Mininet-IDS: A Step Towards Reproducible Research for Machine Learning Based Intrusion Detection Systems

Rana Uzair Ahmed, Muhammad Siraj Rathore
Capital University of Science & Technology, Islamabad, Pakistan
*Correspondence: ranauzairahmed.official@gmail.com

Citation | Ahmed. R. U, Rathore. M. S, "Mininet-IDS: A Step Towards Reproducible Research for Machine Learning Based Intrusion Detection Systems", IJIST, Special Issue. pp 93-103, Oct 2024

Received| Oct 07, 2024 **Revised|** Oct 12, 2024 **Accepted|** Oct 17, 2024 **Published|** Oct 20, 2024.

Software-Defined Networking (SDN) has revolutionized network management by enabling more flexible, programmable, and controlled networks. However, the SDN controller can be a target for attacks that could bring down the entire network. In this context, intrusion detection systems (IDS) are essential for maintaining network security. Modern IDS are often enhanced with machine learning models to detect a range of network attacks. This process typically includes dataset preprocessing, model training, and integration of these models into network emulators like Mininet. However, this workflow can be complex and error-prone. To address these challenges, we present Mininet-IDS, a comprehensive command-line interface (CLI) tool that streamlines the process by offering integrated functionalities for dataset preprocessing, feature selection, model training, and deployment within the Mininet environment. Our tool simplifies the workflow by eliminating compatibility issues and ensuring reproducibility. We evaluate Mininet-IDS using the NSL-KDD dataset, training various machine learning models to detect DDoS attacks. Our results demonstrate the tool's efficiency and accuracy, making it a valuable resource for network security researchers to conduct experiments with minimal machine learning expertise

Keywords: Software Defined Networking (SDN); Intrusion Detection System (IDS); Mininet; Ryu Controller; Machine Learning; Network Security.

Introduction:

In recent years, network architecture has experienced a significant transformation with the emergence of Software-Defined Networking (SDN). SDN represents a paradigm shift in network management by decoupling the control plane from the data plane. This separation allows for centralized network control, enhancing flexibility, programmability, and ease of management. At the core of SDN is the central controller, which serves as the brain of the network, making critical decisions regarding traffic flow and network policies [1][2]. However, this centralized nature, while beneficial, introduces new security vulnerabilities; if the central controller is compromised, the entire network may be at risk [3][4][5]. This vulnerability underscores the urgent need for robust security measures in SDN environments, particularly through the implementation of Intrusion Detection Systems (IDS) [2][5].

Traditional IDS for SDN have primarily relied on signature-based or rule-based approaches, which monitor network traffic for predefined patterns associated with known attacks. While effective against familiar threats, these traditional systems struggle to identify novel or sophisticated attacks that do not conform to existing signatures [2][6][7][8][9]. The dynamic landscape of cyber threats presents a formidable challenge, as new attack vectors continually emerge, exploiting zero-day vulnerabilities and employing advanced evasion techniques. These sophisticated attacks often bypass conventional security measures, highlighting the necessity for more adaptive and intelligent defense mechanisms.

In response to these challenges, researchers are increasingly leveraging Machine Learning (ML) techniques to enhance IDS capabilities. ML-based IDS can learn from data, recognize patterns, and make decisions with minimal human intervention. A variety of ML algorithms, including Support Vector Machines (SVM), Random Forests [3][8], and more recently, deep learning techniques like Long Short-Term Memory (LSTM) [10] networks and Artificial Neural Networks [5], have been applied to intrusion detection. These ML-based approaches show promise in detecting both known and unknown attacks, adapting to new threats, and reducing false positive rates [3].

Despite these advancements, significant challenges remain in the development and deployment of ML-based IDS. The processes involved—data preprocessing, feature selection, model training, and integration with SDN environments—are often complex and difficult to standardize or reproduce. This complexity creates a gap between theoretical research and practical implementation, hindering the widespread adoption of ML-based IDS in real-world SDN settings. To bridge this gap, we introduce Mininet-IDS, a comprehensive tool designed to streamline the entire workflow of IDS development and deployment within SDN environments. Mininet-IDS integrates dataset preprocessing, feature selection, model training, and deployment into a single platform. By providing a unified interface for these tasks and ensuring seamless integration with the Mininet SDN emulator [1][4][11][12] and Ryu controller [4][11][13], our tool facilitates the transition from IDS research to practical application.

