[1]M Sri Lakshmi
[2]G. Rajavikram
[3*] V Dattatreya
[4]B. Swarna Jyothi,
[5] Shruti Patil
[6]M Bhavsingh

Evaluating the Isolation Forest Method for Anomaly Detection in Software-Defined Networking Security

JES Journal of Electrical Systems

Abstract: **-** The research addresses the critical anomaly detection problem in Software-Defined Networking (SDN), a domain where network integrity and security are paramount. Employing the Isolation Forest algorithm, a machine learning model renowned for its efficacy in identifying outliers, the study systematically generates synthetic network traffic data to train and test the model's detection capabilities. The methodology encompasses simulating a range of contamination rates to reflect varying degrees of anomalous activities within the network. Key findings indicate that while the model exhibits potential in anomaly detection, as reflected by the progressive increase in triggered alerts and policy changes, its performance metrics, such as precision, recall, F1-score, and AUC, reveal limitations in its current application. The research contributes to the field by providing a detailed analysis of the Isolation Forest algorithm's performance in an SDN context and laying the groundwork for future enhancements in machine learning-based security measures within these networks.

Keywords: Anomaly Detection, Software-Defined Networking, Isolation Forest, Synthetic Data Simulation, Network Security.

I. INTRODUCTION

The advent of Software-Defined Networking (SDN) marks a paradigm shift in network management, allowing for more agile and centralized control mechanisms [1]. This innovation is critical as networks expand in size and complexity, particularly with the advent of cloud computing and the Internet of Things (IoT). SDN's ability to dynamically programmatically manage, configure, and optimize network resources underscores its significance in modern digital infrastructure [2]. Within this dynamic environment, the role of anomaly detection becomes increasingly vital. Anomalies in network traffic can indicate various issues, from system faults to cybersecurity threats [3].

The capacity to swiftly identify and address these irregularities is central to maintaining network integrity and security. As such, anomaly detection is not just a feature of SDN—it is essential for its sustained operation and reliability [4]. The complexity of network interactions in SDN environments, coupled with the sheer volume of data, necessitates the integration of machine learning [5]. Traditional statistical methods are often inadequate in parsing the nuanced patterns that may signal a network anomaly. With their ability to learn from data, machine learning algorithms offer a promising solution for enhancing anomaly detection capabilities in SDN [6, 7]. This paper aims to evaluate the applicability and efficiency of the Isolation Forest machine learning algorithm in detecting network anomalies within an SDN context. By simulating diverse network behaviors and potential anomalies, the study seeks to establish a benchmark for the algorithm's performance and identify potential improvement areas.

3* (*Corresponding author*): Professor, CSE Department, CVR College of Engineering, Telangana,India.
Email ID: dattatreya.valiveti@gmail.com
[1]Associate Professor, Department of Computer Science and Engineering , G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India. Email Id: srilakshmicse@gpcet.ac.in
[2]Professor , Department Of Computer Science And Engineering, Vignan Institute Of Technology And Science, Deshmukhi, Telangana ,India. Email ID: grajavikram@gmail.com
[4] Assistant professor ,CSE(DS) ,RGMCET Nandyal , Andhra Pradesh, India , Email ID: badimela1508@gmail.com
[5] Assistant Professor , Department of Information Technology,MLR Institute of Technology, Hyderabad ,India ,
Email ID: shrutisib@gmail.com
[6]Associate Professor, Department of Computer Science and Engineering, Ashoka Womens Engineering College, Kurnool, Andhra Pradesh, India. Email ID: bhavsinghit@gmail.com

This paper aims to evaluate the applicability and efficiency of the Isolation Forest machine learning algorithm in detecting network anomalies within an SDN context. By simulating diverse network behaviors and potential anomalies, the study seeks to establish a benchmark for the algorithm's performance and identify potential improvement areas.

Key Research Highlights

- **Centralized Network Management**: Emphasizes how SDN's centralized control facilitates rapid response and adaptation to network changes, a capability that is leveraged by the integration of machine learning for anomaly detection.
- **Automated Anomaly Detection**: Highlights the transition from manual to automated anomaly detection mechanisms, showcasing the potential for machine learning to operate in real-time, thereby enhancing network security.
- **Machine Learning Efficacy**: Investigates the effectiveness of the Isolation Forest algorithm in identifying anomalies, contributing to the body of knowledge on machine learning applications in SDN.
- **Benchmarking Performance**: Provides a systematic algorithm evaluation against simulated network scenarios, offering insights into its operational strengths and weaknesses.
- **Future-Ready SDN**: Aligns the research with the forward trajectory of network technology, identifying how SDN can evolve to meet future demands, particularly regarding security and performance.

The findings and discussions presented herein aim to contribute to the strategic development of SDN capabilities, offering a roadmap for integrating advanced machine learning techniques to bolster network anomaly detection and, by extension, network security. Concluding the introductory discourse, the paper proceeds with a structured exposition. Section 2 delves into the existing literature, situating our work within the broader context of network security and machine learning applications. Section 3 articulates the methodology, detailing the synthetic data generation process, the Isolation Forest algorithm's implementation, and the simulation-based evaluation setup. Section 4 presents a critical analysis of the results, examining the algorithm's performance metrics and their implications for SDN anomaly detection. The practical implications of our findings are followed by discussions in the same section, reflecting on the integration of machine learning in SDN environments. Finally, Section 6 concludes the research and proposes future avenues for investigation, ensuring the study's relevance and applicability to ongoing advancements in SDN security. This organization guides the reader through a coherent narrative, from theoretical underpinnings to practical applications, culminating in a forward-looking perspective on the field.

II. 2. BACKGROUND AND RELATED WORK

The literature background section provides a comprehensive examination of the existing research on the application of machine learning in network security, delving into diverse methodologies and their implementations in various contexts. This exploration includes studies on intrusion detection in network traffic, security in Industrial IoT networks, communication security in drone swarms, network anomaly detection, and mobile malware detection. Each area reflects network security challenges' growing complexity and urgency in an increasingly connected world. The review highlights the advancements made and underscores the pivotal role of machine learning in enhancing and innovating network security solutions.

2.1. SDN Architectures

The concept of Software-Defined Networking (SDN) has become a cornerstone in the evolution of network architectures. It revolutionizes network management by decoupling the control plane from the data forwarding plane, thereby introducing a high degree of programmability in network configurations. The exploration of SDN architectures is rich and varied, covering centralized, distributed, and hybrid models incorporating Network Functions Virtualization (NFV) [8].

- **Centralized and Distributed SDN Architectures:** Research has thoroughly investigated load balancing within SDN architectures, particularly in centralized and distributed frameworks. A pivotal study delineates a classification of load-balancing strategies employed in these differing architectures, examining their operational mechanisms and efficiencies [9].

- **Flat Distributed SDN Architectures:** Routing and scalability within flat distributed SDN architectures have also been scrutinized. Considering the innate centralized nature of traditional SDN, one study probes the merits of distributing control across the network, hence addressing scalability issues. This analysis underscores the balance required between control plane routing overhead and inter-controller communications when decentralization is introduced [10].
- **Integrated NFV/SDN Architectures:** The synergy of NFV and SDN is another area that has attracted significant attention. NFV seeks to virtualize network functions onto generic hardware, while SDN separates control and data planes for enhanced network management. Comprehensive reviews have synthesized the current designs and pinpointed opportunities for advancement within integrated NFV/SDN architectures, propelling the conversation on architectural innovation and optimization [11].
- **SDN Architectures with Multiple Controllers:** To enhance scalability, reliability, and availability, the concept of multi-controller architectures in SDN has been examined. Overviews of this subject matter elucidate the diversity of multi-controller designs and dissect the trade-offs associated with various implementations, both theoretical and practical [12]. SDN architectures present a spectrum of designs, each with its unique approach to addressing the challenges of network programmability, scalability, and robustness. These architectures are fundamental to the ongoing discourse on network evolution and the pursuit of optimal operational efficacy.

2.2. Anomaly Detection

Anomaly detection is a crucial challenge in numerous fields, attracting diverse research methodologies tailored to specific application domains. These methodologies extend from specialized domain-centric techniques to more universally applicable strategies. Commonly adopted approaches in anomaly detection include the application of knowledge distillation, advanced deep learning models like convolutional neural networks (CNN), recurrent autoencoders, and innovative deep transformer networks. The utility of these techniques spans various sectors, notably including the analysis of IoT and multivariate time series data. Their efficacy is often established through rigorous experimentation and comprehensive benchmarking processes [13].

