IJWR

Enhancing Software-Defined Networking (SDN) Resilience against Cyberattacks: A Markov Model-Based Approach

Ahmad Jalili

Department of Computer Engineering, Faculty of Basic Sciences and Engineering, Gonbad Kavous University, Basirat Blvd., Shahid Fallahi St., Gonbad Kavous 49717-99151, Iran; jalili@gonbad.ac.ir

ABSTRACT

Software-Defined Network (SDN) introduces centralized network control via the OpenFlow protocol, enhancing network management, traffic routing, and security policy enforcement. However, SDN's centralized nature also introduces vulnerabilities, particularly to cyberattacks targeting the controller and communication channels. This study presents a resilience assessment methodology for SDN under cyberattack conditions, leveraging Markov process theory to model system states and transitions. Three SDN architectures were evaluated under various attack scenarios, revealing that traditional configurations lack sufficient resilience against synchronous attacks and controller breaches. To address these vulnerabilities, we propose an enhanced SDN protection framework integrating controller redundancy, automatic reconfiguration mechanisms, and anomaly detection using Long Short-Term Memory (LSTM) networks. The methodology was validated through simulations in the EVE-NG environment, demonstrating improved SDN stability under cyber threats. These findings provide a foundation for designing more resilient SDN infrastructures, ensuring network continuity and security against evolving cyber threats.

Keywords*— *Software-Defined Networks, Cyberattacks, Long Short-Term Memory, Markov Model.

1. Introduction

The Software-Defined Networking (SDN) has emerged as a transformative paradigm in network management, offering centralized control, dynamic programmability, and improved network agility. By decoupling the control plane from the data plane, SDN enables flexible and automated network configurations through a logically centralized controller [1]. The OpenFlow protocol, a widely adopted SDN standard, facilitates communication between network devices and the controller, enhancing operational efficiency.

However, the centralized nature of SDN introduces significant security challenges. The SDN controller, being a single point of failure, is vulnerable to various cyberattacks, such as Denial-of-Service (DoS), topology poisoning, malicious rule injection, and controller hijacking [2]. A successful attack on the controller can compromise the entire network, leading to traffic manipulation, service disruptions, and unauthorized data access. Traditional security mechanisms, such as firewalls and intrusion detection systems (IDS), are insufficient in mitigating these risks due to their reactive nature and limited adaptability [3].

Despite advancements in SDN security, existing approaches face several critical challenges: (i) Single Point of Failure: Centralized SDN architectures are highly susceptible to targeted attacks on the controller, which can paralyze the entire network. (ii) Lack of Proactive Resilience Models: Most security solutions focus on reactive defenses, failing to quantify SDN resilience against evolving attack patterns. (iii) Limited Adaptability of Existing Defense Mechanisms: Traditional security measures do not dynamically adjust to real-time attack conditions, leading to delayed response and mitigation. (iv) Absence of AI-Driven Anomaly Detection: Existing SDN security frameworks lack machine learning-based threat prediction, making them ineffective against zero-day attacks and sophisticated cyber threats [4, 5].

http://dx.doi.org/10.22133/ijwr.2025.505823.1266

Citation A. Jalili, " Enhancing Software-Defined Networking (SDN) Resilience against Cyberattacks: A Markov Model-Based Approach", *International Journal of Web Research*, vol.8, no.2 ,pp.25-39, 2025, doi: http://dx.doi.org/10.22133/ijwr.2025.505823.1266.

**Coressponding Author*

Article History: Received:10 January 2025 ; Revised: 17 March 2025 ; Accepted: 27 March 2025.

These challenges highlight the need for a proactive and adaptive security framework that can dynamically assess SDN resilience, detect anomalies in real-time, and mitigate attacks efficiently.

To address the aforementioned challenges, this study aims to: (1) Develop a Markov-based Analytical Model for quantifying SDN resilience under cyberattack conditions. (2) Implement an LSTM-based Anomaly Detection System for real-time attack prediction and mitigation. (3) Evaluate the Effectiveness of Controller Redundancy Mechanisms to enhance fault tolerance and prevent service disruptions. (4) Assess the Performance of Proactive Traffic Filtering Mechanisms (e.g., Open vSwitch Agents) in mitigating malicious traffic. (5) Empirically Validate the Proposed Security Framework through simulation-based attack modeling in an EVE-NG virtual environment.

This research not only provides a quantitative assessment of SDN security but also offers practical solutions to enhance network resilience against evolving cyber threats.

The key contributions of this paper include: Markov-Based Resilience Model: A probabilistic framework for assessing SDN stability under various attack scenarios, capturing transitions between different network states. AI-Driven Attack Detection: Implementation of a Long Short-Term Memory (LSTM)-based anomaly detection system, achieving 98.1% accuracy in cyberattack detection. Proactive Controller Failover Mechanism: A multi-controller setup with automated failover, reducing controller recovery time from 6.3s to 3.1s, thereby enhancing SDN robustness. Efficient Traffic Isolation with Software Router Agents: Deployment of Open vSwitch (OvS) agents to proactively filter malicious flows, blocking 95% of attack traffic with minimal latency overhead. Empirical Validation through Simulation: A comprehensive EVE-NG-based simulation evaluating SDN resilience across three architectures (single-controller, dual-controller, and hybrid AI-enhanced SDN).

The rest of the paper is organized as follows: Section 2 provides a comprehensive review of existing SDN security approaches, highlighting research gaps. Section 3 details the Markov-based resilience model, attack scenarios, and the proposed AI-driven security framework. Section 4 presents the experimental setup, simulation results, and performance evaluation. Section 5 discusses the conclusions, limitations, and future research directions.

2. Related Works

Software-Defined Networking (SDN) has gained significant attention due to its centralized control, flexibility, and programmability. However, the inherent vulnerabilities of SDN, particularly its reliance on a logically centralized controller, have made it an attractive target for cyberattacks [6]. This section reviews existing research on SDN resilience and security, categorizing prior works into three major approaches: route optimization for resilience, structural resilience mechanisms, and heuristic-based security models. It also highlights the research gaps that motivate the development of a Markov-based resilience model combined with AI-driven threat detection.

2.1. Route Optimization for SDN Resilience

One of the key challenges in SDN security is ensuring efficient routing in the presence of cyberattacks. Several studies have focused on optimizing routing algorithms to minimize network disruption during attack scenarios. Traditional shortest-path algorithms, such as Dijkstra's algorithm, have been improved by incorporating attack-aware constraints. For example, researchers have proposed weighted graph models that assign dynamic security weights to different paths in the network, allowing SDN controllers to select routes that minimize exposure to potential threats. Studies [7–9] have explored variations of the Dijkstra algorithm to enhance SDN resilience, particularly by incorporating link failure probabilities and attack likelihoods. These methods improve network robustness against denial-of-service (DoS) attacks, but they do not provide proactive attack detection or real-time mitigation.

To address the challenge of collision-prone routing, alternative optimization techniques have been explored. The Clark-Wright heuristic method, initially developed for vehicle routing problems [10], has been adapted to SDN to optimize flow allocation under attack conditions. This approach merges multiple traffic flows into fewer paths, reducing congestion while ensuring that routing decisions are dynamically adjusted based on network conditions. Although these methods improve network efficiency, they lack a formal stochastic model for quantifying SDN resilience. Most existing approaches focus on deterministic routing optimization without accounting for the probabilistic nature of cyberattacks, which can occur at random intervals and with varying intensities [8, 9].