We evaluate the effectiveness of Mininet-IDS using the NSL-KDD dataset, a widely recognized benchmark in IDS research [6][8][14][15]. The tool supports dataset preprocessing and feature selection, followed by the training of various ML algorithms. For our evaluation, we focus on training machine learning models to detect DDoS attacks as a specific example of network intrusion. Our results demonstrate the tool's capability to simplify the development process, enhance reproducibility, and maintain high detection accuracy.

The remainder of this paper is structured as follows: Section 2 presents a detailed literature review, examining recent advancements in ML-based IDS for SDN and identifying existing research gaps. Section 3 outlines the architecture of Mininet-IDS and our methodology, detailing data preprocessing, model training, and evaluation procedures. Section 4 presents our experimental results and discusses their implications. Finally, Section 5 concludes the paper and proposes directions for future research. By introducing Mininet-IDS, we aim to contribute to

the standardization and accessibility of ML-based IDS research in SDN environments. Our tool not only facilitates more efficient and reproducible research but also lays the groundwork for the practical implementation of advanced intrusion detection techniques in real-world SDN deployments.

Objectives and Novelty Statement:

This paper presents Mininet-IDS, a tool specifically designed to simplify and streamline the IDS research workflow. By adopting an integrated approach, Mininet-IDS enables all essential steps to be performed within a single platform, enhancing both efficiency and reproducibility. Its innovative design allows for comprehensive management of the entire IDS lifecycle—from data preprocessing to real-time traffic prediction—within one cohesive tool.

Literature Review:

Machine learning algorithms have demonstrated promising results in detecting both known and unknown threats in SDN environments. Recent studies have explored a variety of techniques, from traditional algorithms like Support Vector Machines (SVM) and Random Forests to advanced deep learning approaches. These methods have shown improved accuracy and reduced false positive rates compared to signature-based systems [6]. However, implementing machine learning-based IDS in SDN environments presents unique challenges. A key issue is the need for real-time threat detection, which necessitates efficient processing of large volumes of network data. To address this, researchers have proposed distributed architectures and optimized feature selection methods [5][10][16]. Additionally, some studies have investigated the use of deep reinforcement learning for adaptive intrusion detection in dynamic network environments [2][6][10][17].

Another critical aspect of IDS research is the availability of relevant and up-to-date datasets. Recent efforts have focused on developing more representative datasets [18], including those that capture emerging threats in IoT and cloud environments. The integration of IDS within SDN architecture has also been a major research focus, with studies indicating that SDN's centralized control and programmability can enhance intrusion detection capabilities, allowing for more dynamic and adaptive security measures [2][5]. The development of machine learning-based IDS in SDN environments typically involves several discrete steps, including dataset acquisition and preprocessing, feature selection and engineering, model training and evaluation, and integration with SDN architectures [8][9][16][19][20]. Recent studies have applied these approaches using well-known datasets like NSL-KDD and UNSW-NB15 while employing various machine learning algorithms. For example, one study [16] proposed a hybrid feature selection algorithm combined with Light GBM for intrusion detection, while another [8] utilized Grid Search with SVM. A comparative study [9] evaluated signature-based methods like Snort IDS alongside machine learning algorithms such as Random Forest, J48, Naive Bayes, and SVM. These studies often leverage network simulation tools like Mininet to create virtual SDN environments, with controllers like Open Daylight hosted on cloud platforms. Implementations typically involve generating both normal and attack traffic to assess the effectiveness of the intrusion detection techniques. While these approaches have shown promising results, with some achieving detection accuracies exceeding 99% [8][19], they often require expertise across multiple domains and tools, leading to challenges in reproducibility and standardization among different research groups.

The use of ensemble methods [21] and hybrid approaches has gained traction in recent years, as researchers combine multiple machine learning algorithms to improve detection accuracy and reduce false positives. Additionally, the application of deep learning techniques, such as Convolutional Neural Networks (CNN) [4] and Long Short-Term Memory (LSTM) networks, has proven effective in capturing complex patterns in network traffic [3][6][10]. Researchers have also developed ML-enabled intrusion detection systems specifically for IoT networks, addressing the challenges posed by resource-constrained devices. Utilizing recent

datasets like TON-IoT and UNSW-NB15, these studies illustrate the effectiveness of ML-based IDS in diverse and computationally limited IoT environments, contributing to a more comprehensive approach to network security across various technological domains [22][23].

