INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 11, Issue 1 - V11I1-1217)
Available online at: https://www.ijariit.com

Anomaly Detection in SDN-enabled Cloud-Fog Collaborative Networks

Ibraheem Kateeb
i.kateeb@qu.edu.sa
Qassim University, Saudi Arabia

Yasser Ahmed
hsanien@qu.edu.sa
Qassim University, Saudi Arabia

ABSTRACT

Software-defined networking (SDN) has brought about a paradigm shift in network management and control, offering increased flexibility and automation capabilities. This transformation is particularly relevant in the context of smart cities, where integrating IoT devices and smartphones is essential for delivering efficient and responsive city services. As SDN gains prominence, the security of these devices in fog computing environments becomes a critical concern for maintaining the integrity and reliability of smart city infrastructures. Effective access control mechanisms are needed to safeguard the network, tailored to the unique requirements of SDN networks. These mechanisms are crucial for smart cities, where many devices and systems need to interact seamlessly while maintaining high-security standards. Additionally, KPI anomaly detection in SDN networks poses challenges due to the real-time processing of copious data. To address these challenges, this paper proposes a cloud-fog collaborative architecture with a GAN-GRU model for malicious activity detection. This architecture is designed with smart cities in mind, where fog nodes divide the network into subregions, mirroring the geographical divisions of a city. The cloud is responsible for model training, while fog nodes execute detection tasks, ensuring a responsive and efficient system for anomaly detection. The proposed method outperforms benchmark algorithms in terms of precision, recall, and F1 score, demonstrating its potential for implementation in smart city infrastructures. Furthermore, the impact of time window length on anomaly detection is analyzed, revealing optimal performance with a window length of 70. This paper also introduces a reputation-based access restriction management mechanism, demonstrating its effectiveness in preventing unauthorized access while ensuring secure operations.

Keywords: *Software Defined Networks (SDN), Fog computing, Anomaly Detection, Generative Adversarial Networks (GANs), Gated Recurrent Unit (GRU), Wireless Networks, Smart Cities.*

1. INTRODUCTION

SDN (Software-defined networking) has ushered in a new era for network oversight and operations. Its adaptability, programmable nature, and enhanced automation make it a forward-looking approach to internet evolution [1]. This shift holds significant importance, especially in urban environments aiming to become smart cities, where integrating devices like IoT and smartphones is crucial for streamlined and prompt urban services. A standout feature of SDN is its capability to distinguish between control and data functionalities, paving the way for centralized network governance. This centralization offers administrators an amplified view and command over the network's flow, facilitating dynamic adjustments in line with evolving requirements [2]. However, with the rise of fog computing, ensuring the security of devices like IoT and smartphones is becoming increasingly challenging [3]. Given their constrained memory and computational capacities, updating them with the latest security patches becomes a task, exposing them to potential threats [17].

When malicious devices connect to a network, they can launch a variety of attacks, including DoS attacks and malware infections, which can cause significant damage. To prevent unauthorized access and maintain the integrity of the network and its resources, SDN

networks need effective access control mechanisms, particularly in environments with separate control and data planes [4]. However, managing access to devices in SDN networks can be challenging, as the current access control methods may not be sufficient [5]. As a result, it is crucial to develop appropriate access control strategies that are tailored to the unique requirements of SDN networks, especially in the context of smart cities where the security of interconnected devices is paramount [18].

If an authorized device is unlawfully commandeered and exploited, the entire network may sustain extensive harm if timely protective measures are not taken. Detecting malicious activities in Key Performance Indicators (KPI) can assist in preventing intrusions. By monitoring devices' performance, operators can identify abnormal situations that may indicate security threats or intrusion attempts. However, KPI anomaly detection in SDN networks presents a significant challenge due to the copious amounts of data generated by network devices, which must be processed in real time for effective network management. To overcome this challenge, several studies have explored the use of AI-based techniques in SDN networks such as clustering algorithms and Deep Learning based methods to detect KPI anomalies. While advanced AI-based techniques such as Deep Neural Networks have superior big data processing and analysis capabilities compared to other machine learning methods, they require substantial computation to achieve high accuracy. Since IoT devices require real-time responses to detect malicious activities, neither cloud computing nor fog computing alone can fulfill this requirement.