An intriguing development in this realm is the application of knowledge distillation for unsupervised anomaly detection. A notable study in this area introduced a distinctive teacher-student framework utilizing a reverse distillation paradigm. This method enhances the diversity in anomalous data representation, significantly advancing anomaly detection methodologies [14].

An integrated approach combining CNN and recurrent autoencoder models has been proposed in the IoT time series domain. This approach leverages a dual-stage sliding window technique in the data pre-processing phase, enhancing the model's ability to capture relevant features. The model's performance, as evidenced by empirical results, surpasses traditional models across various classification metrics, offering an effective solution for anomaly detection in IoT time series [15].

A novel deep transformer network-based model, TranAD, has also been designed for multivariate time-series data. TranAD employs attention-based sequence encoders that rapidly infer anomalies using comprehensive temporal trends. Incorporating model-agnostic meta-learning (MAML) allows for efficient training with limited datasets. Empirical studies demonstrate TranAD's superiority over existing baseline methods, both in detection accuracy and diagnostic capabilities, underpinning its effectiveness in handling complex data sets [16].

The development of an anomaly detection benchmark, ADBench, has provided a platform for evaluating the performance of various algorithms across 57 benchmark datasets. This benchmarking exercise encompasses different supervision levels, anomaly types, and noisy or corrupted data scenarios. ADBench offers invaluable insights into the influence of supervision and anomaly characteristics on detection algorithms, guiding future research in algorithm selection and design [17]. The landscape of anomaly detection is marked by diverse and evolving techniques, ranging from knowledge distillation to sophisticated deep-learning models. These methods have proven their mettle across various domains, substantiated by extensive experimental validation and benchmarking. Such advancements demonstrate the effectiveness of these methods and pave the way for future innovations in anomaly detection across different application domains [18].

2.3 Machine Learning in Network Security

The integration of machine learning in network security has evolved significantly, offering innovative solutions to combat various cyber threats. Here is an overview of some notable implementations:

- **Intrusion Detection in Network Traffic:** In network security, a groundbreaking approach was introduced with the development of a genetic machine learning ensemble model designed explicitly for intrusion detection in network traffic. This model, tailored to IoT networks, was crafted to excel beyond traditional Intrusion Detection Systems (IDS) capabilities, presenting a more dynamic and efficient method for identifying and mitigating intrusions [19].
- **Security Attacks in Industrial IoT Networks:** Addressing security concerns in Industrial IoT networks, a pioneering study combined the strengths of Blockchain algorithms with machine learning techniques. This composite solution was engineered to identify potential threats and initiate secure information transfer protocols while adapting to the unique computational demands of industrial IoT environments [20].
- **Securing Communications in a Swarm of Drones:** Another innovative application involved machine learning in conjunction with Software Defined Network (SDN) technologies to bolster the security of communication networks within drone swarms. The implemented solution employed a Random Forest Classifier-based machine learning model to detect prevalent network attacks, including denial of service, port scanning, and brute force attacks, showcasing the versatility of machine learning in diverse network scenarios [21].
- **Network Anomaly Detection:** Focusing on network anomaly detection, a comprehensive study employed multilayer feature selection techniques alongside machine learning algorithms. The goal was to refine intrusion detection methods, utilizing the Intrusion Detection Evaluation Dataset (CIC-IDS 2017) as a benchmark. This research provided an in-depth analysis of intrusion detection techniques, enhancing the understanding of anomaly detection in network security [22].
- **Mobile Malware Detection:** A systematic overview of machine learning methods for mobile malware detection was conducted in the domain of mobile security. This extensive study analyzed 154 selected articles, critically evaluating their methodologies and findings. The aim was to equip researchers with profound insights into the field and identify potential directions for future research in mobile malware detection [23]. These diverse implementations underscore the breadth of machine learning applications in network security. From enhancing intrusion detection systems to safeguarding communication networks and mobile devices, machine learning has become integral to developing advanced, resilient cybersecurity solutions.

2.4. Research Gaps Identified from the Literature Review

Despite the advancements in machine learning applications within network security, as outlined in the literature review, several research gaps remain evident:

1. **Adaptability to Evolving Threats**: Current models often lack the dynamic adaptability to keep pace with rapidly evolving cyber threats, especially in IoT and Industrial IoT networks.
2. **Scalability and Efficiency in Diverse Environments**: Many existing solutions do not adequately address scalability and efficiency, particularly in complex network environments like drone swarms and large-scale IoT networks.
3. **Integration of Advanced Machine Learning Techniques**: While some studies have implemented sophisticated machine learning models, there is a gap in the comprehensive integration of newer techniques, such as deep learning and ensemble methods, across various network security applications.
4. **Real-World Data Validation**: Most research relies on simulated or benchmark datasets. It is necessary to validate and test these machine-learning models using real-world data to enhance their practical applicability and reliability.
5. **Holistic Security Solutions**: Many studies focus on specific aspects of network security, such as intrusion detection or malware prevention, without considering a holistic approach encompassing all network security facets.

2.5. Proposed System

To address these gaps, the proposed system integrates the Isolation Forest algorithm within a Software-Defined Networking (SDN) environment for enhanced anomaly detection. The system utilizes a sophisticated approach to generate synthetic network traffic data, simulating various network behaviors and potential anomalies. This setup allows for a controlled yet realistic evaluation of the model's performance.

2.5.1. Key aspects of the proposed system include:

1. **Dynamic Anomaly Detection**: Leveraging the Isolation Forest algorithm, the system can dynamically identify anomalies in network traffic, catering to the changing patterns of cyber threats.
2. **Scalable Architecture**: The integration with SDN ensures scalability, allowing the system to adapt to different network sizes and complexities.
3. **Real-time Network Monitoring**: The system provides real-time monitoring and rapid response capabilities by continuously analyzing network traffic.
4. **Feedback Loop for Continuous Improvement**: Incorporating a feedback mechanism, the model is retrained and updated based on its performance and new data, ensuring its relevance over time.
5. **Comprehensive Security Approach**: The system is designed to detect anomalies and influence SDN policy changes, offering a more holistic approach to network security.

This proposed system aims to bridge the identified gaps in current research, offering an advanced, adaptable, and comprehensive solution for network security in the era of complex and evolving cyber threats.The literature background underscores the significant strides in integrating machine learning into network security. While these studies have laid a solid foundation, they also reveal critical gaps, such as the need for adaptable, scalable, and comprehensive security solutions to keep pace with the evolving landscape of cyber threats. The diversity of applications, from IoT to industrial systems and mobile networks, highlights the versatility and necessity of machine learning in this field. This review sets the stage for the proposed system, which seeks to address these gaps by implementing the Isolation Forest algorithm within an SDN framework, offering a novel, dynamic approach to anomaly detection and network security management. The insights gleaned from this review inform the development of the proposed system and pave the way for future research directions in machine learning-driven network security.

III. PROPOSED SYSTEM ARCHITECTURE

This section delineates the comprehensive mathematical models and algorithms employed in integrating anomaly detection within a Software-Defined Networking (SDN) framework using machine learning.