Despite these advancements, significant gaps remain in the reproducibility and standardization of IDS research within SDN environments. The process of developing, training, and deploying machine learning models for IDS often involves multiple disconnected steps, complicating result reproduction and comparison between different approaches. The traditional fragmented methodology presents several challenges, including reproducibility issues stemming from varying environments and manual processes, as well as a steep learning curve that requires expertise in multiple tools and domains. Furthermore, integrating trained models into SDN controllers for real-time traffic analysis is often not straightforward, creating barriers between theoretical research and practical application. The integration and deployment processes can be time-consuming, and potential compatibility issues between different components may arise. These challenges highlight the urgent need for more integrated, streamlined approaches to IDS development and deployment in SDN environments, which could significantly enhance research efficiency and reproducibility in this field.

Methodology:

To address these challenges, we introduce Mininet-IDS, a comprehensive tool designed to streamline the entire workflow of IDS development and deployment in SDN environments. Mininet-IDS integrates dataset preprocessing, feature selection, model training, and deployment into a single platform, ensuring reproducibility and simplifying the research process. By offering a unified interface for these tasks and seamless integration with the Mininet SDN emulator and Ryu controller, Mininet-IDS effectively bridges the gap between IDS research and practical implementation in SDN environments. Table 1 summarizes the advantages of using Mininet-IDS compared to the traditional manual approaches to employing machine learning for IDS.

Table 1. Comparison between traditional approach and Mininet-IDS

Aspect	Traditional Approach	Mininet-IDS
Workflow	Fragmented: separate steps for preprocessing, model training, and deployment	Integrated: all steps performed within a single tool
Reproducibility	Challenging due to varied environments and manual steps	Ensured through consistent, automated processes
Ease of Use	Requires expertise in multiple tools and environments	User-friendly CLI interface, accessible to researchers with limited ML knowledge
Compatibility	Potential issues when integrating different components	No compatibility issues as all components are integrated
Extensibility	Depends on individual components	Easy to extend with new functionalities through modular design
Time Efficiency	Time-consuming due to manual integration	Faster development and deployment cycle
Learning Curve	Steep, requires understanding of multiple systems	Gentler, focuses on core IDS concepts rather than tool intricacies

Figure 1 presents a comprehensive architectural overview of the Mininet-IDS system, highlighting the key components and their interactions as described in the codebase. This architecture enables the integration of various functionalities, including dataset management, machine learning, network simulation, and intrusion detection, into a unified and modular system. At the core of the system is the Mininet-IDS class, which serves as the central interface

and controller. This class orchestrates interactions among all components, ensuring coordinated and efficient operation throughout the system and managing the flow of data and commands between different modules.

The dataset management component, directly supervised by the Mininet-IDS class, handles all operations related to datasets, including importing, listing, selecting, and manipulating them. The functionality of this component is encapsulated in the Dataset class. Data managed by this component is subsequently utilized by the machine learning module for model training and evaluation. Machine learning activities are overseen by the Machine Learning class, which receives data from the dataset management component and is responsible for training and evaluating models based on this data. Once trained, the models are stored and managed by the Model class, which is part of the model management component. This component facilitates the import, export, and listing of trained models, thereby maintaining a repository for future use.