Our proposed method for malicious activity detection combines a cloud-fog collaborative architecture with deep learning, utilizing the GAN-GRU model. Given the significant number of malicious activity detection tasks, fog nodes divide the network into subregions, with each subregion assigned a set of detection tasks performed by the cloud. Collaborative intelligence between the cloud and fog is leveraged to complete the detection of KPI-based anomalies. The cloud is responsible for model training, while fog nodes execute detection tasks, ensuring the safety of operational data and minimizing data transmission delays. This shared computing strategy helps alleviate the computational burden on the cloud.

The main contributions of this work are as follows:

Contribution 1: Cloud-Fog Collaborative Architecture with GAN-GRU Model
The paper introduces a novel cloud-fog collaborative architecture for anomaly detection in SDN-enabled networks for smart cities. By combining the computational capabilities of the cloud with the low-latency processing of fog nodes, the proposed architecture achieves efficient and real-time KPI anomaly detection. The architecture utilizes the GAN-GRU model, which leverages the power of deep learning and adversarial learning to effectively capture complex patterns in large-scale network data.

Contribution 2: Subregion Partitioning and Distributed Detection Tasks
To handle the copious amounts of real-time data generated by network devices, the proposed method divides the network into subregions. Each subregion is assigned a set of detection tasks, which are performed collaboratively between the cloud and fog nodes. The cloud is responsible for model training, maintaining a global view of network behavior, and updating the detection model. On the other hand, fog nodes execute the detection tasks locally on the data collected from devices within their respective subregions of smart cities. This distributed approach not only ensures efficient processing but also minimizes data transmission delays.

Contribution 3: Real-time Malicious Activity Detection for IoT Devices
The paper addresses the specific security concerns associated with IoT devices in fog computing environments.

For smart cities. Given the limited memory and processing capabilities of IoT devices, traditional security mechanisms may not be sufficient to protect them from known security threats. The proposed method provides a real-time detection mechanism for identifying and preventing malicious activities in IoT devices. By combining the cloud-fog collaborative architecture with the GAN-GRU model, the method offers a timely response to detect security threats.

The paper is organized as follows: In Section, an extensive review of related work is presented. The section provides an overview of the system architecture and in-depth information about the proposed scheme. The detection method of KPI-based anomaly detection is described in Section. The section delves into the discussion and evaluation of the results obtained from the proposed system. Finally, the Section concludes the paper.

2. KPI ANOMALY DETECTION IN CLOUD-FOG COLLABORATION USING GAN-GRU

2.1 System Architecture

In Figure 1, we can see the architecture of the proposed work that detects anomalies and provides an access control mechanism. The system is designed for SDN-based cloud-fog collaboration, which comprises three layers: the devices layer, the SDN-fog layer, and the cloud layer. Both devices and fog nodes are strategically deployed within the same region, enabling real-time supervising of different devices and data collection for detecting anomalies.
To achieve access control, ABAC model-based RestFul services are provided by the SDN controller. Subsequently, the fog nodes implement access control using the results obtained from the supervised device's anomaly detection process. This way, the system ensures secure and authorized access to the supervised devices based on their behavior. For efficient KPI anomaly detection, an

intelligent collaboration between the centralized cloud and fog nodes is established. The training of the model is carried out in the cloud, and subsequently, the trained model parameters are transferred to the fog nodes for effective detection of anomalies in the local data. This collaborative approach ensures a comprehensive and decentralized KPI anomaly detection mechanism.