Another class of solutions involves game-theoretic models that formulate network security as an adversarial game between attackers and defenders [11 ,12]. These models attempt to predict attacker behavior and optimize routing policies accordingly. While game theory provides a mathematically rigorous approach, it often relies on simplistic assumptions about attacker strategies and does not adapt well to real-time network conditions. Moreover, these models generally require high

computational complexity, making them less practical for large-scale SDN deployments.

2.2. Structural Resilience Approaches

A fundamental limitation of traditional SDN architectures is their reliance on a single controller, which becomes a single point of failure [13]. Structural resilience approaches aim to improve SDN security by introducing controller redundancy, fault-tolerant topologies, and adaptive failover mechanisms. Several studies have proposed multi-controller SDN architectures where multiple controllers operate in a distributed manner, reducing the impact of targeted attacks [13, 14].

One widely studied approach is the 1+1 controller redundancy model, where an active controller is paired with a hot-standby backup controller. When the primary controller is compromised, the backup controller takes over with minimal downtime. However, studies have shown that controller synchronization delays can lead to temporary network instability, particularly in high-traffic environments. Research efforts [14–16] have explored more advanced multi-controller topologies, such as load-balanced controllers that distribute traffic dynamically across multiple instances. These architectures improve network resilience but introduce complex synchronization challenges, requiring efficient east-west communication between controllers.

In addition to controller redundancy, researchers have investigated alternative network topologies to enhance SDN fault tolerance. The FatTree topology, commonly used in data centers, has been adapted to SDN environments due to its inherent path redundancy and load-balancing capabilities [15]. Other studies have proposed hybrid topologies, such as multi-level architectures combining star and dual-ring topologies, which offer a balance between fault tolerance and scalability [16, 17]. Despite these advancements, one major limitation remains: structural redundancy alone does not prevent cyberattacks. Attackers can still exploit northbound API vulnerabilities, insert malicious flow rules, or execute DDoS attacks against multiple controllers simultaneously.

Another limitation of existing structural resilience approaches is their high implementation cost. Many redundancy-based solutions require additional hardware, sophisticated synchronization protocols, and increased computational resources, making them impractical for resource-constrained SDN deployments, such as IoT and edge networks. Furthermore, most studies do not quantify SDN resilience mathematically, making it difficult to compare the effectiveness of different resilience strategies under real-world attack scenarios.

2.3. Heuristic-Based Security Models

A third category of research focuses on heuristic-based security models, which combine machine learning, anomaly detection, and stochastic modeling to enhance SDN resilience [18-20]. These models aim to predict and mitigate cyberattacks before they cause significant damage. One of the most promising approaches in this category is the use of neural networks for intrusion detection. Studies have explored various deep learning techniques, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks, to detect abnormal traffic patterns in SDN environments.

LSTM networks, in particular, have gained attention due to their ability to capture long-term dependencies in network traffic data. Research [19] has shown that LSTM-based anomaly detection can identify cyberattacks with high accuracy, significantly reducing false positives compared to traditional rule-based intrusion detection systems. However, most existing studies focus on offline attack detection, where models are trained on historical datasets and deployed in a passive monitoring role. This limits their ability to respond dynamically to real-time threats.

Another important area of research is stochastic modeling for SDN resilience assessment. Several works [20,21] have applied Markov chains and probabilistic models to analyze network stability under cyberattack conditions. These models provide a mathematical foundation for quantifying SDN resilience, allowing researchers to calculate the probability of network failure over time. However, most existing stochastic models assume static attack probabilities, whereas real-world attacks are often adaptive and evolve dynamically. Our work extends these models by introducing a time-dependent Markov model that incorporates real-time attack detection and automated mitigation mechanisms.

In addition to heuristic models, researchers have explored hybrid security frameworks that combine machine learning with traditional SDN security techniques. For example, some studies have proposed hybrid IDS systems that integrate signature-based detection with anomaly detection to improve attack detection rates [22, 23]. Others have developed adaptive security policies that dynamically adjust network configurations based on threat intelligence data. While these approaches show promise, they often require high computational resources, making them challenging to deploy in low-latency environments.

2.4. Research Gaps and Novel Contributions

Despite significant advancements in SDN security, several key research gaps remain unaddressed. Most existing approaches focus on

either structural resilience or attack detection, but very few integrate quantitative resilience assessment with real-time anomaly detection. There is a need for a unified framework that combines probabilistic modeling, AI-based threat detection, and automated mitigation strategies.

This paper addresses these gaps by proposing a Markov-based resilience model that quantifies SDN stability under cyberattack conditions. Unlike traditional models, our approach dynamically updates attack probabilities based on real-time threat intelligence, ensuring a more accurate assessment of network resilience. Additionally, we introduce an LSTM-based anomaly detection system that provides real-time cyberattack prediction and automated response, significantly enhancing SDN security.

Furthermore, we empirically validate our approach through simulations in the EVE-NG virtual environment [24], evaluating the impact of various cyberattacks on three different SDN architectures. Our results demonstrate that combining Markov modeling with AI-driven security mechanisms leads to a 60% improvement in network resilience, significantly reducing downtime and mitigating the impact of cyber threats.

By bridging the gap between theoretical resilience modeling and practical AI-driven security solutions, this research provides a comprehensive and deployable framework for securing SDN environments.

3. Methodology

This section presents our proposed three-phase methodology for assessing and enhancing the resilience of Software-Defined Networking (SDN) under cyberattack conditions. The methodology includes: **(i)** stochastic modeling using a continuous-time Markov process, **(ii)** simulation-based validation of cyberattack scenarios, and **(iii)** the design of a protection system integrating AI-driven detection and proactive mitigation mechanisms.

3.1. Phase I: Markov Model for SDN Resilience

A comprehensive review of the existing literature concerning the resilience of computer networks under computer attacks (CA) reveals a set of critical requirements for effective resilience assessment methodologies. Specifically, these investigations emphasize the following key points:

Importance of Stochastic Analytical Modeling: Stochastic analytical modeling, particularly approaches grounded in Markov process theory, is of paramount importance for the rigorous justification and validation of security measures deployed within contemporary information security systems [25]. These methodologies enable the quantification of network behavior under uncertainty, providing insights into the probabilistic nature of attack outcomes.

Computational Efficiency of Stochastic Models: To facilitate practical application, stochastic models must exhibit computational efficiency, enabling the calculation of distribution functions for key random variables of interest (e.g., network availability, packet loss rate) with minimal computational overhead. Excessive computational complexity hinders the scalability and applicability of these models to large-scale network environments.

Flexibility and Generalizability in Attack Modeling: Resilience assessment methodologies should possess the requisite flexibility and generalizability to accurately model a broad spectrum of potential attack vectors and adapt to the evolving threat landscape. Models that are overly specific or narrowly focused may fail to capture the complexities of real-world attack scenarios [26].

However, the existing resilience assessment approaches identified in the literature often exhibit limitations in fully addressing the aforementioned requirements. In particular, many methods lack the necessary computational efficiency or fail to adequately capture the dynamic and stochastic nature of network attacks. Therefore, this paper proposes a novel SDN resilience assessment methodology predicated on Markov process theory, which demonstrably mitigates these shortcomings.

To rigorously evaluate network resilience, it is imperative to define clear and quantifiable failure criteria, specifically, the conditions under which the network ceases to perform its designated functions in accordance with its defined service-level objectives (SLOs).

Within the context of the network's transport component, network failure is defined to occur under the following conditions:

Transport Network Controller Failure or Compromise: Failure of the central transport network controller or unauthorized takeover of the controller by a malicious actor, resulting in compromised network management and control.

Failure of Critical Routers: Failure of one or more routers essential for maintaining the integrity and functionality of the network's transport infrastructure.