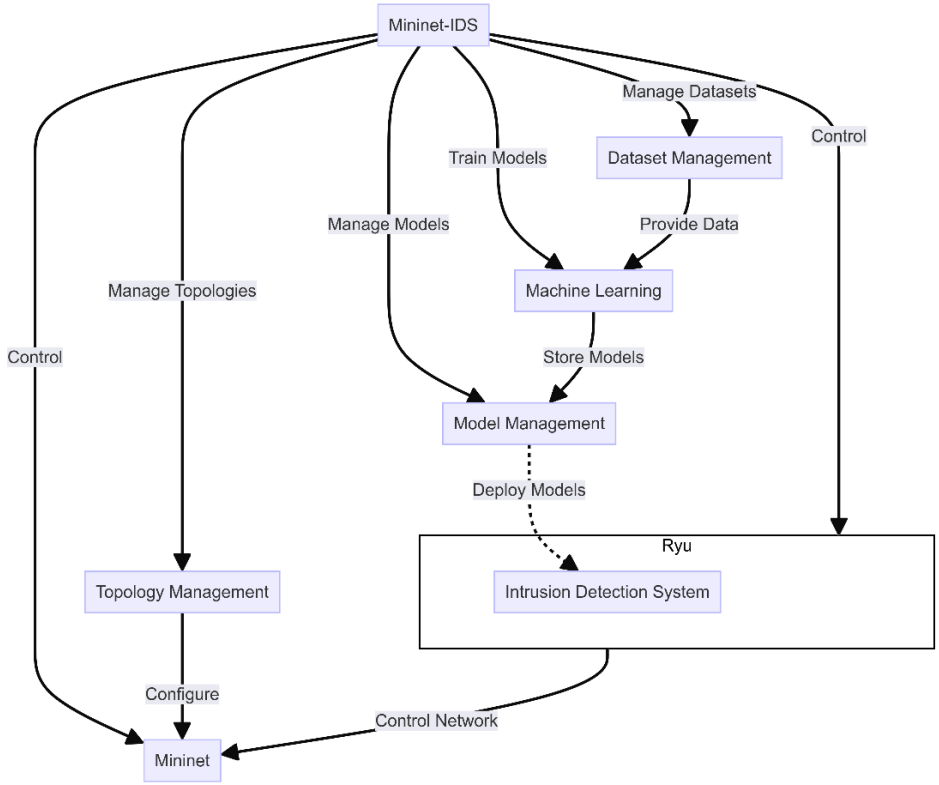

Figure 1. Architecture of Mininet-IDS

Topology management, represented by the Topology class, oversees the creation, listing, and removal of network topologies. This component interacts with Mininet, a network simulation environment, through subprocess calls, enabling the configuration and control of network simulations and tests. This functionality facilitates the evaluation of network performance and behavior under various conditions.

The intrusion detection system (IDS) is implemented within the Ryu class, operating on the Ryu SDN controller. It utilizes trained models from the model management component to detect and respond to network intrusions. The Ryu class is responsible for initiating and configuring the IDS, ensuring seamless integration with the SDN controller for real-time intrusion detection.

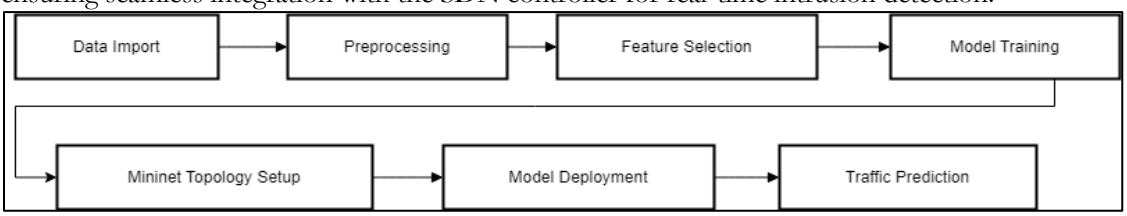

Figure 2. Flow of Methodology

Figure 2 illustrates the workflow of Mininet-IDS. The process begins with data import, loading datasets such as NSL-KDD into the system. Next, preprocessing occurs, which includes feature selection and data normalization. The preprocessed data is then utilized to train various machine learning models. Simultaneously, a Mininet topology is established to emulate the SDN environment. Once trained, the models are deployed within this environment using the Ryu controller. Finally, real-time network traffic is collected and processed through the deployed model for prediction, classifying it as either normal or potentially malicious.

Investigation Site:

Our research environment features a Mininet-based Software-Defined Networking (SDN) testbed controlled by Ryu, both of which are designed to operate seamlessly within a Linux environment. Mininet serves as a network emulator that enables researchers to create virtual networks on a single machine, facilitating the simulation of complex topologies. Its lightweight architecture is particularly well-suited for developing and testing SDN applications, as it can quickly configure various network setups. Ryu, an open-source SDN controller, works in tandem with Mininet by managing network resources and implementing different networking protocols.

The network topology is organized in a linear configuration with six switches, labeled s1 through s6. Each switch connects to three hosts, resulting in a total of 18 hosts, named h1 through h18. These switches are governed by a central Ryu controller, which oversees the entire network. Ryu's modular design allows for easy integration of various applications, enabling researchers to implement custom functionalities such as traffic management, security, and monitoring. By leveraging Mininet and Ryu within a Linux environment, this testbed provides an efficient platform for exploring intrusion detection techniques using the NSL-KDD dataset, thereby advancing SDN security research.