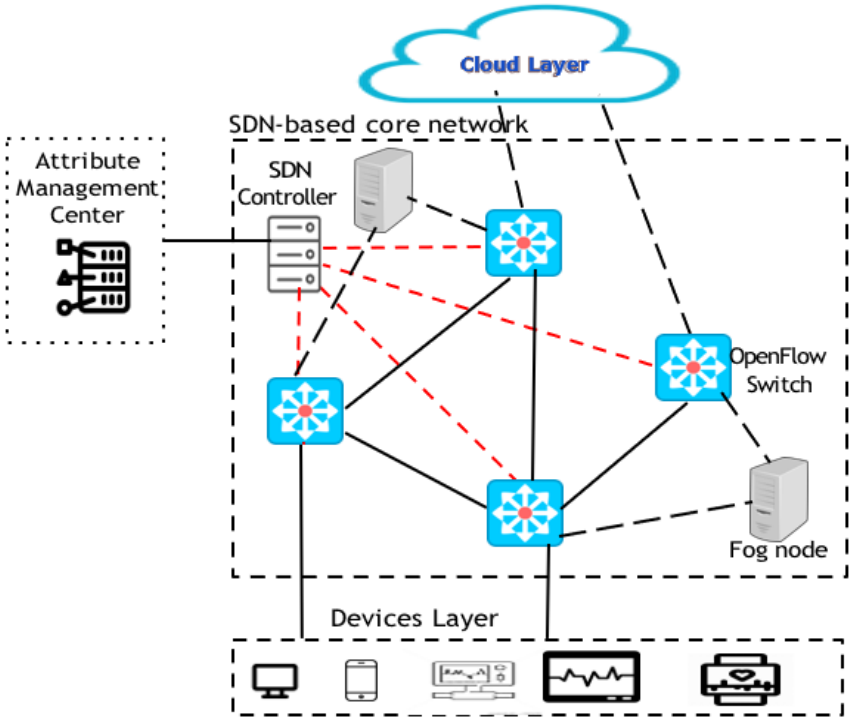


Figure 1. System Architecture

2.2 Supervised devices

The supervised device inputs the necessary attribute data for access using the *Options* section of the IP packet and forwards it to the OpenFlow Switch. These switches being programmed using P4 language (Programming Protocol-Independent Packet Processors), then process the packet according to the defined rules and protocols, which may involve granting or denying access to the SDN network based on the attributes provided in the packet's Options field. Additionally, the KPI data generated in real time is gathered and transferred to the fog nodes for subsequent data processing and evaluation.

2.2.1 Fog node

In the system's architecture, the fog node acts as a pivotal component, enhancing and broadening the functionalities of cloud computing right to the data's origin. It gathers data flows from supervised devices and provides immediate computational services, including access management, data retention, and anomaly identification. After retrieving the model parameters from the cloud, the fog node can independently conduct anomaly detection. To ascertain access permissions, the fog node captures IP packets from devices, extracting details from the Options segment to formulate the device's access request. The fog node then liaises with the SDN controller via RestFul protocols to confirm if the device is authorized to join the network. Furthermore, the fog nodes keep a reputation score for supervised devices. Upon detecting any irregularities, the related device's reputation score is diminished. If the reputation score drops beneath a set threshold, the device's network connectivity is curtailed. This reputation-based system improves security and prevents unauthorized network entries.

2.2.2 Centralized Cloud

The primary function of the cloud is to harness its vast computational resources to manage tasks that demand significant computational power. Given the detection of KPI-based anomalies and the model's dependence on Deep Learning-based algorithms, the cloud employs its formidable computational strength to undertake model training activities, subsequently crafting a comprehensive collection of anomaly detection models suited for diverse classifications. By formulating these models, the cloud can dispense pertinent model parameters to various fog nodes, considering the distinct KPI sequences they oversee. This collaboration empowers the fog nodes to adeptly execute anomaly detection tasks, enabled by the cloud's computational resources.

2.3 Attribute Management Center (AMC)

In the AMC, an access control model based on Attribute-Based Access Control (ABAC) is implemented, which the SDN controller accesses. The AMC ensures secure and flexible access management.

2.3.1 SDN Controller and Switches

In the system architecture, the SDN controller plays a vital role in separating the control plane from the data plane while overseeing the attribute set of connected devices in AMC.

For data packet processing, our SDN switches include both P4 and OpenFlow switches. P4 switches handle quick data packets' pre-processing by checking the validity of the carried Token and granting rapid access based on the query result. Additionally, P4 parses the IHL field of IP packets, filtering those without a valid token. On the other hand, OpenFlow switches encapsulate the received data packets into *Packet in* messages, sending them to the controller for further processing. The controller then issues flow table instructions to the OpenFlow switches, enabling them to perform normal packet forwarding according to the predefined rules and policies.