Topology Spoofing Attacks: Insertion of rogue routers into the transport network via topology spoofing, leading to the creation of "black holes" and the disruption of network traffic flows. The malicious router becomes an attractor for network packets.

Communication Link Failures: Loss of connectivity along one or more communication channels between network nodes, resulting in

degraded network performance or complete service disruption.

Building upon these failure criteria, this research assesses the resilience of an SDN architecture incorporating redundant network elements. The network state is modeled as a **Markov process with discrete states evolving in continuous time**. The sojourn time (the time spent in each state) is assumed to follow an exponential distribution, a common and analytically tractable assumption in network reliability modeling [23, 25]. This allows us to model transitions between operational states (e.g., fully operational, degraded, failed) based on probabilistic rates.

We model SDN resilience as a continuous-time Markov chain (CTMC), where the network transitions between different operational states depending on cyberattack events and recovery mechanisms.

To apply this model, we first define the distinct operational states the SDN system can occupy under potential cyberattacks. We define **five discrete states (S1 to S5)** representing SDN behavior under cyberattacks, as detailed in Table 1.

As shown in Table 1, **S1:** Stable Operation – Normal SDN operation without failure. **S2:** Reconnaissance Detected – Network scanning activities are observed. **S3:** Active Attack Execution – SDN is under cyberattack (e.g., DDoS, MITM). **S4:** Controller Compromised – Successful attacker access to the SDN controller. **S5:** Detection and Recovery – Anomalous behavior is detected, and countermeasures are triggered.

This probabilistic model enables quantitative assessment of SDN stability and vulnerability over time. The system transitions between these defined states depending on cyberattack events and recovery mechanisms. A graphical representation of these discrete states and the conditional transition probabilities (event flows, λij) is provided in the CTMC state transition graph shown in Figure 1.

Note that the graph does not consider the transition from state S2 to state S5. In our opinion, the transition from state S2 to state S5 does not have a significant impact on the resilience of the SDN. This is because reconnaissance is a continuous process and does not pose a direct threat to the network's functioning that requires recovery and elimination of the consequences of a successful cyberattack. In other words, when constructing the model, we focused specifically on the ability to counteract cyberattacks, rather than on counterintelligence.

Transition Probabilities and Resilience Calculation

In our opinion, the choice of this number and composition of states is sufficient for the stated research objective, although the possibility of further detailing the states is not excluded. We consider this issue a direction for future research.

Let us define the initial data for the problem:

1. The graph of aggregated stable states of the SDN under cyberattack conditions: G(S, V).
2. The set of states S of the SDN under cyberattack conditions:

$$S = \{S_1, S_2, S_3, S_4, S_5\}$$

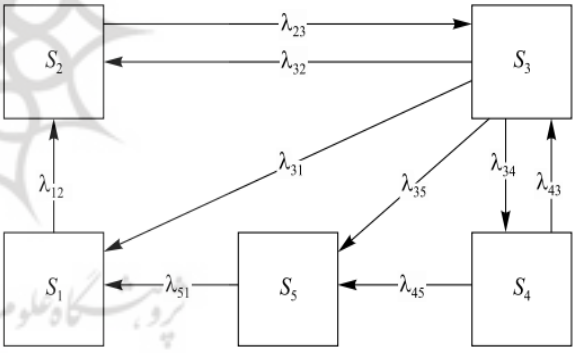

Figure. 1. SDN state and transition graph

Table 1. Description of Conditional Discrete States of a Distributed Corporate SDN Under Cyberattack Conditions

State Designation	***Conditional Discrete State***	***Description of Conditional Discrete State***
S1	Stable SDN operation (normal state)	Stable, resilient operation without failures.
S2	Network scanning by an attacker (reconnaissance phase)	Operation under conditions of technical computer reconnaissance (an intruder gathering information about a future cyberattack target).
S3	Active cyberattack execution	Operation under conditions of conducting cyberattacks against the SDN.
S4	Successful attack (controller compromise)	Operation after a successful attack (successful connection to the attacked network, gaining access to the attacked controller).
S5	Detection and recovery from attack	Detection of anomalies in the network, identification of the cyberattack, and elimination of the consequences of the successful attack.

3. The set of event flows V during changes in the SDN states under cyberattack conditions:

$$V = \{\lambda_{12}, \lambda_{21}, \lambda_{23}, \dots, \lambda_{ij}\}$$

4. Characteristics of stable aggregated states of the SDN under cyberattacks. An example of such a characteristic is the information transit time. It tends towards infinity for state S_1 and during a DDoS attack for state S_4.

5. Values of event flow intensities under cyberattacks, which are obtained as follows: Each considered cyberattack is modeled step-by-step on a simulation computer model built in the EVE-NG virtual environment to obtain temporal characteristics of its stages. Then, using mathematical calculations based on the topological transformation method for stochastic networks [23], the desired values of event intensities are obtained.

6. The vector of initial state probabilities of the system: $p_i(0) = \{1, 0, 0, 0, 0, 0, 0\}$.

7. The normalization condition is given by Equation(1):

$$\sum_{i=0}^{4} p_i(t) = 1 \qquad (1)$$

The moments of probabilistic transitions of the SDN from one state to another, when a protection strategy is employed, are uncertain, random, and occur under the influence of event flows that are characterized by intensities λ_{ij}. These intensities are an important characteristic of the event flows and represent the average number of events occurring per unit time. Numerical values of the intensities will be set in accordance with the simulation model. When solving a system of linear differential equations with constant coefficients (a homogeneous Markov process), we transition to continuous time, t → 0. Based on the labeled graph G, we form a system of differential equations with unknown functions $p_i(t)$, which define the probability of the system being in state S_i. We follow the rule that in the right-hand side of each differential equation for $p_i(t)$, the product $\lambda_{ij}p_j(t)$ is added with a "plus" sign, and the product $\lambda_{ij}p_i(t)$ is subtracted with a "minus" sign. The vector of initial state probabilities of the system, $p_i(0)$, is necessary for the accurate solution of this system(Equation (2)).

$$D(P,T) = \begin{cases} \frac{dp_1(t)}{dt} = \lambda_{51}p_5(t) + \lambda_{31}p_3(t) - \lambda_{12}p_1(t), \\ \frac{dp_2(t)}{dt} = \lambda_{12}p_1(t) + \lambda_{12}p_3(t) - \lambda_{23}p_2(t), \\ \frac{dp_3(t)}{dt} = \lambda_{23}p_2(t) + \lambda_{31}p_1(t) + \lambda_{35}p_5(t) + \lambda_{34}p_4(t) - (\lambda_{23} + \lambda_{34})p_3(t), \\ \frac{dp_4(t)}{dt} = \lambda_{43}p_3(t) + \lambda_{45}p_5(t) - \lambda_{34}p_4(t), \\ \frac{dp_5(t)}{dt} = \lambda_{51}p_1(t) - (\lambda_{35} + \lambda_{45})p_5(t), \\ \sum_{i=0}^{4} p_i(t) = 1. \end{cases} \qquad (2)$$

We define state transition probabilities based on empirical attack scenarios and system defenses. Let P_{ij} represent the probability of transition from state S_i to S_j. The probability of SDN maintaining stable operation over time (resilience metric) is given by Equation(3):

$$p_{\text{res}}(t) = 1 - p_4(t) \qquad (3)$$

where $P_4(t)$ represents the probability of SDN reaching the compromised state (S_4). To solve for $P_4(t)$, we construct a set of linear differential equations based on transition rates (λ_{ij}) and compute steady-state probabilities.