The NSL-KDD dataset was selected for evaluation due to its widespread adoption in intrusion detection system (IDS) research, serving as a standard benchmark for comparison. This dataset improves upon the original KDD Cup 1999 dataset by addressing issues related to redundancy and the lack of diversity in attack types. The NSL-KDD dataset encompasses a wide range of attack categories, making it a valuable resource for training and evaluating IDS models.

Data Acquisition and Preprocessing:

Mininet-IDS facilitates the import of various IDS datasets. For this study, we selected features from the NSL-KDD dataset that correspond to Mininet's environment statistics. Categorical features were converted to numerical values, and a new feature was created by combining existing ones. Subsequently, the features were renamed to align with the expectations of the prediction application.

The data preprocessing phase comprised four key steps: feature selection, which involved identifying relevant features in the NSL-KDD dataset that matched Mininet's network statistics; categorical mapping, which converted categorical features into numerical values; feature engineering, aimed at creating new features to enhance model performance; and feature renaming, to ensure compatibility with the prediction application. These steps effectively prepared the dataset for use within the Mininet-IDS framework, ensuring alignment with both the training algorithms and the deployment environment.

Model Training:

The model training process in Mininet-IDS consisted of three key steps. First, we selected standard machine learning algorithms supported by the scikit-learn library, including Naive Bayes (NB), Decision Trees (DT), Random Forests (RF), Logistic Regression (LR), and K-Nearest Neighbors (KNN). Next, we evaluated these models using confusion matrices and accuracy metrics to gauge their performance. Finally, we saved the trained models for deployment in the Mininet environment, ensuring seamless integration into our SDN-based IDS system for real-time traffic classification.

Deployment and Prediction:

The trained models were deployed using Mininet-IDS. We began by starting a Mininet topology and launching the Ryu prediction application to collect network traffic data. The traffic was then classified in real-time as either normal or indicative of an attack. The deployment and prediction process within Mininet-IDS involved several critical stages, showcasing the tool's practical application in a Software-Defined Network environment.

After initializing the Mininet topology with predefined configurations, we established the network structure for our experiments. With the topology in place, we proceeded to deploy our trained machine learning models within the Ryu controller's prediction application. To evaluate the system's performance, we generated both normal and attack traffic within the Mininet environment. The core functionality of our system lies in its ability to perform real-time traffic prediction. As network traffic flowed through the system, our deployed models continuously collected and classified the data. These results underscore the Mininet-IDS tool's capability to seamlessly integrate trained models into an SDN environment and execute real-time intrusion detection, representing a significant advancement in network security research and implementation.

Results and Discussion:

Results:

The evaluation of the Mininet-IDS tool was conducted using the NSL-KDD dataset. We assessed the performance of several machine learning models, including Logistic Regression, K-Nearest Neighbors, Naive Bayes, Decision Tree, and Random Forest. The key metrics for evaluation included Accuracy, Precision, Recall, and F1 Score. Below are the results for each model.

Confusion Matrix:

Table 2 is utilized to evaluate the performance of a classification algorithm. It includes four key values:

Table 2. Sample confusion matrix

	Predicted Negative	**Predicted Positive**
Actual Negative	TN	FP
Actual Positive	FN	TP

- **True Positive (TP):** The number of correctly predicted positive instances.
- **True Negative (TN):** The number of correctly predicted negative instances.
- **False Positive (FP):** The number of incorrectly predicted positive instances (Type I error).
- **False Negative (FN):** The number of incorrectly predicted negative instances (Type II error).

This matrix provides insights into the types of errors made by the classification model and the balance between precision and recall.

Table 3. Confusion Matrices for LR, KNN, NB, DT & RF

Model	**TN**	**FP**	**FN**	**TP**
Logistic Regression	14323	1072	13629	680
K-Nearest Neighbors	14635	760	180	14129
Naive Bayes	15128	267	14008	301
Decision Tree	14619	776	143	14166
Random Forest	14634	761	150	14159

The confusion matrices presented in Table 3 offer valuable insights into each model's ability to accurately identify normal and attack instances. A more detailed comparison of the models' performance is provided in Table 4, which summarizes key metrics such as Accuracy, Precision, Recall, and F1 Score for each model.