2.4 KPI Anomaly Detection

The collaboration network architecture of the cloud-fog divides the network into various small regions. In this configuration, the detection task of KPI-based anomalies includes the training phase of the model that occurs in the cloud, while anomaly detection itself is performed by each fog node. To maintain data security and leverage the computational capabilities of fog nodes, data generated by each sub-region is restricted to flowing between the device and the fog node. After completing anomaly detection model training, the cloud transmits encrypted model parameters to the fog nodes. This work considers a scenario involving the cloud, a fog node, and a supervised device to streamline the anomaly detection process. This approach optimizes resource utilization while preserving data privacy and enables effective KPI anomaly detection in cloud-fog collaboration using SDN-based networks.

As shown in Figure 2, during the training stage, the supervised device collects KPI data and sends it to the fog node, which forwards it to the cloud. It trains the model to create a robust KPI anomaly detection model. During the detection phase, the fog node retrieves model parameters from the cloud, finalizes the model initialization, and conducts instantaneous KPI anomaly identification on the data sourced from the supervised device. This approach ensures efficient utilization of fog node resources while maintaining data privacy and security. Additionally, model training in the cloud optimizes detection accuracy, leading to a more reliable KPI anomaly detection system.

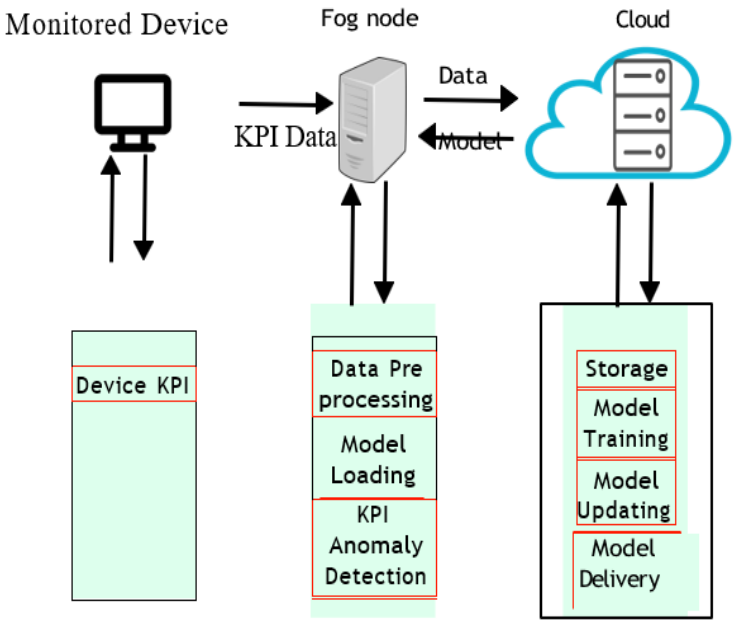


Figure 2. Cloud-fog collaboration to detect KPI-based Anomalies in SDN

2.5 Data Pre-processing

To optimize the performance of the KPI-based anomaly detection model, it is imperative to pre-process the gathered data. This pre-processing step is essential due to occasional data missing and the presence of redundant data during the data collection phase. To address redundant data, any occurrence of the same KPI value at the same timestamp is deleted to guarantee a one-to-one correspondence between the KPI value and timestamp. When dealing with missing data, since KPIs represent *time-series* data, to fill in the missing values this this work utilizes *linear interpolation*. Therefore, preserving the time series characteristics of KPIs. To estimate the missing value in the time interval.
$[t_0, t_1]$ between two known KPI values v_0 and v_1 at the time points t_0 and t_1, a linear equation using the linear interpolation method can be drawn. Mathematically,

$$v_m = v_0 + \frac{(t_m - t_0)}{(t - t)}(v_1 - v_0), \quad (1)$$

1 0

Where v_m is the estimated missing KPI value at time point t_m, and t_m is the time point for the missing KPI value, which falls within the interval $[t_0, t_1]$. Furthermore, considering the variety of gathered sequences that possess varying magnitudes and lengths, it is crucial to standardize the KPI-based data to ensure the effectiveness of the detection model. Therefore, in this work, the Z-score method for KPI-based data standardization is used. The Z-score equation is presented as follows:

$$z = xt - \mu/\sigma \quad (2)$$

Where z represents the standardized KPI value (Z-score), x is the original KPI value at time t. The mean and standard deviation of the KPI sequence are denoted with μ and σ.