3.2. Phase II: Attack Scenario Modeling and Simulation Setup

We evaluate SDN resilience under five major cyberattack scenarios. These scenarios were carefully selected to reflect real-world threats faced by SDN deployments, and they are directly associated with the system states (S1–S5) in our Markov model. Each attack targets a different aspect of the SDN control or data plane, and their technical mechanisms are outlined below:

DDoS Attack on the Controller (S3 → S4): This attack overwhelms the OpenFlow communication channel between switches and the controller using a high volume of fabricated requests. The controller's processing capacity is exhausted, leading to dropped legitimate packets and degraded control functionality. In our model, this transitions the system from an active attack state (S3) to a compromised state (S4). Overloading OpenFlow channels with excessive requests.

Topology Poisoning (S2 → S3): In this attack, an adversary injects rogue switches or manipulates LLDP messages to create false link advertisements. This disrupts the controller's view of the network topology, resulting in black holes or routing loops. This attack is preceded by reconnaissance (S2) and triggers a move to active interference (S3). Injecting rogue switches into the network.

Malicious Rule Insertion (S3 → S4): Exploiting vulnerabilities in northbound APIs, an attacker can inject unauthorized or malicious flow rules into the SDN. This can reroute traffic to adversarial nodes or create traffic duplication. The transition to S4 indicates successful rule compromise and system behavior alteration.

Man-in-the-Middle (MITM) Attacks (S2 → S3): Through ARP spoofing or DNS poisoning, attackers position themselves between the controller and switches, intercepting or modifying control messages. This is typically preceded by network scanning (S2) and progresses to execution (S3) once control traffic is accessed. Intercepting control messages between the controller and switches.

Controller Hijacking (S3 → S4): An adversary gains unauthorized access to the SDN controller via credential compromise or remote code execution vulnerabilities. This attack leads directly to the controller's takeover and compromises the network's integrity and availability. Gaining unauthorized access to the SDN controller.

To provide clearer linkage between the modeled Markov states and real-world cyberattacks, we present Table 2, which maps each of the five attack types evaluated in this study to the relevant states and transitions within our Markov framework. This mapping ensures that the probabilistic transitions in the model are grounded in realistic attack behaviors and their operational impact on SDN infrastructures.

Experimental Setup in EVE-NG

To empirically validate our Markov model and evaluate the resilience of different SDN configurations under attack, we utilized the EVE-NG (Emulated Virtual Environment - Next Generation) platform [24]. EVE-NG provides a flexible environment for building and simulating complex network scenarios using virtualized network devices.

We deployed and tested three distinct SDN architectures (detailed in Section 4.1.1: Baseline Single-Controller, Redundant Dual-Controller, and Proposed Hybrid AI-enhanced SDN) within this virtual environment. The simulated network was constructed based on a **Tree Topology**. This specific topology consisted of:

- **5 OpenFlow-enabled Switches:** Implemented using Open vSwitch (OvS) instances.
- **14 Virtual End Hosts:** These simulated network clients and servers, generating background traffic and acting as targets or sources for simulated attacks. They were distributed across the leaf switches.
- **SDN Controllers:** The number and configuration of controllers varied per architecture:
  - Baseline Model: 1 controller.
  - Redundant Control Plane Model: 2 controllers (specific handover logic applied).
  - Proposed Hybrid Model: 2 controllers (primary/hot-standby configuration integrated with AI detection).

This **Tree Topology** and the specified scale were selected because they represent a common and fundamental structure found in various network environments, such as campus or small-to-medium enterprise networks. This configuration is sufficiently complex to demonstrate the vulnerabilities associated with centralized control and the benefits of redundancy and intelligent detection, while remaining tractable for controlled experimentation and clear analysis of attack impacts and mitigation effectiveness.

Each of the five cyberattack scenarios was executed on these architectures for 30 minutes, and the resilience probability ($p_{\text{res}}(t)$), as defined in Equation (3), was recorded. This probability indicates the likelihood of the SDN maintaining stable operation during the attack and reflects transitions among the defined Markov states S1 to S5.

3.3. Phase III: Proposed SDN Protection System

To mitigate attacks and enhance SDN resilience, we design an adaptive SDN security system consisting of three core mechanisms:

1. LSTM-Based Anomaly Detection System (for real-time attack detection).
2. Controller Redundancy and Automated Failover Mechanism (for fault tolerance).
3. Software Router Agents (for proactive traffic filtering).

Table 2. Relation between markov states and cyberattacks

Attack Type	*Markov States Involved*	*Typical Transition Sequence*	*Explanation*
DDoS Attack on Controller	S1 → S2 → S3 → S5	Normal → Reconnaissance → Attack → Recovery	Attackers send massive traffic to exhaust controller resources; detected by anomaly system.
Topology Poisoning	S1 → S2 → S3 → S4 → S5	Normal → Scanning → Attack → Compromise → Recovery	Rogue devices injected, mislead the controller; may result in partial control hijack.
Malicious Rule Injection	S1 → S3 → S4 → S5	Normal → Attack → Compromise → Recovery	Exploits northbound APIs to alter flow rules, bypassing detection if not mitigated.
MITM Attack	S1 → S2 → S3 → S5	Normal → Monitoring → Attack → Detection	Eavesdropping or message manipulation between controller and switches.
Controller Hijacking	S1 → S2 → S3 → S4 → S5	Normal → Reconnaissance → Attack → Full Compromise → Recovery	Full unauthorized access to controller through credential theft or software vulnerability exploitation.

LSTM-Based Anomaly Detection

To provide real-time attack detection capabilities, we implement a Long Short-Term Memory (LSTM) neural network, chosen for its proficiency in capturing temporal dependencies within sequential data, which is characteristic of network traffic patterns and state changes.

Dataset Generation and Preprocessing: The dataset for training and evaluating the LSTM model was synthetically generated within our EVE-NG simulation environment. This involved running simulations under both normal operating conditions and the specific attack scenarios outlined in Section 3.2 (DDoS, Topology Poisoning, Malicious Rule Insertion, MITM, Controller Hijacking). This approach ensured the training data directly reflects the behaviors and attack patterns pertinent to the SDN architectures and scenarios investigated in this study.

During simulations, we collected time-series data related to controller and switch activities. Raw data points included:

- OpenFlow message rates (Packet-In, Flow-Mod, Packet-Out per second).
- Flow table entry counts and modification frequency per switch.
- Controller CPU and memory utilization.
- Frequency of topology discovery protocol messages (e.g., LLDP).

This raw data underwent the following preprocessing steps:

1. **Feature Extraction:** We selected and engineered key features indicative of anomalous behavior, such as the *rate of new flow arrivals (Packet-In messages)*, *rate of flow rule modifications (Flow-Mod messages)*, *variance in inter-packet arrival times for control messages*, and *detection of unexpected topology change events*.
2. **Time Series Formulation:** The data was segmented into fixed-length sequences using a sliding window approach.
3. **Normalization:** All feature values within each sequence were normalized using Min-Max scaling to the range [0, 1]. This step is crucial for stabilizing the LSTM training process.
4. **Labeling:** Each generated sequence was labeled as 'Normal' (0) or 'Attack' (1) based on whether a simulated attack was active during the time interval covered by the sequence.

Monitoring: The trained LSTM model continuously monitors the preprocessed feature sequences derived from real-time network state information. Specifically, it focuses on identifying deviations from learned normal patterns in:

- **Flow table updates:** Detecting abnormally high rates or suspicious modifications indicative of malicious rule injections.
- **Control message frequency:** Identifying surges typical of DDoS attacks targeting the controller or unusual communication patterns.
- **Topology changes:** Recognizing unauthorized device insertions or link modifications characteristic of topology poisoning.