Table 4. Model Performance Metrics

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.51	0.39	0.05	0.08
K-Nearest Neighbors	0.97	0.95	0.99	0.97
Naive Bayes	0.52	0.53	0.02	0.04
Decision Tree	0.97	0.95	0.99	0.97
Random Forest	0.97	0.95	0.99	0.97

To provide clarity on the evaluation criteria, Table 5 outlines the definitions of each performance metric, along with their corresponding formulas:

Table 5. Performance Metrics Definitions

Metric	Description	Formula
Accuracy	Overall correctness of the model	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	Proportion of correct positive predictions among all positive predictions	$\frac{TP}{TP + FP}$
Recall	Proportion of actual positive cases that were correctly identified	$\frac{TP}{TP + FN}$
F1 Score	Weighted average of precision and recall	$2 \times \frac{Precision \times Recall}{Precision + Recall}$

Discussion:

The results of this study demonstrate the effectiveness and efficiency of Mininet-IDS in streamlining the development and deployment process for Intrusion Detection Systems in Software-Defined Networks. By integrating dataset preprocessing, feature selection, model training, and deployment into a single, user-friendly tool, Mininet-IDS addresses several key challenges faced by researchers in network security. The performance metrics of the machine learning models, illustrated in Figure 3, provide valuable insights into their effectiveness for intrusion detection tasks. The K-Nearest Neighbors, Decision Tree, and Random Forest models exhibited exceptional performance, each achieving an accuracy of 0.97, along with high precision, recall, and F1 scores. This high performance can be attributed to these models' ability to capture complex patterns and non-linear relationships within the NSL-KDD dataset.

In contrast, the Logistic Regression and Naive Bayes models showed lower performance, highlighting the importance of selecting appropriate algorithms that align with the specific characteristics of network traffic data. The complex, non-linear nature of intrusion patterns may not be adequately captured by simpler linear models or those with strong independence assumptions. The integration of these diverse machine learning models within a CLI-based framework marks a significant advancement in IDS research. By simplifying the traditionally complex processes of model development and deployment, Mininet-IDS makes sophisticated intrusion detection techniques more accessible to researchers, including those with limited machine learning expertise. This increased accessibility has the potential to accelerate innovation in the field of network security.

Furthermore, the tool's ability to seamlessly deploy trained models within a Mininet-Ryu environment for real-time traffic classification demonstrates its practical applicability. This feature bridges the gap between theoretical model development and practical implementation in SDN environments, representing a crucial step in translating research into real-world security solutions.