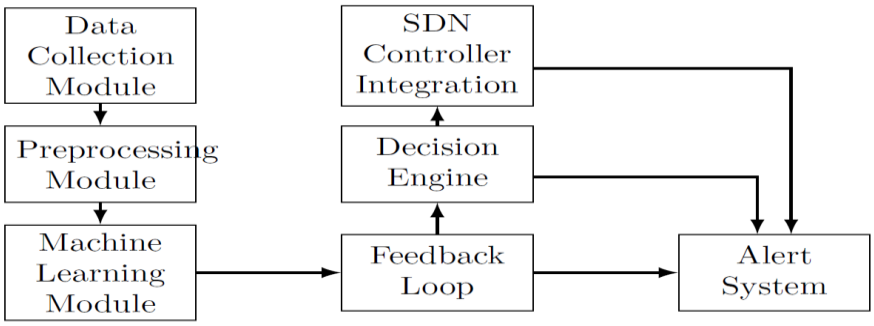

Figure 1: Proposed System Architecture

3.1 Data Collection / Generation

This module is responsible for gathering and generating the required data for the anomaly detection process It involves the creation of synthetic datasets that mimic normal and anomalous network behaviors. We synthetically create the data as we do not have an existing dataset for this research. The creation of synthetic data is pivotal to emulating network behaviors and anomalies. The generation process follows a probabilistic model defined by:

$$X_{\text{normal}} \sim \mathcal{N}(\mu_{\text{normal}}, \Sigma_{\text{normal}})$$
$$X_{\text{anomalous}} \sim \mathcal{N}(\mu_{\text{anomalous}}, \Sigma_{\text{anomalous}})$$

Where X_{normal} and $X_{\text{anomalous}}$ represent the feature vectors for normal and anomalous data instances, respectively. The parameters μ_{normal} and Σ_{normal} are the mean and covariance matrix of the multivariate normal distribution for regular network traffic, while $\mu_{\text{anomalous}}$ and $\Sigma_{\text{anomalous}}$ correspond to the anomalous traffic. The contamination rate θ determines the proportion of anomalies introduced into the dataset.

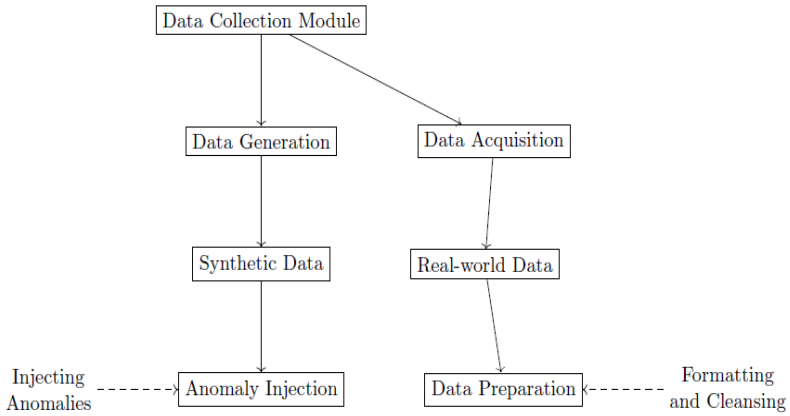

Figure 2: Data Collection / Generation Model

It involves the creation of synthetic datasets that mimic normal and anomalous network behaviors. We synthetically create the data as we do not have an existing dataset for this research. The creation of synthetic data is pivotal to emulating network behaviors and anomalies. The generation process follows a probabilistic model defined by:

$$\mathrm{X}_{\text{normal}} \sim \mathcal{N}(\mu_{\text{normal}}, \Sigma_{\text{normal}})$$
$$\mathrm{X}_{\text{anomalous}} \sim \mathcal{N}(\mu_{\text{anomalous}}, \Sigma_{\text{anomalous}})$$

Where X_{normal} and $X_{\text{anomalous}}$ represent the feature vectors for normal and anomalous data instances, respectively. The parameters μ_{normal} and Σ_{normal} are the mean and covariance matrix of the multivariate normal distribution for regular network traffic, while $\mu_{\text{anomalous}}$ and $\Sigma_{\text{anomalous}}$ correspond to the anomalous traffic. The contamination rate θ determines the proportion of anomalies introduced into the dataset.

3.2 Pre-processing Module

The pre-processing module standardizes and normalizes the data, preparing it for practical analysis. This involves scaling features to a uniform range and potentially applying other data transformation techniques.

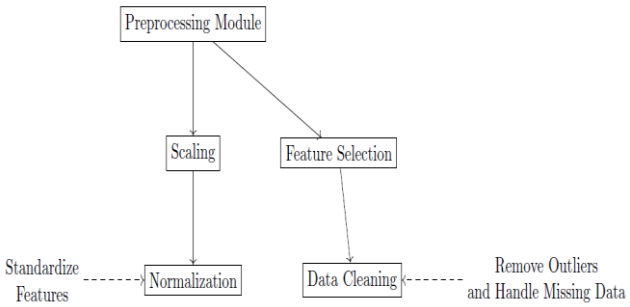

Figure 3: Pre-Processing Model

3.3 Machine Learning Model: Isolation Forest

In this module, the Isolation Forest algorithm is utilized for anomaly detection. It involves training the model on the pre-processed data and adjusting its parameters to optimize performance.

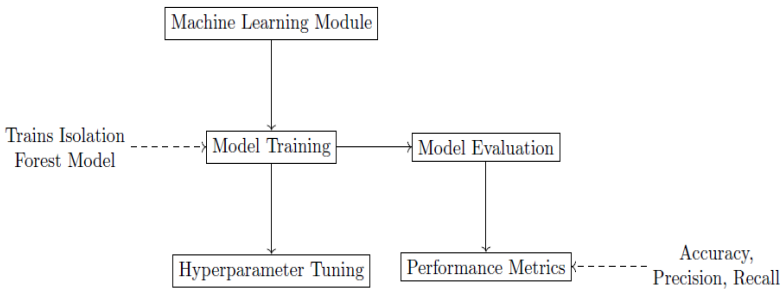

Figure 4: Machine Learning Model: Isolation Forest Model

The Isolation Forest algorithm forms the core of the anomaly detection mechanism. The fundamental principle is to isolate anomalies rather than profile standard instances. The isolation measure quantified as path length $h(x)$ is shorter for anomalies within the constructed trees. For a forest of t trees, the expected path length $E(h(x))$ and the scoring function $s(x, n)$ are defined as follows:

$$E(h(x)) = \frac{1}{t}\sum_{i=1}^{t} h_i(x)$$

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

where $h_i(x)$ is the path length of x in the i^{th} tree and $c(n)$ is the normalization term representing the average path length in an unsupervised Binary Search Tree (BST) for n instances.

3.4 Decision Engine

The decision engine interprets the anomaly detection results to make real-time decisions. This could involve triggering alerts or initiating automated responses to detect anomalies.

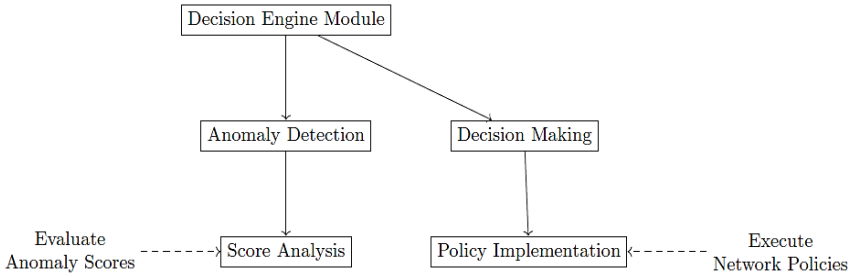

Figure 5: Decision Engine Model

The decision engine processes the anomaly scores to determine policy actions within the SDN controller. The decision function $d(x)$ is defined as:

$$\begin{cases} 1 & \text{if } s(x, n) < \tau \\ -1 & \text{otherwise} \end{cases}$$

Where τ is a threshold set following the contamination rate θ.

3.5 SDN Controller Integration

This module integrates the outcomes of the anomaly detection process into the SDN controller. It translates the model's predictions into actionable insights and policy updates for network management.

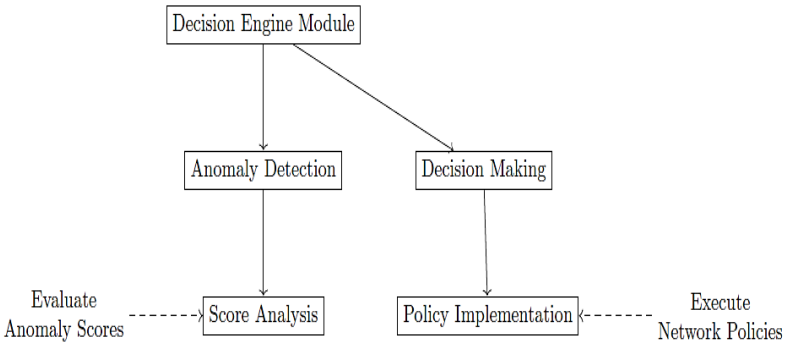

Figure 6: SDN Controller Integration

The SDN controller adopts the decisions $d(x)$ to adjust its policies accordingly. The number of policy changes $\Delta\mathcal{P}$ is a summation of the policy adjustments required across all instances X in the test set:

$$\Delta\mathcal{P} = \sum_{x \in X} \mathbb{I}(d(x) = -1)$$

3.6 Alert System Simulation

The alert system module is responsible for notifying the network administrators of potential threats or anomalies detected by the model, enabling timely interventions.