2.6 Constructing the GAN-GRU Model

The GAN is adept at generating approximations of real data samples, while the GRU excels at capturing patterns within time series data. Consequently, this work considers a GAN-GRU fusion model for the detection of KPI-based anomalies. This section, will provide details of the GAN-GRU model and its associated network architecture. GAN architectures consist of two models: generator and discriminator. After preprocessing, the data is broken down. Into several time segments. These segments, along with synthetic samples produced by the generator model using noise vectors, are input into the discriminator model. This model assesses the authenticity of the input samples and provides feedback. Both models adjust their parameters using the backpropagation technique.

Due to their streamlined structure and reduced parameter count, GRU networks demonstrate superior performance when handling sequential data. By employing GRU networks, which serve as both the generator and discriminator, within a GAN framework for time-series KPI-based data, a more accurate representation of the features and relationships can be achieved. As a result, the quality of the synthetically produced samples is improved. A visual depiction of the model's structure can be seen in Figure 3.

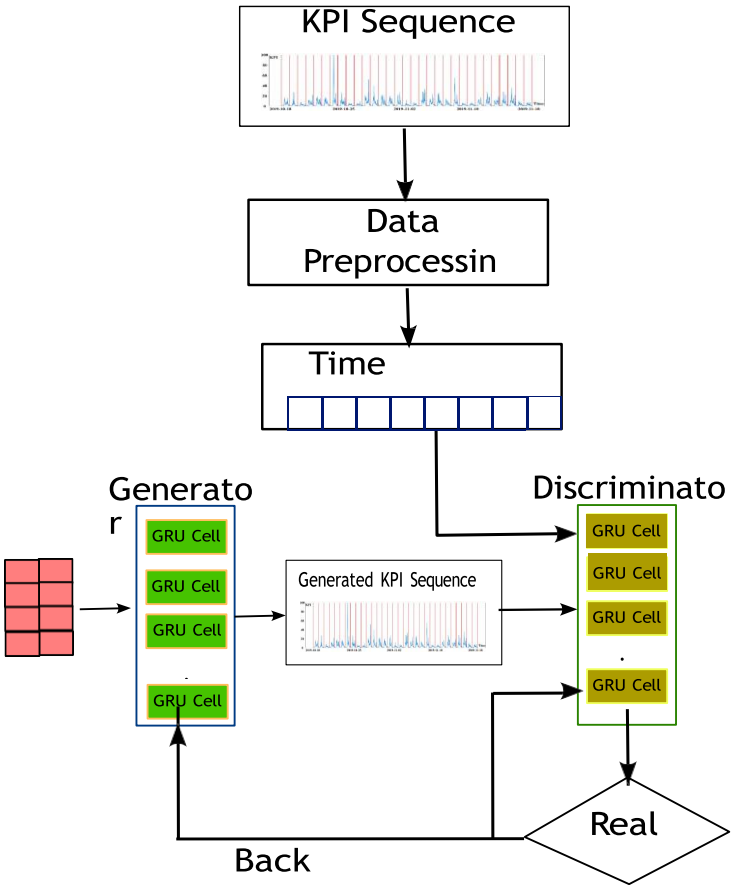


Figure 3. The GAN-GRU Model Architecture

The architecture of the generator model comprises an input layer, four layers of GRU units within the GRU layer, followed by a densely connected layer that employs the Tanh as its activation function. Mathematically,

$$\tanh x = \frac{e^{2x} - 1}{e^{2x} + 1}$$

discriminator model's structure consists of (1) an input layer, (2) a GRU layer, and (3) a densely connected layer. On the other hand, the discriminator model's GRU layer consists of just two GRU unit tiers. The primary function of the discriminator model is to act as a

binary classifier, determining if the input is an authentic real-world sample or a synthetic one produced by the generator model. Moreover, the discriminator model employs the Sigmoid activation function, which outputs the likelihood of samples being real or synthetic. The Sigmoid function's output spectrum is [0,1], making it suitable for binary classification tasks. The function can be represented as

$$Sigmoid\ (x) = 1/\ (1 + \exp(-x)). \quad (4)$$

2.7 GAN-GRU-Based KPI Anomaly Detection Model

In the process of detecting anomalies in KPIs using the GAN-GRU approach, the model undergoes training on typical sequences. The synthetic examples are crafted by the generator model, leveraging a stochastic noise vector from the latent domain and alternating its training with the discriminator model to refine its parameters. As the rivalry between the two GAN models escalates, the adeptness of both is enhanced. The distribution of the crafted samples progressively aligns with the distribution of real samples, making it intricate for the discriminator model to distinguish between them. When the likelihood of the outputs being real or synthetic nears 0.5, the training phase is deemed finalized. In the phase of anomaly identification, the abnormal score for each time segment is deduced using the reconstruction discrepancy of the generator and the loss of the discriminator. Moreover, based on the set of abnormal scores, the detection threshold is set, and anomalies are detected by comparing the abnormal score with the threshold.