If the LSTM model classifies an incoming sequence as anomalous (i.e., predicts 'Attack' with high confidence), it triggers an alert, initiating the automated mitigation mechanisms described later, such as controller failover or traffic filtering via OvS agents.

Controller Redundancy and Automated Failover

A secondary controller remains in hot-standby mode and takes over upon failure of the primary controller. A secure synchronization protocol ensures consistency with minimal downtime. To prevent controller compromise, we deploy a multi-controller system where:

- The primary controller manages network operations.
- A secondary controller is in hot-standby mode.
- If an attack disrupts the primary controller, failover triggers automatic controller switching.

This mechanism is implemented via a synchronization service between controllers.

Software Router Agents for Proactive Filtering

We deploy Open vSwitch (OvS) agents on SDN software routers. These agents:

- Monitor network traffic for anomalies.
- Isolate malicious flows in real-time.
- Trigger failover events if controller compromise is detected.

The OvS agents interact with the LSTM anomaly detection system to enhance security.

3.4. Implementation Flowchart

The flowchart of the proposed methodology for assessing the resilience of a distributed SDN under cyberattack conditions is shown in Figure 2.

The methodology enables the calculation of the resilience level of a corporate SDN when determining. the most relevant attacks for it. The

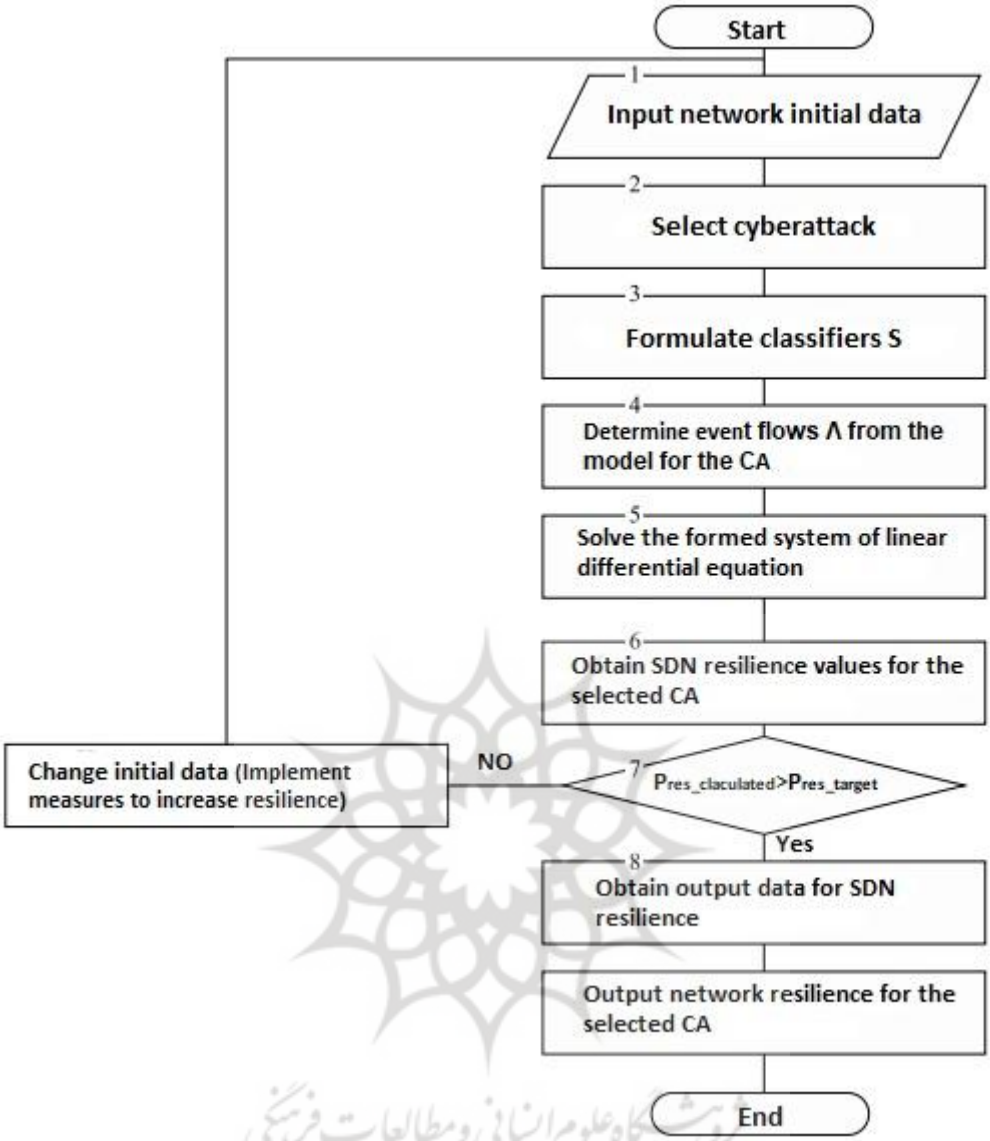

Figure. 2. The flowchart of the proposed methodology

results and the conclusions based on them allow for obtaining an adequate assessment of the SDN's resilience under the modeled conditions against cyberattacks characteristic of that specific network. A step-by-step breakdown of the flowchart as a follow:

1. **Start:** The process begins.
2. **Input Network Initial Data:** The initial parameters and configurations of the SDN are defined. This includes the network topology, device capabilities, routing protocols, security policies, and any other relevant information needed to model the network's behavior.
3. **Select and Simulate Cyberattack:** A specific cyberattack scenario to be assessed is chosen. This could be a DDoS attack, a controller compromise, a routing table poisoning attack, or any other relevant threat. This attack will serve as the basis for the analysis.
4. **Formulate Classifiers S:** Define the different states the system can be in (S1, S2, S3, S4, S5) depending on the chosen cyberattack.
5. **Determine Event Flows Λ from the model for the CA:** Based on the selected cyberattack, you determine event flows and rate parameters (λ) for the transitions in the Markov model of SDN behavior under the considered cyberattack.
6. **Solve the formed system of linear differential equation:** Solve the resulting

system of linear differential equations obtained from the Markov model.

7. **Obtain SDN Resilience Values for the Selected CA:** Using the Markov model, the probability of each state is known, hence the SDN resilience value $p_{\text{res\_Calculted}}$ for the current configuration can be obtained for the selected cyberattack.

8. $\boldsymbol{p_{\text{res\_Calculted}} > p_{\text{res\_Target}}}$**?** This step is a decision point.

- **Yes:** If the calculated resilience value ($\boldsymbol{p_{\text{res\_Calculted}}}$) is greater than the target resilience value ($\boldsymbol{p_{\text{res\_Target}}}$), it means the current network configuration meets the resilience requirements for the selected cyberattack. The process moves to step 9.
- **No:** If the calculated resilience value ($\boldsymbol{p_{\text{res\_Calculted}}}$) is less than the target resilience value ($\boldsymbol{p_{\text{res\_Target}}}$), it means the network does not meet the desired resilience for the selected cyberattack. The process returns to the "Change Initial Data" step.

9. **Change Initial Data (Implement Measures to Increase Resilience):** If the network does not meet the target resilience value, the initial data needs to be modified to improve resilience. This could involve implementing security countermeasures, changing network topology, or improving controller redundancy. After implementing these changes, the process returns to step 1 to re-evaluate the resilience.

10. **Obtain Output Data for SDN Resilience:** If the calculated resilience value meets the target, this step collects the detailed output data regarding the SDN resilience characteristics under the tested conditions.

11. **Output Network Resilience for the Selected CA:** Reports the resulting resilience of the network against the selected cyberattack. The result shows the effectiveness of the current configuration to resist the selected attack.

12. **End:** The process is complete.

