 11994–12000.
- [22] Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2019). Deep learning approach for network intrusion detection in software defined networking. *IEEE Access*.
- [23] Wang, M., Cui, Y., Wang, X., Xiao, S., & Jiang, J. (2018). Machine learning for networking: *Workflow, advances and opportunities*. *IEEE Network*, 32(2), 92–99.
- [24] Wang, P., Chao, K.-M., Lin, H.-C., Lin, W.-H., & Lo, C.-C. (2016, November). An efficient flow control approach for SDN-based network threat detection and migration using support vector machine. In Proceedings of the *IEEE 13th International Conference on e-Business Engineering (ICEBE)*. 56–63.
- [25] Wang, R., Jia, Z., & Ju, L. (2015). An entropy-based distributed DDoS detection mechanism in software-defined networking. *In 2015 IEEE Trustcom/BigDataSE/ISPA*. 1: 310–317.
- [26] Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J., Wang, C., & Liu, Y. (2019). A survey

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

- of machine learning techniques applied to software-defined networking (SDN): Research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(1), 393–430.
- [27] Xu, J., Wang, L., & Xu, Z. (2020). An enhanced saturation attack and its mitigation mechanism in software-defined networking. *Computer Networks*, 169, 107092.
- [28] Xu, S., Wang, X., Yang, G., Ren, J., & Wang, S. (2020). Routing optimization for cloud services in SDN-based Internet of Things with TCAM capacity constraint. *Journal of Communications and Networks*, 22(2), 145–158.
- [29] Xie, R., Cao, J., Li, Q., Sun, K., Gu, G., Xu, M., & Yang, Y. (2022). Disrupting the SDN control channel via shared links: Attacks and countermeasures. *IEEE/ACM Transactions on Networking*, 30(5), 2158–2172.
- [30] Xie, J., Yu, F. R., Huang, T., Xie, R., & Liu, J. (2021). A survey on machine learning-based mobile big data analysis: Challenges and applications. *IEEE Access*, 8, 119091–119108.
- [31] Yin, C., Zhu, Y., Fei, J., & He, X. (2021). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5, 21954–21961.
- [32] Yazdinejadna, A., Parizi, R. M., Dehghantanha, A., & Khan, M. S. (2021). A kangaroo-based intrusion detection system on software-defined networks. *Computer Networks*, 184, 107688.
- [35] Zanero, S., & Savaresi, S. M. (2024). Unsupervised learning techniques for an intrusion detection system. In *Proceedings of the 2004 ACM Symposium on Applied Computing - SAC '04*. ACM Press.