2.7.1 The Training of Model

GANs involve optimizing a shared objective function while training the generator and discriminator models, turning the optimization process into a minimax two-player game problem. Therefore, the objective function aims to optimize both models to bring the distribution of generated sample data, Pz, as close as possible to the distribution of real sample data, $Pdata$. The corresponding equation can be expressed as:

$$\mathrm{Min_G\ Max_D}\ V\ (D, G) = \mathrm{E}x \sim Pdata(x)\ [\log D(x)] + \mathrm{E}z \sim Pz(z)\ [\log\ (1 - D(G(z)))], \quad (5)$$

Where $V\ (D,\ G)$ is the value function to be optimized. The discriminator and generator models are represented as D and G, respectively. E denotes the expected value, x is a sample from the real data distribution $Pdata\ (x)$, and z is a sample from the generated data distribution $Pz\ (z)$.

i. The goal of the generator model is to reduce the loss of the objective function, whereas the discriminator model strives to maximize it. The training steps for the combined GAN-GRU model are as follows:
ii. Define the total number of training iterations as n. Assign the label 1 to real samples and the label 0 to synthetic samples.
iii. Obtain a real KPI sequence $x1, x2, x3, ..., xi$. For a synthetic sample z crafted from random noise vectors
iv. $z1, z2, z3, ..., zi$ via the generator $G(z)$, assign the label 0. For the noise vector transformed through $G(z)$, assign the label 1.
v. While keeping the generator constant, optimize the discriminator. Initially, introduce the real sample x to the discriminator, determine the loss from its output and label 1, apply backpropagation, and modify the network parameters. Subsequently, introduce the synthetic sample $G(z)$, determine the loss from its output and the label 0, apply backpropagation, and modify the network parameters. The accumulated loss from both real and synthetic samples represents the total loss for the discriminator model, which is iteratively refined.While keeping the discriminator constant, optimize the generator. Introduce the random noise z to the generator to produce the synthetic sample $\mathrm{G(z)}$. The generator adjusts its parameters based on the feedback from the discriminator for synthetic data, striving to reduce log $(1 - D(G(z)))$ and incrementally understand the distribution of real sample data.
vi. Repeat steps (3) and (4) until you reach the training cycle n.
vii. Upon completing n training iterations, both the generator and discriminator models stabilize, indicating the end of the training process.

2.7.2 Phase of Detecting Anomalies

In this phase, the evaluation of the KPI-based data is carried out, which is segmented into multiple temporal windows and then processed by the pre-trained GAN-GRU model. The variance between the crafted KPI sequence and the real input KPI sequence, as produced by the generator model, encompasses both the reconstruction discrepancy.
And discernment loss. This results in an anomaly score that forms the foundation for detecting KPI anomalies. A window is deemed abnormal if its anomaly score surpasses the detection threshold.

Reconstruction Error: In the t-the time window, represented as $xt = \{xt1, xt2, xt3, . . ., xtl\}$ with l indicating the window length, the GAN-GRU model generates a sample denoted as $G(zt) = \{G(zt1), G(zt2), G(zt3), . . . , G(ztl)\}$. The reconstruction error, ϕr, evaluates the dissimilarity between the actual sample xt at every time point within the
t-the time window and the generated sample $G(zt)$. The reconstruction error is calculated as:

$$\phi r\ (x) = \Sigma^{l}\ |xti - G(zti)\ | \quad (6)$$

i=1

Discriminator Loss: The loss of the discriminator, denoted as ϕd, evaluates the performance deficit of the discriminator model. Meanwhile, f (·) indicates the result from the discriminator model's intermediary layer. The formulation for the discriminator's loss is given by:

$$\phi_d (x) = L_D (f (x)) \quad (7)$$

Where L_D is the loss function for the discriminator model, and f (x) is the output of the intermediate layer of the discriminator model.