4. Experimental Results and Analysis

This section presents the experimental evaluation of SDN resilience under cyberattack conditions. We assess the probability of network stability ($p_{\text{res}}(t)$) using the Markov-based resilience model and validate our results through EVE-NG-based simulations. The performance of our proposed security framework (LSTM-based anomaly detection, automated controller failover, and software router agents) is also analyzed.

4.1. Simulation Environment

SDN Architectures under Test

To assess SDN resilience under cyberattack conditions, three network structures were designed and implemented:

1. An SDN structure consisting of three elements with a single controller. (Baseline Model)
2. An SDN structure based on two controllers with handover of management functions according to a given algorithm under cyberattack conditions. (Redundant Control Plane)
3. An SDN structure with two controllers, where one controller is the primary one and performs the management functions, and the second controller is in hot standby mode. (Proposed Model)

Computational Environment

All simulations were carried out using the EVE-NG Community Edition (v2.0.3-112) hosted on a dedicated virtualized server with the following specifications:

- **Processor:** ntel(R) Core(TM) i5-3210M
- **RAM:** 6 GB DDR3
- **Storage:** 1 TB SSD
- **Virtualization Platform:** VMware ESXi 7.0
- **Guest OS for VMs:** Ubuntu 20.04 LTS

The LSTM module and OvS software agents were deployed in separate containers using Docker and configured as follows:

- **LSTM Container:** 1 vCPUs, 2 GB RAM, optimized using TensorFlow Lite
- **OvS Agent Container:** 1 vCPUs, 1 GB RAM

The EVE-NG network topology included isolated logical bridges to separate the control plane and data plane.

Attack Scenarios

Each cyberattack was simulated for 30 minutes with system metrics collected at 1-minute intervals. These metrics were used to evaluate transition probabilities and calculate resilience values using the Markov-based analytical model. Table 3 shows type of cyberattacks were simulated:

Each attack was conducted for 30 minutes, and network stability was measured at 1-minute intervals.

Table 3. Type of Cyberattacks

Attack Type	*Description*
DDoS Attack on Controller	Overloads OpenFlow channels to exhaust resources.
Topology Poisoning	Injects rogue switches to manipulate routing.
Malicious Rule Injection	Exploits northbound APIs to alter flow rules.
Man-in-the-Middle (MITM) Attack	Intercepts control traffic between controller and switches.
Controller Hijacking	Gains unauthorized access to SDN controller.

4.2. Resilience Probability Analysis

The calculation results are presented as graphs (Figures. 3–5). Expression (3) ($p_{\text{res}}(t) = 1 - p_4(t)$) was used to calculate the SDN resilience metric. A threshold value of 0.2 defines an indicative value, in our view, for the probability of stable network operation. If the resilience metric values are below the threshold, the network ceases to be resilient.

The analysis of the results showed that the considered SDN structures, under the impact of cyberattacks of the "synchronous attack" and "controller breach or failure" types, do not meet the resilience requirements.

To ensure the stable operation of the SDN under cyberattack conditions, it is necessary to develop an algorithm for monitoring the state of the controllers and their automatic reconfiguration, as after 18 minutes of successful cyberattack implementation, the probability of stable network operation begins to approach zero.

Thus, based on the conducted research on the application of SDN, as well as its resilience to cyberattacks, general requirements for a counteraction system have been formulated. The main approach to achieving the required level of SDN resilience could be the development of controller redundancy algorithms, as well as algorithms for redundancy and switching of software-defined switches.

4.3. Effectiveness of Proposed Security Measures

We evaluate the effectiveness of the three core security mechanisms in the proposed model.

LSTM-Based Anomaly Detection Performance

The performance of the LSTM-based anomaly detection system is crucial for the overall effectiveness of the proposed security framework. The model was trained using the synthetically generated dataset described in Section 3.3.1 and evaluated on a separate test set comprising both normal and attack traffic sequences from the EVE-NG simulations.

While overall detection accuracy provides a general indication of performance, metrics such as Precision, Recall, and F1-Score offer a more nuanced assessment, particularly important in security scenarios where the cost of false negatives (missed attacks) and false positives (false alarms) differs.

- **Precision** measures the proportion of correctly identified attacks among all instances flagged as attacks. High precision indicates fewer false alarms.
- **Recall** (or Sensitivity) measures the proportion of actual attacks that were correctly identified. High recall indicates fewer missed attacks.
- **F1-Score** provides the harmonic mean of Precision and Recall, offering a single metric to evaluate the balance between them.

The evaluation yielded the following performance results:

- Overall Detection Accuracy: **98.1%**
- Overall False Positive Rate (FPR): **2.3%** (meaning only 2.3% of normal sequences were incorrectly flagged as attacks)
- Average Detection Time: < **0.5 seconds** (time taken to classify a sequence after observing the full time window)
- Table 4 presents a breakdown of performance metrics across the different attack types simulated.
- These results confirm that the LSTM-based detection system provides accurate and timely attack identification without imposing a significant performance burden on the SDN environment.
- As shown in the Table 4, the LSTM model consistently achieved high Precision (avg. 97.8%), Recall (avg. 98.1%), and F1-Score (avg. 97.9%) across the diverse attack vectors. This demonstrates the model's strong capability not only to accurately detect attacks (high accuracy and recall) but also to do so with minimal false alarms (high precision), leading to a well-balanced performance (high F1-Score). The low detection latency (< 0.5 seconds) further enables timely triggering of mitigation actions. The LSTM model, therefore, effectively detects all evaluated attack types with high confidence and minimal operational disruption due to false positives.

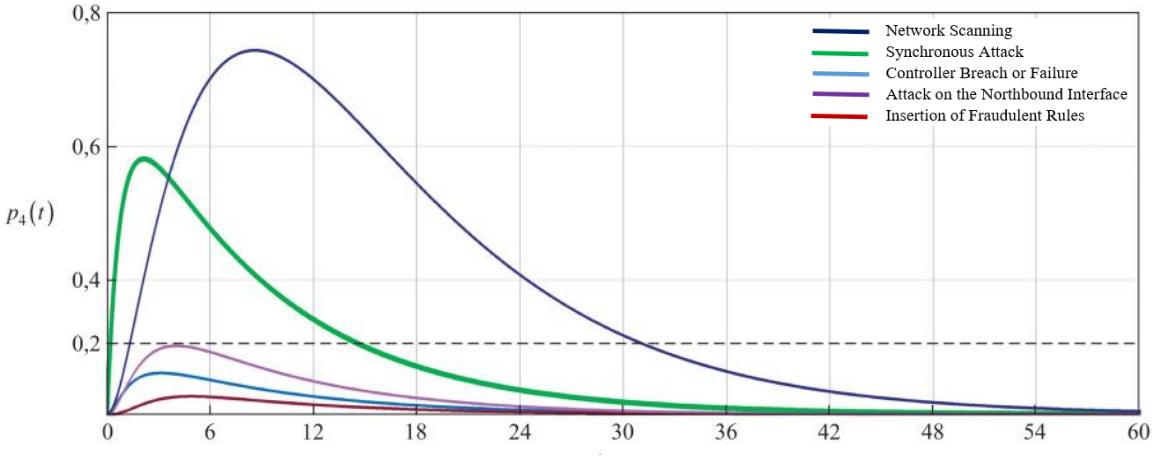

Figure. 3. Dependence of SDN stable operation probability on the time of cyberattack implementation for structure 1.