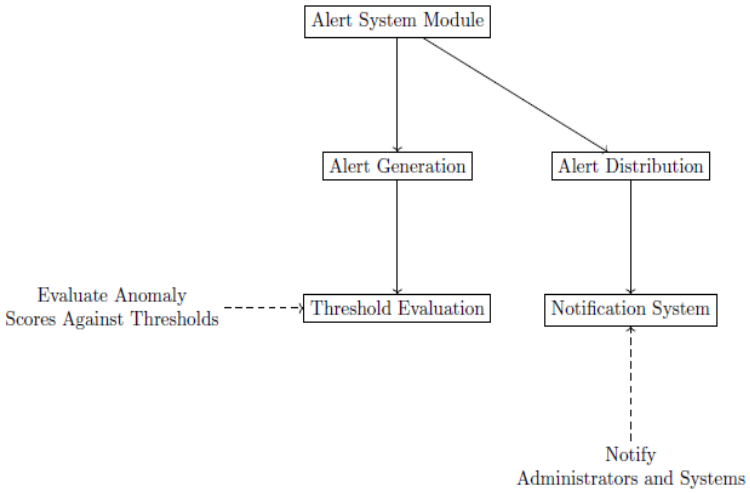

Figure 7: Alert System Model

The alert system acts upon the decisions rendered by the machine learning model. The total number of alerts triggered A is given by:

$$A = \sum_{x \in X} \mathbb{I}(d(x) = -1)$$

3.7 Feedback Loop

This module allows for continuous improvement of the machine learning model based on its performance and additional data inputs. It refines the model to enhance accuracy and adapt to evolving network patterns.

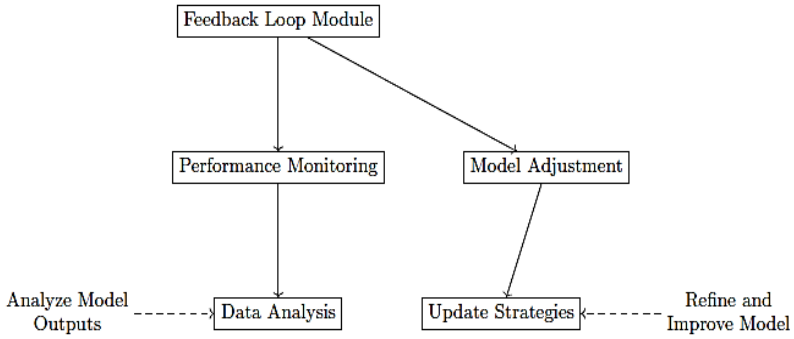

Figure 8: Feedback Loop Module

The feedback loop enhances the model by recalibrating it based on the initial decisions. The updated model M' is obtained by retraining with additional labeled instances $(\mathrm{X}', \mathrm{Y}')$:

$$\mathrm{M}' = \mathrm{Train}\ (\mathrm{M}, \mathrm{X}', \mathrm{Y}')$$

3.8 Performance Evaluation

Precision, recall, and the area under the Receiver Operating Characteristic (ROC) curve are the performance metrics that evaluate the model's efficacy. Precision and recall are derived from the confusion matrix C, and the ROC curve is plotted using true favorable rates TPR against false positive rates FPR, defined as:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},\ \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

The area under the curve (AUC) provides a scalar value quantifying the overall performance:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(\mathrm{FPR}^{-1}(\mathrm{u}))\mathrm{du}$$

3.9. Full Flow Algorithm

Algorithm 1 delineates a structured approach for detecting anomalies within Software-Defined Networking (SDN) using the Isolation Forest method. This approach systematically generates synthetic data, processes it, models it, and then evaluates the presence of anomalies, catering to the dynamic needs of SDN security frameworks.

Algorithm 1: Anomaly Detection in Software-Defined Networking Using Isolation Forest

Inputs:

N : Total number of samples for synthetic data generation

M : Number of features in each data sample

Θ: Set of contamination rates for different simulations

$\mathcal{O}$: Output directory for saving results

Outputs:

A collection of results from the anomaly detection process, stored in $\mathcal{O}$

Algorithm Procedure:

Setup Environment:

Create necessary directories in $\mathcal{O}$ for storing results.

Initialize an empty list R for ROC curve data.

Perform Simulations:

For each θ in Θ do:

Data Generation:

$X, y \leftarrow$ 'GenerateSyntheticData (N, M, Itheta) '

Pre-processing:

$X' \leftarrow$ 'StandardizeData (X) '

Model Training:

Split X' into X_{train} , X_{test}

$\mathcal{M} \leftarrow$ 'TrainIsolationForest(X_{train}, Itheta)

Anomaly Detection:

Predictions, scores $\leftarrow$ 'ApplyDecisionEngine (\ mathcal $\{M\}, X\_\{\text{test }\})$ '

SDN Integration and Alert Simulation:

Simulate policy updates and alerts in the SDN environment.

Feedback Loop:

Update $\mathcal{M}$ based on system feedback.

Save Results:

Store results of the simulation in $\mathcal{O}$.

End for

The execution of Algorithm 1 facilitates a robust anomaly detection process, yielding an array of evaluative results that are systematically archived for analysis. This algorithm efficiently orchestrates simulation, detection, and feedback mechanisms that are foundational for reinforcing the security posture of SDN environments.

IV. RESULTS & DISCUSSION

This section delves into the comprehensive analysis of the results obtained from the simulations conducted to evaluate the efficacy of the Isolation Forest model in network anomaly detection. It encompasses a detailed examination of various metrics such as classification accuracy, precision, recall, F1-score, AUC-ROC, AUC-PR, and the impact of detected anomalies on SDN policy changes, providing a holistic view of the model's performance across different simulated environments.

4.1 Overview of Synthetic Data

The synthetic dataset comprises several key network traffic features, each representing a dimension of the underlying network behavior:

- **packet_rate** and **byte_rate** describe the frequency and volume of data packets transferred across the network, respectively, providing insight into the overall network traffic load.
- **session_duration** captures the temporal aspect of network connections, allowing for analyzing long-standing versus transient communication patterns.

Table 1: Summary of Synthetic Data

Features	Simulation 1	Simulation 1	Simulation 1	Simulation 1	Simulation 2	Simulation 2	Simulation 2	Simulation 2	Simulation 3	Simulation 3	Simulation 3	Simulation 3
	mean	std	min	max	mean	std	min	max	mean	std	min	max
packet_rate	0.049094	1.042932	-1.71713	3.472813	-0.02063	1.082283	-1.75783	2.645359	0.006661	0.977062	-1.57598	2.35641
byte_rate	0.039	0.969	-	3.547	0.051	1.045	-	2.844	-	0.993	-	2.054

	146	361	1.78233	412	605	63	1.43984	426	0.05093	63	1.55305	805
session_duration	0.007393	1.034251	-1.87437	3.306325	0.026113	0.955127	-1.1782	3.139 47	-0.04994	1.002011	-1.97435	2.795502
error_rate	0.019281	1.040226	-1.80225	3.967443	0.047696	0.997336	-1.60459	2.528878	-0.07816	0.981879	-2.18501	2.325842
packet_size_avg	0.000638	0.982103	-1.95408	3.583781	0.055238	1.024431	-1.43595	2.432533	-0.01512	0.994021	-2.04788	2.047103
traffic_volume	0.020951	0.967489	-1.75569	3.042338	0.066992	1.087694	-1.74879	2.915375	-0.00212	0.952327	-2.25852	2.246914
port_usage_rate	-0.01592	0.947257	-1.64372	3.102819	0.016264	1.048595	-2.12473	2.641 14	-0.04303	0.955542	-1.72307	2.168273
protocol_type	-0.02405	0.947361	-1.86679	3.509376	0.038866	1.012228	-1.54835	2.6462	-0.01546	0.984516	-1.64299	2.344399
traffic_entropy	-0.01635	1.023647	-1.60717	3.816236	-0.00242	1.036696	-1.45131	2.749156	-0.05864	1.025515	-1.77869	2.3547
signal_strength	-0.03016	1.024642	-2.36991	4.084417	0.00603	1.01541	-1.85707	2.718774	-0.02257	0.957965	-1.52762	2.477996
label	0.105	0.307323	0	1	0.22	0.415286	0	1	0.285	0.452547	0	1