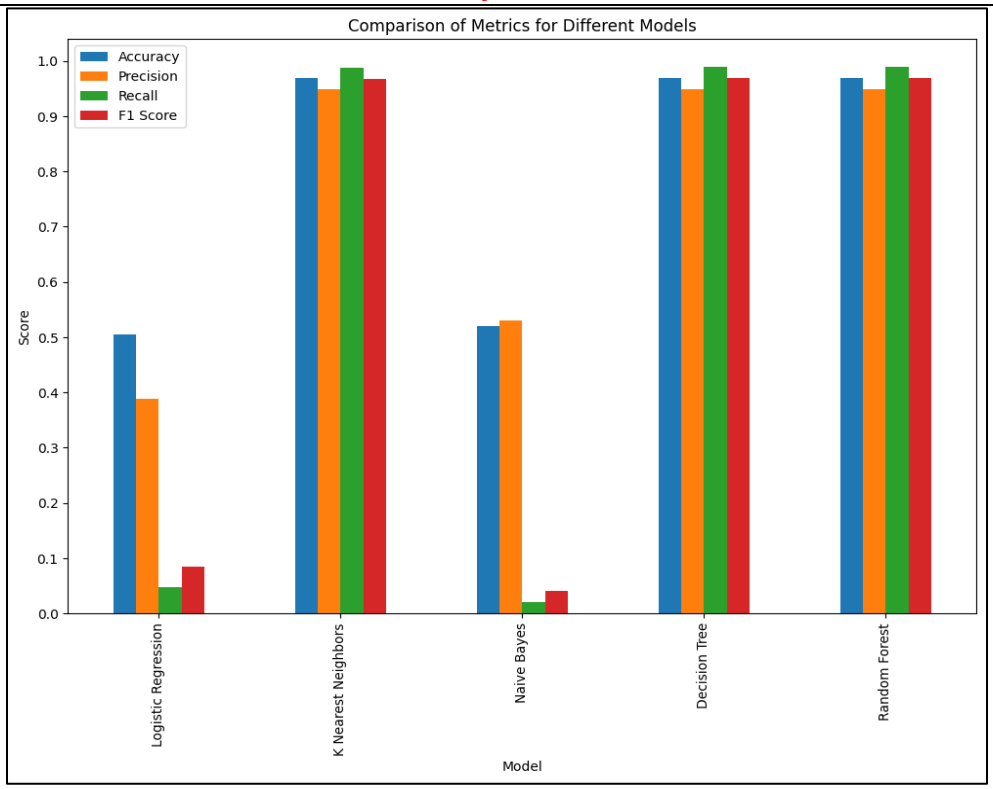

Figure 3. Comparison of Model Performance Metrics

It's important to highlight that Mininet-IDS is an open-source project available on GitHub [24]. This open-source nature encourages collaboration, facilitates community-driven improvements, and promotes transparency in research methodologies. However, Mininet-IDS does have some limitations. First, its current implementation focuses on a specific set of machine learning algorithms and may not incorporate more advanced techniques, such as deep learning models. Second, while the NSL-KDD dataset used in this study is widely recognized, it may not fully capture the complexity and diversity of modern network attacks. Future work should involve testing the tool with more recent and varied datasets. Lastly, the tool's performance in large-scale, real-world networks requires further investigation to ensure its scalability and effectiveness under varying network conditions.

Conclusion:

Mininet-IDS represents a significant advancement in intrusion detection research for Software-Defined Networks (SDNs). By providing a comprehensive, integrated platform for dataset management, model training, and deployment, it effectively addresses many challenges traditionally associated with Intrusion Detection System (IDS) development and implementation. The tool streamlines the entire workflow—from data preprocessing to real-time traffic classification—enhancing efficiency and promoting reproducibility in research. This is particularly crucial in the rapidly evolving field of network security, where replicable results are essential for building upon existing knowledge.

The performance evaluation of various machine learning models within the Mininet-IDS framework highlights its versatility, with ensemble and non-linear models like Random Forest, Decision Tree, and K-Nearest Neighbors demonstrating superior performance. This underscores the complexity of intrusion detection tasks and the necessity for sophisticated analytical approaches. Additionally, the open-source nature of Mininet-IDS, with its code available on GitHub [24], fosters collaboration among researchers and practitioners. This accessibility supports the continuous improvement of the tool and promotes transparency and knowledge sharing within the network security community.

Looking ahead, Mininet-IDS provides a solid foundation for future research and development in SDN security. Its modular design allows for easy extensions, enabling researchers to incorporate new algorithms, features, or analytical techniques as the field evolves. However, it is essential to acknowledge the current implementation's limitations. Future work should focus on expanding the range of supported machine learning algorithms, testing with more diverse and recent datasets, and evaluating performance in large-scale, real-world network environments. In conclusion, Mininet-IDS not only simplifies IDS research but also paves the way for more advanced and accessible network security solutions in the era of SDNs. By lowering the barrier to entry for IDS development and deployment, it has the potential to encourage innovation and collaboration within the network security community, ultimately contributing to effective defenses against evolving cyber threats.

Author's Contribution: All authors contributed equally to this work.

Conflict of Interest: There are no conflicts of interest to declare.

References:

[1] Nisha Kumari and Kapil Kathuria. "Overview of SDN Building Foundations and Applications." "Journal of Research in Science and Engineering" 2024, 6(7), 43–53.

[2] Gulshan Kumar and Hamed Alqahtani. "Machine Learning Techniques for Intrusion Detection Systems in SDN-Recent Advances, Challenges and Future Directions" "Computer Modeling in Engineering & Sciences" 2023, 134(1), 89-119.

[3] Nisha Ahuja, Gaurav Singal, Debajyoti Mukhopadhyay, and Neeraj Kumar. "Automated DDOS attack detection in software defined networking." "Journal of Network and Computer Applications" 2021, 187, 103108.

[4] Yousif Al-Dunainawi, Bilal R. Al-Kaseem, and Hamed S. Al-Raweshidy. "Optimized Artificial Intelligence Model for DDoS Detection in SDN Environment." "IEEE Access" 2023, 11, 106733-106748.