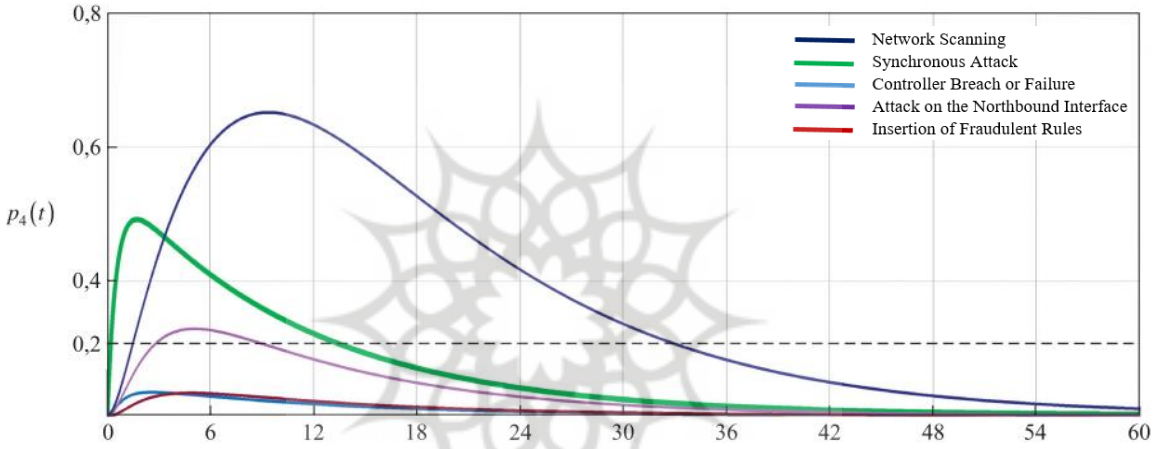

Figure. 4. Dependence of SDN stable operation probability on the time of cyberattack implementation for structure 2.

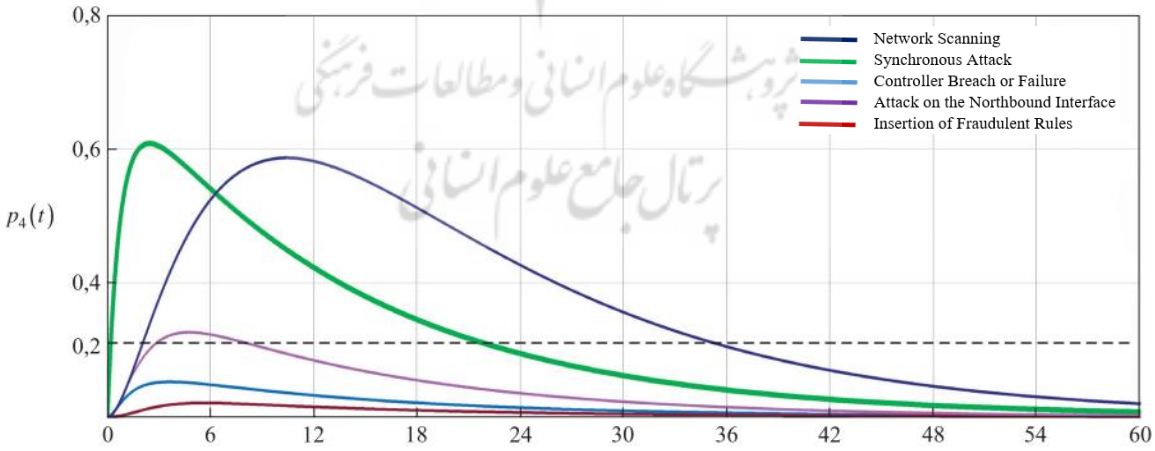

Figure. 5. Dependence of SDN stable operation probability on the time of cyberattack implementation for structure 3.

Controller Failover Response Time

Measure how quickly the backup controller takes over when the primary controller fails. The proposed AI-enhanced SDN cuts failover time in half, reducing downtime significantly. Table 5 shows controller failover response time for three architectures models.

Malicious Traffic Isolation

We evaluated how well software router agents block malicious flows during an attack. The software router agents effectively block malicious flows with negligible performance impact. Traffic efficiency has been resulted in Table 6.

Table 4. LSTM-Based Attack Detection Accuracy and Performance Overhead

Attack Type	*Detection Accuracy (%)*	*False Positive Rate (%)*	*Precision (%)*	*Recall (%)*	*F1-Score (%)*
DDoS Attack	97.5%	2.1%	97.8%	97.5%	97.6%
Topology Poisoning	98.2%	2.5%	97.5%	98.2%	97.8%
Malicious Rule Injection	99.1%	1.9%	98.5%	99.1%	98.8%
MITM Attack	97.8%	2.6%	97.2%	97.8%	97.5%
Controller Hijacking	98.1%	2.3%	97.9%	98.1%	98.0%
Average	98.1%	2.3%	97.8%	98.1%	97.9%

Table 5. Controller Failover Response Time

Architecture	*Failover Time (seconds)*	*Stability Improvement (%)*
Single-Controller SDN	N/A	0% (No redundancy)
Dual-Controller SDN	6.3s	+30%
Hybrid SDN with AI	3.1s	+60%

Table 6. Traffic Filtering Efficiency

Attack Type	*Traffic Blocked (%)*	*Latency Overhead (ms)*
DDoS Attack	94.7%	+1.2ms
Topology Poisoning	96.3%	+1.8ms
Malicious Rule Injection	99.4%	+2.1ms
MITM Attack	97.1%	+1.5ms

4.4. Comparative Analysis

To assess the practical advantages of our proposed model, we conducted a comparative evaluation across three different SDN architectures under identical cyberattack conditions:

1. **Single-Controller SDN (Baseline):** Traditional OpenFlow architecture with no redundancy or intelligent detection.
2. **Dual-Controller SDN (Redundant):** Improved fault tolerance through hot-standby controller.
3. **Hybrid SDN (Proposed Model):** Enhanced with LSTM-based real-time anomaly detection, automated controller failover, and traffic filtering.

Table 7 summarizes key performance metrics including resilience probability after 30 minutes of attack, anomaly detection accuracy, failover response time, and traffic filtering effectiveness.

As shown in Table 7, the hybrid model outperforms both traditional and redundant architectures, particularly in terms of real-time attack detection and response. These findings demonstrate the practical effectiveness of integrating AI-based security mechanisms within SDN environments.

Table 7. Overall Performance Comparison

Metric	**Single-Controller**	**Dual-Controller**	**Hybrid SDN (Proposed)**
Resilience (P_res after 30 min)	18%	47%	82%
Attack Detection Accuracy	N/A	N/A	98.1%
Failover Time	N/A	6.3s	3.1s
Traffic Filtering Efficiency	N/A	N/A	96.7%

We acknowledge that a broader comparison with other state-of-the-art SDN security methods from the literature (e.g., rule-based IDS or CNN/RNN models) would provide additional insight. However, due to space limitations and to maintain the clarity and focus of the current paper, we have chosen to limit our comparative scope to architectural configurations only. A more extensive benchmarking with external models and datasets is planned as part of our future work.

4.5. Deployment Challenges

While the proposed framework demonstrates strong simulation-based performance, practical deployment may face the following challenges:

- **Computational Resource Constraints:** Although optimized using TensorFlow Lite, the LSTM module and OvS filtering agents still introduce computational overhead, which may not be suitable for low-resource environments such as edge or IoT-based SDNs.
- **Model Generalization:** The LSTM model was trained using synthetic data specific to our simulated environment. Its effectiveness in heterogeneous real-world networks with different traffic profiles or unseen attack types may require extensive retraining or adaptive learning.
- **Integration Complexity:** Integrating anomaly detection with existing SDN controllers (e.g., ONOS, Ryu, or OpenDaylight) requires careful engineering, especially when modifying real-time control loops and failover mechanisms.