- **error_rate** offers a measure of transmission integrity, with higher rates potentially indicative of network issues or malicious activity.
- **packet_size_avg** reflects the average payload capacity used in network transmissions, which may vary according to the nature of the traffic, whether it be bulk data transfers or regular browsing activities.
- **traffic_volume** quantifies the total communication over the network, which is essential for understanding network capacity usage.
- **port_usage_rate** provides information on the distribution of traffic across different service ports, a feature that can be crucial for identifying unusual traffic patterns often associated with security breaches.
- **protocol_type** indicates the communication protocols in use, a vital determinant of the network's operational profile.
- **traffic_entropy** measures the randomness in network traffic, where higher entropy can signal complex and possibly suspicious traffic patterns.
- **signal_strength** is relevant in scenarios where wireless communications are involved, affecting the reliability and quality of the network service.

The label serves as the ground truth, differentiating between standard (0) and anomalous (1) traffic, which is essential for supervised learning.

The Isolation Forest algorithm's performance can be dissected through the lens of the generated features. The packet_rate and byte_rate directly influence the model's ability to discern patterns in the data flow intensity, while session_duration offers contextual clues about the persistence of traffic, which may be pivotal in detecting slow and stealthy data exfiltration attempts. Anomalies in error_rate and packet_size_avg might suggest packet corruption or manipulation, often associated with network attacks. Traffic_volume and port_usage_rate are instrumental in identifying volumetric anomalies and port-scanning activities. The variance in protocol_type aids

in understanding the model's sensitivity to using different protocols, which attackers might exploit to bypass traditional detection mechanisms. traffic_entropy is particularly telling, as anomalies often manifest as deviations from established traffic patterns. Finally, signal-strength fluctuations in a wireless network scenario could unveil interference or jamming attempts, which are critical for maintaining network integrity. The synthetic data, with its rich set of features, provides a comprehensive playground for the Isolation Forest model to learn and identify network anomalies, with the potential to highlight the strengths and weaknesses of the anomaly detection approach in various simulated network environments. The results from this simulation study could inform the development of more robust network anomaly detection systems and guide the implementation of adequate security measures.

4.2 Model Hyperparameters

The hyperparameters for the Isolation Forest model, as summarized in Table 2, are critical to its performance and effectiveness in anomaly detection within the network traffic data. The choice and tuning of these hyperparameters are driven by the underlying assumptions of the model and the characteristics of the data being analyzed. Here is a detailed discussion of the rationale behind the chosen hyperparameters:

Table 2: Model Hyperparameters Summary

Parameters	Simulation 1	Simulation 2	Simulation 3
n_estimators	100	100	100
contamination	0.1	0.2	0.3

The impact of these hyperparameters on the model's performance can be multifold:

- The constant **n_estimators** across simulations ensure that any changes in the model's detection ability are primarily due to variations in the **contamination** parameter rather than differences in the number of trees.
- Increasing the **contamination** rate is expected to lower the threshold for detecting anomalies, which could result in a higher actual positive rate and potentially increase the false positive rate.
- The performance trade-off for different contamination levels is likely to be evident in the precision-recall characteristics of the model. For instance, a lower contamination rate may lead to higher precision (fewer false positives), while a higher contamination rate may improve recall (fewer false negatives) at the expense of precision.

4.3 Anomaly Detection Performance

We evaluate the performance of the Isolation Forest anomaly detection algorithm across different simulated network environments. The analysis focuses on the algorithm's ability to identify anomalous traffic patterns accurately, considering varying levels of presumed anomaly prevalence as dictated by the contamination hyperparameter. This section interprets the results to discern the efficacy and robustness of the model in detecting network anomalies, which is critical for ensuring network security and integrity.

4.3.1 Classification Metrics

The classification metrics presented in Table 3 reveal a distinct progression in the performance of the Isolation Forest algorithm across the simulations as the contamination rate increases. Here is a concise analysis:

- **Precision (Class 1)**: Precision is the proportion of actual positive results among all cases labeled as positive by the model. The extremely low precision in Simulation 1 suggests that the model did not correctly identify any true anomalies, which is corroborated by the zero values for precision, recall, and F1-score. As the contamination rate increases in Simulations 2 and 3, there is a slight improvement in precision, indicating that while the model begins to identify anomalies, it still struggles with a high rate of false positives.
- **Recall (Class 1)**: Recall measures the proportion of actual positives correctly identified by the model. The increasing recall values from Simulation 1 to Simulation 3 suggest that the model becomes better at detecting true anomalies as the contamination rate increases, but this also comes with a higher number of false positives.
- **F1-Score (Class 1)**: The F1-score is the harmonic mean of precision and recall, balancing the two metrics. The low F1 scores indicate that the model performs poorly regarding precision and recall balance across all

simulations. Even in Simulation 3, where the F1-score is highest, it remains low, indicating that the model's ability to detect anomalies accurately is limited.

- **Accuracy**: Accuracy is the proportion of actual results (both true positives and negatives) among the total number of cases examined. The low accuracy across all simulations indicates that the model is ineffective in correctly classifying normal and anomalous instances. This is particularly problematic in anomaly detection scenarios where the cost of misclassification can be significant.

Table 3: Classification Metrics Comparison

Simulation	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Accuracy
Simulation 1	0	0	0	0
Simulation 2	0.006667	0.022727	0.010309	0.005
Simulation 3	0.072993	0.175439	0.103093	0.05