[5] Sabila Nawshin, Salekul Islam, and Swakkhar Shatabda. "PCA-ANN: Feature Selection Based Hybrid Intrusion Detection System in Software Defined Network." "Journal of Intelligent & Fuzzy Systems" 2024, 1-18.

[6] Tariq Emad Ali, Yung-Wey Chong, and Selvakumar Manickam. "Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review" "Applied Sciences" 2023, 13(5), 3183.

[7] Meryem Chouikik, Mariyam Ouaissa, Mariya Ouaissa, and Zakaria Boulouard. "Detection and Mitigation of DDoS Attacks in SDN Based Intrusion Detection System" "Bulletin of Electrical Engineering and Informatics" 2024, 13(4), 2750-2757.

[8] Omar Jamal and Wesam S. Bhaya. "Intrusion Detection System for Cloud Based Software-Defined Networks." "Journal of Physics Conference Series" 2021, 1804(1), 012007.

[9] Oluwapelumi Fakolujo and Amna Qureshi. "Analysis of Detection Systems in a Software-Defined Network" "Intelligent Computing" 2023, Lecture Notes in Networks and Systems, 739, 1342–1363.

[10] Elsayed Mahmoud Said, Le-Khac Nhien-An, Azer Marianne A. and Jurcut Anca D. "A Flow Based Anomaly Detection Approach with Feature Selection Method Against DDoS Attacks in SDNs" "IEEE Transactions on Cognitive Communications and Networking" 2022, Volume 8, Issue 4, pp. 1862-1880.

[11] Neelam Gupta, Mashael Maashi, Sarvesh Tanwar, and Sumit Badotra. "A Comparative Study of Software Defined Networking Controllers Using Mininet." "Electronics" 2022, 11(17): 2715.

[12] https://mininet.org/ Mininet

[13] https://ryu-sdn.org/ Ryu

[14] Mossa Ghurab, Ghaleb Gaphari, Faisal Alshami, and Reem Alshamy. "A Detailed Analysis of Benchmark Datasets for Network Intrusion Detection System" 2021. "Asian Journal of Research in Computer Science" 2021, Volume 7, Issue 4, pp. 14-33.

[15] https://www.unb.ca/cic/datasets/nsl.html NSL-KDD Dataset

[16] G. Logeswari, S. Bose, and T. Anitha. "An Intrusion Detection System for SDN Using Machine Learning" "Intelligent Automation & Soft Computing" 2023, 35(1), 867-880.

[17] Nguyen Thanh Thi and Reddi Vijay. "Deep Reinforcement Learning for Cyber Security" "IEEE Transactions on Neural Networks and Learning Systems" 2023. Volume 34, Issue 8, Pages 3779-3795.

[18] Siyyal Shafqat Ali, Khuawar Faheem Yar, Saba Erum, Memon Abdul Latif, Shaikh Muhammad Raza. "Analyzing ML-Based IDS over Real-Traffic" "International Journal of Innovations in Science & Technology" 2022. Volume 4, Issue 3, Pages 621–640.

[20] Mohsin Mayadah A. and Hamad Ali H. "Performance Evaluation of SDN DDoS Attack Detection and Mitigation Based Random Forest and K-Nearest Neighbors Machine Learning Algorithms" "Revue d'Intelligence Artificielle" 2022. Volume 36, Issue 2, Pages 233-240.

[21] Alashhab Abdussalam Ahmed, Zahid Mohd Soperi, Isyaku Babangida, Elnour Asma Abbas, Nagmeldin Wamda, Abdelmaboud Abdelzahir, Abdullah Talal Ali Ahmed, and Maiwada Umar. "Enhancing DDoS Attack Detection and Mitigation in SDN Using an Ensemble Online Machine Learning Model" "IEEE Access" 2024, Volume 12, Pages 51630-51649.

[22] Zewdie Temechu & Girma Anteneh. "IoT security and the role of AI/ML to combat emerging Cyber threats in Cloud Computing Environment" "Information Systems Journal" 2020, Volume 21, Issue 4, Pages 253-263.

[23] Al-Ambusaidi Mohammed & Yinjun Zhang. "ML-IDS: an efficient ML-enabled intrusion detection system for securing IoT networks and applications" "Soft Computing" 2023, Volume 28, Issue 1, Pages 1765-1784.

[24] https://github.com/ranauzairahmed/MininetIDS Mininet-IDS