Anomaly Detection: In the process of identifying anomalies, each temporal window's anomaly score, symbolized as the anomaly score$t$ is computed as a weighted combination of the reconstruction error ϕ_r and the discriminator loss ϕ_d. This metric gauges the variation between the crafted KPI sequence and the real input KPI sequence and can be represented as:

$$\text{anomaly score } t = \lambda\phi_r + (1 - \lambda)\, \phi d, \quad (8)$$

Where the parameter $\lambda \in (0, 1)$ adjusts the relative significance of ϕr and ϕ_d. After numerous experiments, it was determined that the optimal value for λ is 0.1, which yields the best performance in anomaly detection.

The set of anomaly scores for all time windows is represented by the set anomaly score. The detection threshold is determined using the mean, denoted as μ, and the standard deviation, represented by σ, of the anomaly score. The detection threshold can be defined as follows:

$$\text{threshold} = \mu + k\, \sigma, \quad (9)$$

Where k is a constant factor that determines the sensitivity of the threshold. Moreover, the current time window' anomaly score is computed and assessed against the threshold to classify whether it is normal or abnormal. In case the anomaly score surpasses the threshold, the current time window is labeled as abnormal, while it is classified as normal otherwise.

2.8 Model Updates

For the best efficiency and dependability of the detection of KPI-based anomalies, and to promptly identify anomalies in recent sequences, this work introduces a cloud-fog collaborative approach for updating models. The proposed approach consists of two methods:

(1) Updating the Timed Model: In this method, the model is updated periodically by defining an update cycle T'. Let N be the total number of update cycles, the update process can be mathematically represented as:

$$\begin{aligned} U_i &= \text{Update model with KPI data if i=0 (mod T)} \\ &\quad \text{No update otherwise} \end{aligned} \quad (10)$$

Where U_i is the update process for the i-th cycle ($1 \leq i \leq N$).

Upon reaching the designated update interval, the fog node compiles the data produced during that period and forwards it to the cloud. It then refines the detection model of KPI-based anomalies by integrating this updated KPI data, resulting in updated model weight parameters. These revised parameters are relayed back to the fog node and integrated into the latest anomaly detection model. This periodic process of updates ensures the resilience of the detection model of KPI-based anomalies and augments the precision of the anomaly detection procedure.

(2) Library Update for Models: This study introduces a library dedicated to KPI anomaly detection models, designed to identify a variety of KPI sequences. Let's denote the total count of KPI types in the library as M. The models within this library are equipped to identify only the known KPIs. When a novel KPI emerges, the corresponding model for this new KPI isn't present in the library, necessitating periodic library updates. As the supervised device gathers data related to the new KPI, this data is relayed to the fog node. This node then encrypts the data and forwards it to the cloud. Here, a detection model tailored for the new KPI sequence is trained. Subsequently, this new model is sent back to the fog node, culminating in the library's update.

This work assumes M_{new} be the number of new KPI types detected. The updated model library can be represented as $M_{updated} = M + M_{new}$. Therefore, the cloud-fog collaborative model update strategy enhances the performance and accuracy of the detection model of KPI-based anomalies, while also ensuring that newly collected KPI data can be quickly and accurately detected.

3 THE ACCESS CONTROL PROCESS

This section delves into the intricacies of the access control process, which encompasses two pivotal elements: the access control strategy grounded on the Attribute-Based Access Control (ABAC) paradigm, and the access restriction technique rooted in anomaly detection and reputation metrics.

3.1 Access Control Mechanism based on ABAC Model

The ABAC model offers a flexible and dynamic approach to access control by defining access rules based on the attributes of the user, resource, environment, and action. The main components of the ABAC model are:

Attributes: Characteristics or properties that define users, resources, actions, and environments. Examples of attributes include role, department, location, and resource type.
Access Control Policy: A set of rules that dictate the conditions under which access is granted or denied. These rules are defined based on the attributes of the entities involved in an access request.
Policy Enforcement Point (PEP): The component responsible for intercepting access requests and enforcing the access control policies.
Policy Decision Point (PDP): The component that evaluates the access request against the access control policies and returns a decision to the PEP.