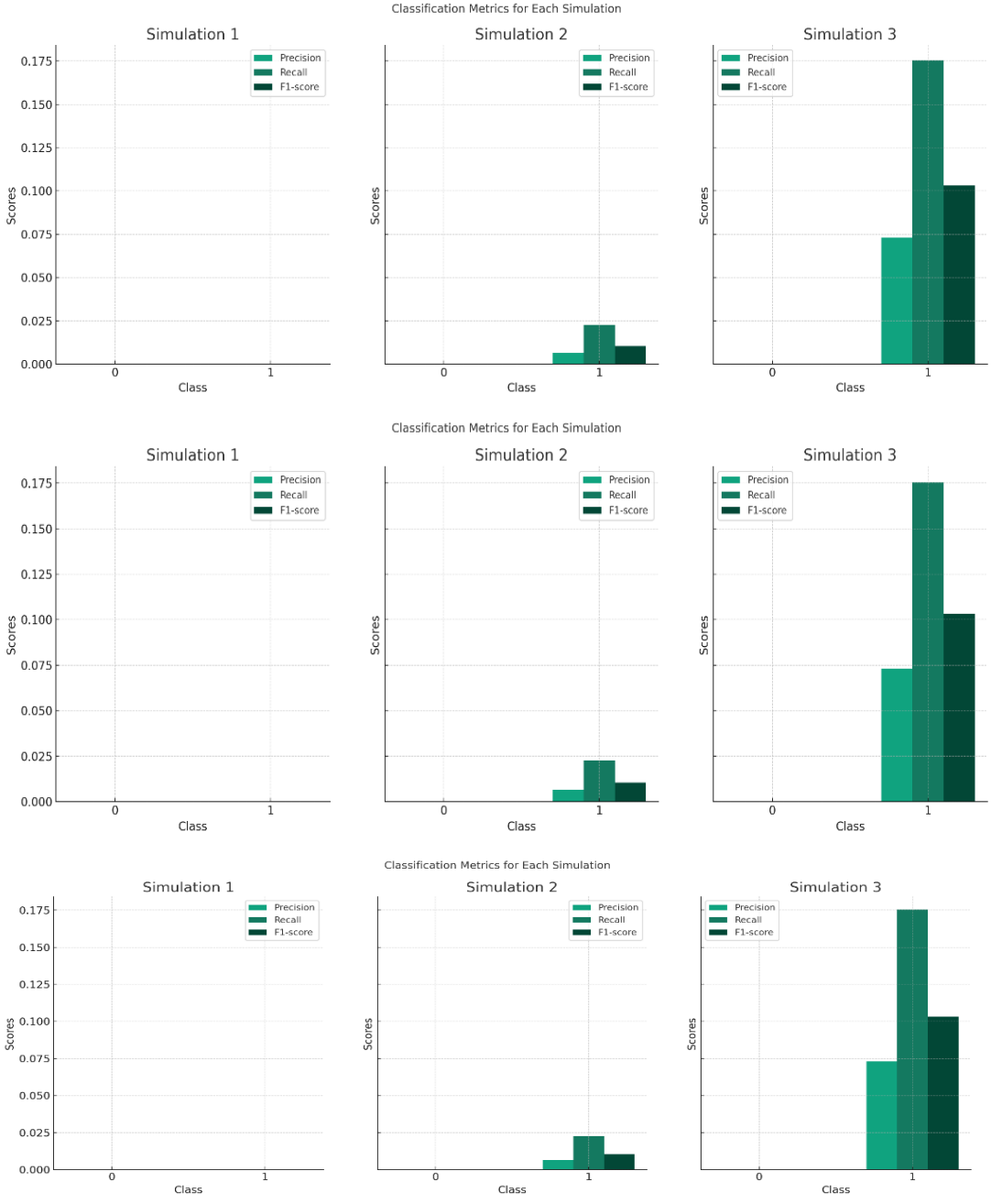

Figure 9: Classification Matrics

The classification metrics suggest that the model's performance is not optimal for the task at hand, with a particularly pronounced issue in correctly classifying anomalies. The metrics indicate a model that is currently

ineffective, likely requiring further tuning, more representative training data, or a re-evaluation of the model's suitability for this specific anomaly detection application.

4.3.2 Confusion Matrix Analysis

Confusion matrices were employed as a critical tool for assessing the performance of the Isolation Forest algorithm in anomaly detection within SDN environments. These matrices provided crucial insights into the accuracy of the model's classifications, revealing the proportion of true positives, true negatives, false positives, and false negatives, thereby clearly depicting the model's efficacy in distinguishing between normal and anomalous network traffic.

Table 4: Summary of Confusion Matrices

Simulation	True Positives + True Negatives
Simulation 1	0
Simulation 2	1
Simulation 3	10

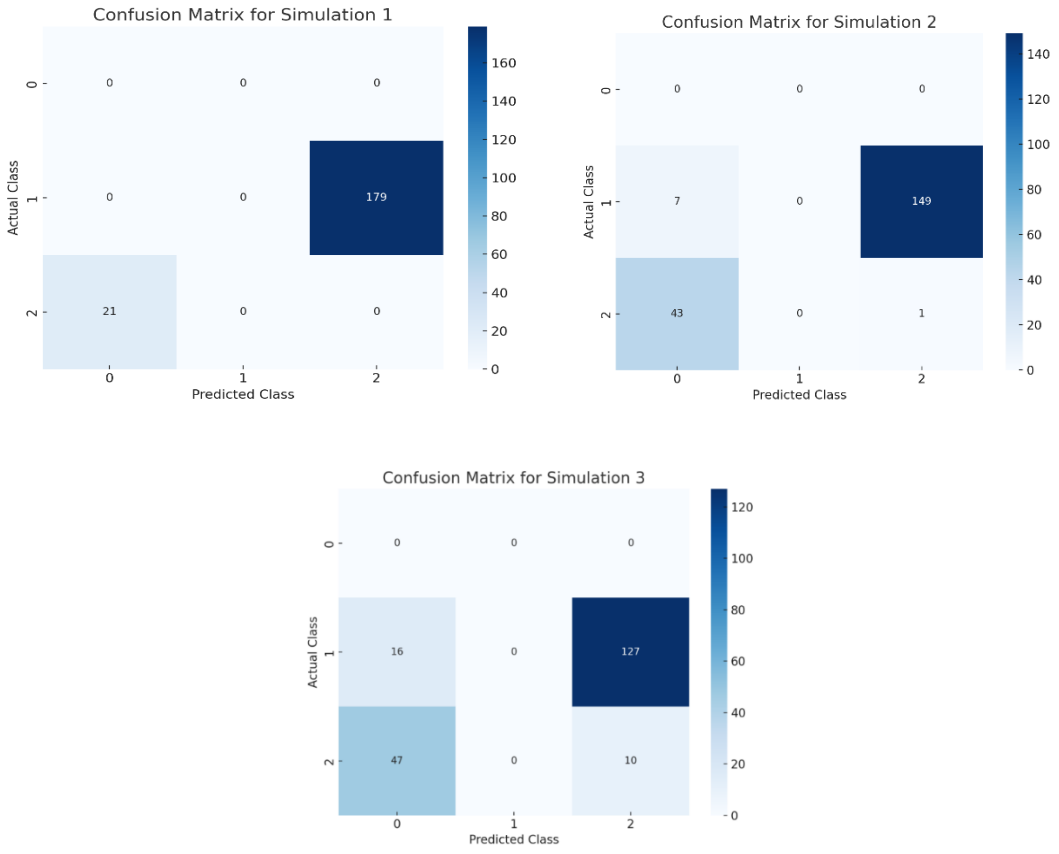

Figure 10: Confusion Matrices

4.4 Receiver Operating Characteristic (ROC) Curves

The ROC Curve Summary table quantifies the model's discriminative performance through the AUC-ROC values for each simulation. The AUC-ROC is a critical statistic in assessing classification models, encapsulating the likelihood that the model will correctly rank a random positive instance higher than a random negative one.

- **Simulation 1** registers an AUC-ROC score of 0, indicating a complete lack of discrimination between normal and anomalous instances within the data. This score implies the model has no predictive power in distinguishing between the two classes.

- The slight increase in AUC-ROC to 0.005682 in **Simulation 2** remains negligible, suggesting the model's predictive accuracy is virtually indistinguishable from a random decision.
- The AUC-ROC value for **Simulation 3** improves to 0.073733, reflecting a small but insufficient ability to differentiate between normal and anomalous network traffic. Although there is a relative increase, an AUC-ROC near this low end of the spectrum denotes a model not adequately capturing the underlying patterns necessary for adequate classification.

Table: ROC Curve Summary

Simulation	AUC-ROC
Simulation 1	0
Simulation 2	0.005682
Simulation 3	0.073733

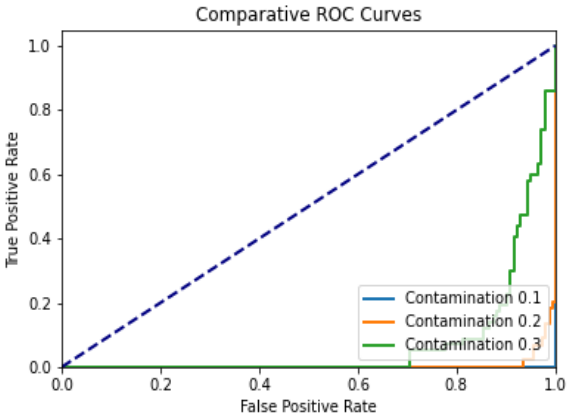

Figure 11: Comparative ROC Curve

These AUC-ROC values reflect a model that has not effectively learned from the data features provided to it. Further refinement of the model parameters, augmentation of the training data, or reconsidering the model's complexity may be required to achieve a level of performance that would be considered acceptable for operational purposes[24][25].

4.5 Precision-Recall (PR) Curve Analysis

The PR Curve Summary table provides the AUC-PR (Area Under the Precision-Recall Curve) values for the three simulations, offering a nuanced view of the model's classification ability, especially when dealing with imbalanced datasets.

- In **Simulation 1**, the AUC-PR is 0.054436, which, despite being low, suggests that when the model predicts an instance to be anomalous, it is correct approximately 5.44% of the time. However, the low value also indicates many false positives—for instance, the model incorrectly labeled anomalous.
- **Simulation 2** shows an AUC-PR of 0.119189, doubling the value from Simulation 1. This suggests an improvement in the balance between precision and recall—the model is becoming more accurate in its predictions and missing fewer actual anomalies.
- **Simulation 3** sees further improvement in the AUC-PR to 0.164389. This progression indicates that, as the model is exposed to a higher contamination rate, it is better at distinguishing the positive class (anomalies) from the negative, although the value still indicates room for significant improvement.