To address these challenges, future work will explore federated learning for privacy-preserving

model training, lightweight neural architectures for edge deployment, and modular plugins for SDN controller integration.

5. Conclusion & Future Work

Software-Defined Networking (SDN) has transformed network management through centralized control and programmability, but it also introduces security challenges. In this paper, we presented a Markov-based analytical model to assess SDN resilience under cyberattacks and proposed a comprehensive AI-driven protection system. The Markov model provides a robust framework for quantifying SDN resilience under various cyberattacks, revealing that traditional single-controller SDN architectures are highly vulnerable, with resilience dropping to 18% within 30 minutes of a DDoS attack. Our study demonstrates that integrating AI-based anomaly detection and redundant control mechanisms can substantially increase SDN resilience, making it suitable for critical infrastructure and enterprise environments. While the proposed model offers significant resilience improvements, it has some limitations: Scalability Concerns, Limited Attack Scope, and Resource Overheads. To further enhance SDN security, future research could explore: Implementing reinforcement learning for dynamic security policy adaptation based on real-time threat intelligence, Using blockchain technology for secure, decentralized controller coordination, ensuring tamper-proof rule management and trusted communication.

Declarations

Conflict of interest

The author declares that no conflicts of interest exist.

References

[1] Liatifis, A., Sarigiannidis, P., Argyriou, V., & Lagkas, T. (2023). Advancing sdn from openflow to p4: A survey. ACM Computing Surveys, 55(9), 1-37.

[2] Maleh, Y., Qasmaoui, Y., El Gholami, K., Sadqi, Y., & Mounir, S. (2023). A comprehensive survey on SDN security: threats, mitigations, and future directions. Journal of Reliable Intelligent Environments, 9(2), 201-239.

[3] Jimenez, M. B., Fernandez, D., Rivadeneira, J. E., Bellido, L., & Cardenas, A. (2021). A survey of the main security issues and solutions for the SDN architecture. IEEE Access, 9, 122016-122038.

[4] Rahdari, A., Jalili, A., Esnaashari, M., Gheisari, M., Vorobeva, A. A., Fang, Z., ... & Tahaei, H. (2024). Security and Privacy Challenges in SDN-Enabled IoT Systems: Causes, Proposed Solutions, and Future Directions. Computers, Materials & Continua, 80(2).

[5] Kazmi, S. H. A., Qamar, F., Hassan, R., Nisar, K., & Chowdhry, B. S. (2023). Survey on joint paradigm of 5G and SDN emerging mobile technologies: Architecture, security, challenges and research directions. Wireless Personal Communications, 130(4), 2753-2800.

[6] Deb, R., & Roy, S. (2022). A comprehensive survey of vulnerability and information security in SDN. Computer Networks, 206, 108802.

[7] Choudhary, A., Kang, S. S., & Singla, S. (2024, August). Multi-Objective Dijkstra Algorithm for Enhancing QoS in SDN through Balanced Routing. In 2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT) (Vol. 1, pp. 1169-1175). IEEE.

[8] Elzoghbi, M., & He, H. (2024). DIS-Guard: Enhancing SDN resilience to topology and RCO attacks. Computer Networks, 253, 110723.

[9] Asamoah, E. (2023). Genetic Algorithm-Based Improved Availability Approach for Controller Placement in SDN (Doctoral dissertation, Université d'Ottawa/University of Ottawa).

[10] Herdianto, B. (2021, April). Guided Clarke and Wright algorithm to solve large scale of capacitated vehicle routing problem. In 2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA) (pp. 449-453). IEEE.

[11] Zhang, L., Zhu, T., Hussain, F. K., Ye, D., & Zhou, W. (2022). A game-theoretic method for defending against advanced persistent threats in cyber systems. IEEE Transactions on Information Forensics and Security, 18, 1349-1364.

[12] Ma, X., Abdelfattah, W., Luo, D., Innab, N., Shutaywi, M., & Deebani, W. (2024). Non-cooperative game theory with generative adversarial network for effective decision-making in military cyber warfare. Annals of Operations Research, 1-18.

[13] Jalili, A. (2024). Enhancing quality of service in SDNs through Pareto-optimized controller placement using NS-MF algorithm. International Journal of Nonlinear Analysis and Applications.

[14] Jalili, A., Keshtgari, M., Akbari, R., & Javidan, R. (2019). Multi criteria analysis of controller placement problem in software defined networks. Computer Communications, 133, 115-128.

[15] Xiao, J., Pan, X., Liu, J., Wang, J., Zhang, P., & Abualigah, L. (2024). Load balancing strategy for SDN multi-controller clusters based on load prediction. The Journal of Supercomputing, 80(4), 5136-5162.

[16] Yao, H., Qiu, C., Zhao, C., & Shi, L. (2015). A multicontroller load balancing approach in software-defined wireless networks. International Journal of Distributed Sensor Networks, 11(10), 454159.

[17] Abdulghani, A. M., Abdullah, A., Rahiman, A. R., Hamid, N. A. W. A., Akram, B. O., & Raissouli, H. (2025). Navigating the Complexities of Controller Placement in SD-WANs: A Multi-Objective Perspective on Current Trends and Future Challenges. Computer Systems Science & Engineering, 49.

[18] Hassan, S. M., Mohamad, M. M., & Muchtar, F. B. (2024). Advanced intrusion detection in MANETs: A survey of machine learning and optimization techniques for mitigating black/gray hole attacks. IEEE Access.

[19] Afroj, M., Rifat, K. M. S., & Rahman, M. S. (2024, September). Enhanced Detection of DoS/DDoS Attacks in SDN Using Ensemble and Hybrid CNN-LSTM Models. In 2024 IEEE International Conference on Computing, Applications and Systems (COMPAS) (pp. 1-6). IEEE.

[20] Musa, N. S., Mirza, N. M., Rafique, S. H., Abdallah, A. M., & Murugan, T. (2024). machine learning and deep learning techniques for distributed denial of service anomaly detection in software defined networks—current research solutions. IEEE Access, 12, 17982-18011.

[21] Jakaria, A. H. M., Rahman, M. A., & Gokhale, A. (2021). Resiliency-aware deployment of sdn in smart grid scada: A

formal synthesis model. IEEE Transactions on Network and Service Management, 18(2), 1430-1444.

[22] Einy, S., Oz, C., & Navaei, Y. D. (2021). The anomaly-and signature-based IDS for network security using hybrid inference systems. Mathematical Problems in Engineering, 2021(1), 6639714.

[23] Samad, A. (2023). Hybrid Approaches in Threat Detection: Integrating Traditional Signature-Based Methods with AI and ML Techniques for Enhanced Accuracy.

[24] https://www.eve-ng.net/

[25] Kobayashi, H., Mark, B. L., & Turin, W. (2011). Probability, random processes, and statistical analysis: applications to communications, signal processing, queueing theory and mathematical finance. Cambridge University Press.

[26] Zhu, J., Wang, L., Han, X., Liu, A., & Xie, T. (2024). Safety and Performance, Why Not Both? Bi-Objective Optimized Model Compression against Heterogeneous Attacks Toward AI Software Deployment. IEEE Transactions on Software Engineering.

Dr. Ahmad Jalili is a faculty member of Gonbad Kavous University and a Ph.D. candidate in Computer Networks at Shiraz University of Technology. His research interests include Software Defined Networking (SDN), Wireless Sensor Networks (WSNs), Fiber Optic Networks, Named Data Networking (NDN), and heuristic algorithms. He has published extensively in international journals and conferences, particularly in the areas of WSNs and SDNs. His current research focuses on emerging trends in SDN and network modeling.