Table: PR Curve Summary

Simulation	AUC-PR
Simulation 1	0.054436
Simulation 2	0.119189
Simulation 3	0.164389

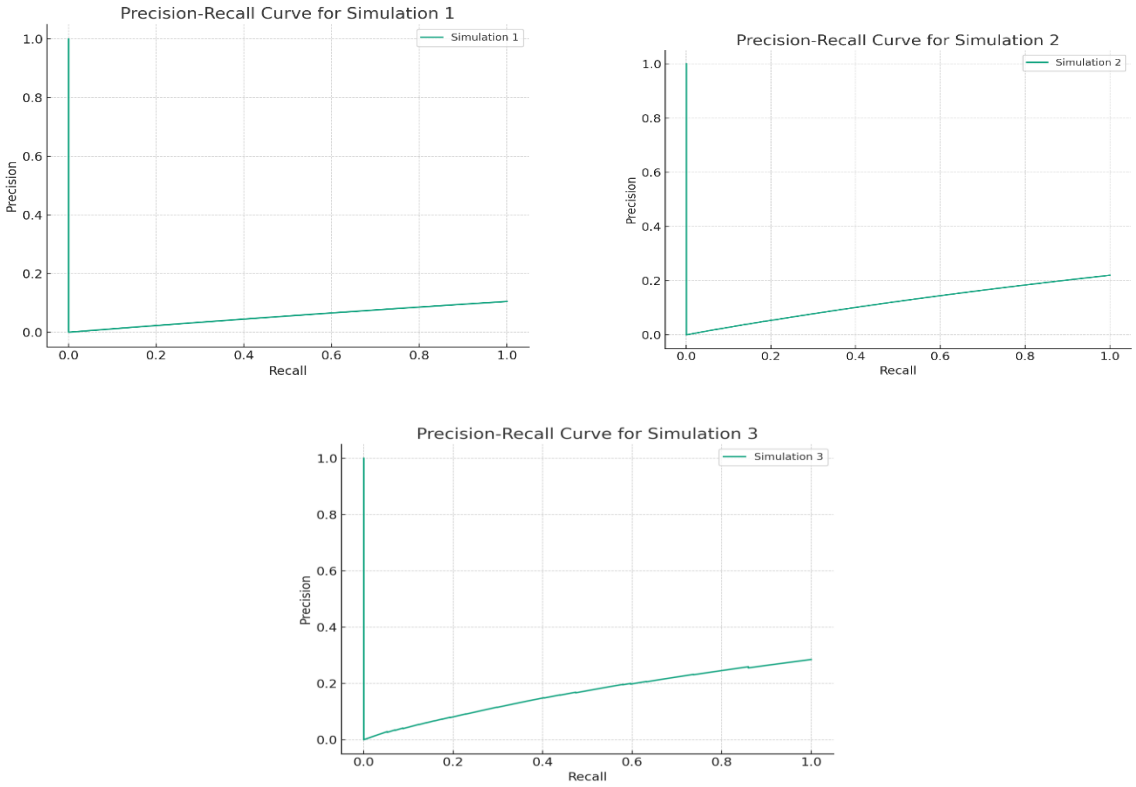

Figure 12: PR- Curves

The Precision-Recall (PR) curve is a valuable tool for understanding the trade-off between precision (the accuracy of the optimistic predictions) and recall (the model's ability to find all the positive instances). Generally, precision tends to decrease as recall increases since expanding the threshold to capture more positives increases the chance of including negatives. In this context, the AUC-PR provides an aggregate performance measure across this trade-off. The observed incremental improvements suggest that the model's ability to identify true anomalies increases with each simulation. However, it is not yet at a level that can be considered adequate for reliable classification, especially in scenarios where the cost of false positives or false negatives is high.

4.6 SDN Policy Changes Analysis

The SDN Policy Changes Summary table enumerates the number of policy alterations necessitated by the model's anomaly detection outcomes across three simulation scenarios.

- **Simulation 1** led to 21 policy changes. This indicates that the model's predictions triggered a moderate number of updates to the network policies, potentially reflecting a conservative threshold for anomaly detection or lower overall detection sensitivity.
- In **Simulation 2**, the number of policy changes increased to 50. The rise in adjustments suggests that the model identified more behavior as anomalous, possibly due to the higher contamination rate influencing the model's detection threshold.
- **Simulation 3** increased to 63 policy changes, the highest among the simulations. This indicates an even more aggressive response to detected anomalies, aligning with the highest contamination rate set for this simulation, which implies a more liberal anomaly detection strategy.

Table 5: SDN Policy Changes Summary

Simulation	Number of Policy Changes
Simulation 1	21
Simulation 2	50
Simulation 3	63

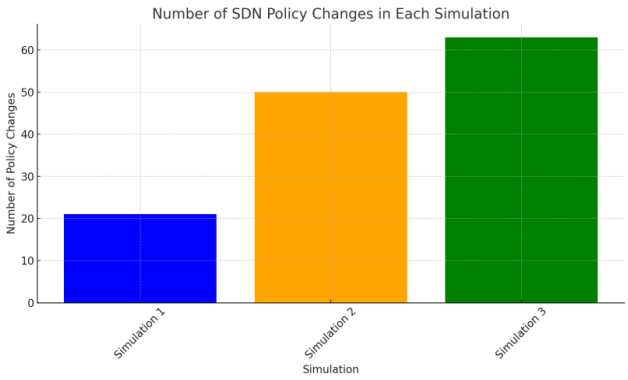

Figure 13: SND Policies

The implications of these policy changes are multifaceted. On the one hand, more policy changes can indicate a more dynamic and responsive approach to network security, potentially leading to a more secure environment by quickly adapting to detected threats. On the other hand, frequent changes can also lead to instability and unpredictability in the network's behavior, disrupting users and systems relying on consistent network policies. Excessive policy changes may also reflect a high rate of false positives in anomaly detection, which can lead to unnecessary policy updates and could erode trust in the automated system. Network administrators must balance the sensitivity of the anomaly detection system to maintain an optimal level of security without overburdening the network with excessive modifications.

4.6.1 Alert Triggers

The bar chart illustrates the total number of alerts triggered in the three simulations. The chart shows a clear increasing trend in alerts from Simulation 1 to Simulation 3. Specifically, Simulation 1 has the lowest number of triggered alerts, represented by the blue bar. Simulation 2, denoted by the orange bar, shows a substantial increase in alerts compared to Simulation 1. Simulation 3, indicated by the green bar, records the highest number of alerts, surpassing the counts of both prior simulations. This pattern suggests a correlation between the simulation parameters—potentially the increasing contamination rates—and the frequency of alert triggers, indicating a more sensitive or responsive system detecting anomalies as the simulation number increases.

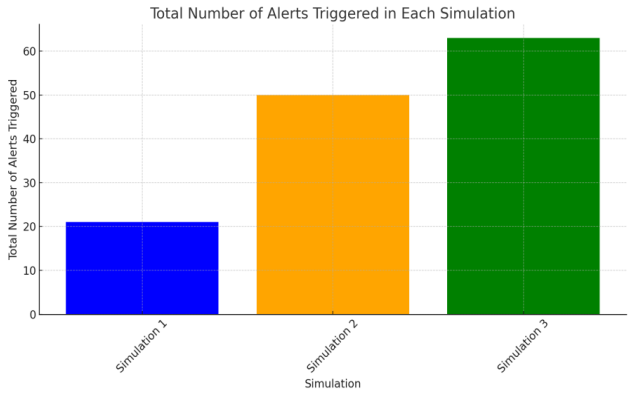

Figure 14: Alert Trigger Analysis

In conclusion, the results and discussions presented in this paper highlight the nuanced performance of the Isolation Forest algorithm in detecting network anomalies. While the model shows incremental improvements in specific metrics across simulations, the overall effectiveness remains limited, as evidenced by low precision, recall, F1 scores, and AUC values. Additionally, the increasing trend in SDN policy changes with higher contamination rates underscores the delicate balance between maintaining network security and operational stability. These findings offer critical insights for further refining anomaly detection approaches in network environments and underscore the importance of continuous evaluation and adaptation of such systems.

V. CONCLUSION AND FUTURE WORK

This research primarily contributed to exploring machine learning integration, explicitly using the Isolation Forest algorithm in a Software-Defined Networking (SDN) environment for anomaly detection. The study involved generating synthetic data to simulate various network traffic scenarios and then applying the Isolation Forest model to detect anomalies within this data. The model's performance was meticulously evaluated across three simulations with varying contamination rates, providing a comprehensive analysis of its effectiveness. The results revealed that while the model showed some capability in identifying anomalies, as evidenced by the incremental improvements in classification metrics and AUC scores from Simulation 1 through Simulation 3, its overall effectiveness was limited. Precision and recall values remained low across all simulations, with F1 scores and accuracy metrics reinforcing this observation. Although showing slight improvements in later simulations, the AUC-ROC and AUC-PR values were not at levels indicative of a robust anomaly detection system. Furthermore, the increasing number of policy changes in the simulated SDN environment, correlating with higher contamination rates, highlighted the model's responsiveness and raised concerns regarding potential over-sensitivity and operational stability.

5.1 Machine Learning Integration in SDN for Anomaly Detection

The findings of this study suggest that while machine learning, notably the Isolation Forest algorithm, holds promise for anomaly detection in SDN environments, significant challenges must be addressed to enhance its effectiveness. The current integration approach demonstrates potential but requires further refinement to achieve the accuracy and reliability needed for practical deployment in network security.

5.2 Future Work

Given the findings, future research could take several directions:

- **Model Refinement and Optimization**: Investigating alternative machine learning models or more advanced versions of ensemble methods could yield better results. Additionally, fine-tuning the existing model by experimenting with different hyperparameters and feature engineering techniques might improve its performance.
- **More Representative Datasets**: Utilizing real-world network traffic data, possibly augmented with synthetic anomalies, could provide a more realistic training and testing environment for the model. This would help in understanding the model's applicability in real-world scenarios.
- **Adaptive Learning Mechanisms**: Implementing adaptive learning where the model continually updates itself based on new data and feedback loops could improve its accuracy and adapt to evolving network behaviors.
- **Hybrid Approaches**: Combining machine learning models with rule-based systems or other traditional anomaly detection techniques could offer a more robust solution.
- **Evaluation of Operational Impact**: Further research should also focus on the operational impacts of integrating such models into SDN, particularly the balance between security and network stability and the practicality of frequent policy updates.

In conclusion, this research underscores the potential of machine learning integration in SDN to enhance network security through anomaly detection while highlighting the areas that require further exploration and improvement.

REFERENCES

[1] Q. Meng, X. Pang, Y. Zheng, G. Jiang, and X. Tian, "Development and optimization of software defined networking anomaly detection architecture by GRU-CNN under deep learning," in Proc. 2021 6th Int. Conf. Intelligent Computing and Signal Processing (ICSP), 2021.

[2] N. M. Raja and S. Vegad, "An empirical study for the traffic flow rate prediction-based anomaly detection in software-defined networking: a challenging overview," Soc. Netw. Anal. Min., vol. 13, no. 1, 2023.

[3] Alpana Gopi, Divya P R, Litty Rajan, Surya Rajan, & Shini Renjith. (2016). Accident Tracking and Visual Sharing Using RFID and SDN. *International Journal of Computer Engineering in Research Trends*, *3*(10), 544–549.

[4] M. Kalpana Devi, & R. Padmaja. (2023). Outlier Detection using Artificial Rabbit Optimizer with Hopfield Neural Network. *International Journal of Computer Engineering in Research Trends*, *10*(9), 9–15.

[5] A. M. El-Shamy, N. A. El-Fishawy, G. Attiya, and M. A. A. Mohamed, "Anomaly detection and bottleneck identification of the distributed application in cloud data center using software–defined networking," Egypt. Inform. J., vol. 22, no. 4, pp. 417–432, 2021.
[6] Asep Bayu Dani Nandiyanto, Chekima Hamza, & Muhammad Aziz. (2023). A Novel Framework for Enhancing Security in Software-Defined Networks. *International Journal of Computer Engineering in Research Trends*, *10*(11), 19–26.
[7] P. S. K. Reddy and K. Sri Raghavendra, "Machine Learning-Based DDoS Saturation Attack Detection and analysis in SDN Environment," Int. J. Computer Engineering in Research Trends, no. 9, pp. 269–274, 2022.
[8] F. Chahlaoui and H. Dahmouni, "A taxonomy of load balancing mechanisms in centralized and distributed SDN architectures," SN Comput. Sci., vol. 1, no. 5, 2020.
[9] Arpita Nusrat, Jasni Mohamad Zain, Mohamed Lachgar, & M.Bhavsingh. (2023). Machine Learning Techniques for Detecting Anomalies in IoT Networks . *International Journal of Computer Engineering in Research Trends*, *10*(10), 16–23.
[10] M. S. Bonfim, K. L. Dias, and S. F. L. Fernandes, "Integrated NFV/SDN architectures: A Systematic Literature Review," arXiv [cs.NI], 2018.
[11] O. Blial, M. Ben Mamoun, and R. Benaini, "An overview on SDN architectures with multiple controllers," J. Comput. Netw. Commun., vol. 2016, pp. 1–8, 2016.
[12] Bezawada , M., & P, V. K. (2023). Comparative Study on Techniques Used for Anomaly Detection in IoT Data. *International Journal of Computer Engineering in Research Trends*, *10*(4), 177–181.
[13] K. Kalkan and S. Zeadally, "Securing internet of things with software defined networking," IEEE Communications Magazine, vol. 56, no. 9, pp. 186-192, 2017.
[14] P. Siva, Cherukuri Sudhish, Ogirala Divyanand, & K Sai Ananya Madhuri. (2023). Routenet: Using Graph Neural Networks for SDN Network Modeling and Optimizations. *International Journal of Computer Engineering in Research Trends*, *10*(7), 32–38.
[15] H. Deng and X. Li, "Anomaly detection via reverse distillation from one-class embedding," arXiv [cs.CV], 2022.
[16] C. Yin, S. Zhang, J. Wang, and N. N. Xiong, "Anomaly detection based on convolutional recurrent autoencoder for IoT time series," IEEE Trans. Syst. Man Cybern. Syst., vol. 52, no. 1, pp. 112–122, 2022.
[17] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data," arXiv [cs.LG], 2022.
[18] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "ADBench: Anomaly Detection Benchmark," arXiv [cs.LG], 2022.
[19] M. A. Akhtar, S. M. O. Qadri, M. A. Siddiqui, S. M. N. Mustafa, S. Javaid, and S. A. Ali, "Robust genetic machine learning ensemble model for intrusion detection in network traffic," Sci. Rep., vol. 13, no. 1, 2023.
[20] A. B. D. Nandiyanto, C. Hamza, and M. Aziz, "A Novel Framework for Enhancing Security in Software-Defined Networks," Int. J. Comput. Eng. Res. Trends, vol. 10, no. 11, pp. 19–26, 2023.
[21] C. Guerber, M. Royer, and N. Larrieu, "Machine Learning and Software Defined Network to secure communications in a swarm of drones," J. Inf. Secur. Appl., vol. 61, no. 102940, p. 102940, 2021.
[22] Ali Vatankhah Barenji, Yaling Zhang, & M Bhavsingh. (2023). A Blockchain-based Framework for Enhancing Privacy and Security in Online Transactions . *International Journal of Computer Engineering in Research Trends*, *10*(11), 1–9.
[23] Y.-K. Kim, J. J. Lee, M.-H. Go, H. Y. Kang, and K. Lee, "A systematic overview of the machine learning methods for mobile malware detection," Secur. Commun. Netw., vol. 2022, pp. 1–20, 2022.
[24] Prasad , C. G. V. N. ., Mallareddy, A., Pounambal, M., & Velayutham, V. . (2022). Edge Computing and Blockchain in Smart Agriculture Systems. *International Journal on Recent and Innovation Trends in Computing and Communication*, *10*(1s), 265–273.
[25] Pasha, M. J., Rao, K. P., MallaReddy, A., & Bande, V. (2023). LRDADF: An AI enabled framework for detecting low-rate DDoS attacks in cloud computing environments. *Measurement: Sensors*, 100828.