The access control process based on the ABAC model follows a series of steps to determine whether a user should be granted access to a specific resource. Those steps are explained as follows.
Device Registration: Before a device can access the network, it needs to submit its attributes to the AMC. This step ensures that the device is registered within the system and its attributes are recorded for access control purposes.
Access Request Initiation: Similar to user access requests, the device initiates an access request to perform a specific action on a resource within the network.
Policy Enforcement Point (PEP): The PEP intercepts the device's access request and gathers the relevant attributes associated with the device, resource, action, and environment.
Policy Decision Point (PDP): The PEP forwards the access request along with the device attributes to the PDP for evaluation against the access control policies.
Policy Evaluation: The PDP evaluates the access request, considering both user and device attributes, against the access control policies defined by the organization.
Decision Response: Based on the evaluation, the PDP sends a decision (grant or deny) back to the PEP.
Device Authentication: Upon deciding to allow access, the device authentication procedure begins. The device incorporates its attribute set into the IP packet's Options section and dispatches the packet to the system.
Authentication Verification: The packet is processed by network components such as P4 and OVS. The fog node parses the packet, extracts the device attributes, and generates a device authentication request (AAR) based on these attributes.
Access Verification: The AAR's compliance with the established access guidelines is checked. If the AAR aligns with the guidelines, a 1 status code is sent from the controller to the fog node. If not, a 0 is dispatched.
Access Enforcement: When the fog node gets a 1 response code, it dispatches a flow directive to OVS, granting the device network access. The fog node then transmits an authentication token to the device. This token, along with its associated DeviceID, is preserved in the fog node's temporary storage, ensuring they remain in sync.
Access Authorization: The device incorporates the authentication token into the Options segment of its IP packet and dispatches it back to the network. This packet undergoes processing by P4, which then cross-references the fog node's temporary storage for the token's details. Should the token be present and remain active, the P4 management layer dispatches a flow directive to OVS, authorizing the device's network entry. If not, the packet is rerouted to the fog node.
Access Control Decision: The fog node evaluates the packet with invalid or missing tokens and makes an access control decision based on the defined policies. The device is either allowed or denied access to the network accordingly.

3.2 Mechanism for Access Restriction via Anomaly Detection and Reputation

This work, it is proposed an access restriction mechanism that combines the detection of KPI-based anomalies and reputation value assessment to enhance the security of the access control process. This mechanism focuses on identifying suspicious access patterns and preventing unauthorized access. Anomaly detection involves analyzing access patterns and detecting deviations from normal behavior, while reputation value assessment evaluates the trustworthiness of users or devices based on their past behavior. In contrast, the ABAC mechanism makes access decisions based on user, resource, and contextual attributes, providing flexibility in dynamically authorizing access rights.
The access restriction mechanism operates by continuously monitoring and collecting user access behavior data. This data is then analyzed using anomaly detection techniques, such as machine learning algorithms, to identify abnormal access patterns. Based on the historical access behavior and detected anomalies, users are assigned a reputation value. This reputation value becomes an attribute in the ABAC model and influences the evaluation of access control policies. If a user's reputation value falls below a predefined threshold, their access privileges may be restricted or revoked, depending on the severity of the detected anomalies. Regular reassessment of reputation values ensures that users demonstrating consistently appropriate access behavior can regain their access privileges over time, thereby promoting a more secure access control process. By integrating the ABAC model, the access control process can dynamically adapt to evolving security requirements.

3.3 Device Reputation-Based Management Mechanism

This section outlines the design of a device management mechanism that utilizes reputation values to ensure reliable and secure operations. Before delving into the detailed design, this work focuses solely on the outcomes of the detection of KPI-based anomalies in devices, disregarding their behavior, such as malfunctioning or potential invasion. As a result, the reputation R_i of device D_i is determined based on the results of KPI anomaly detection and can be expressed as a combination of two components.

$$R_i = \rho_0 + \rho_i^+ + \rho_i^- \quad (11)$$

Where, the reputation score ρ_i^+ is attributed to devices that exhibit normal KPI outcomes, whereas ρ_i^- is linked to Devices that show abnormal KPI outcomes. The initial reputation score, symbolized by ρ_0, represents the starting point for the reputation metric, which can be deduced as follows:

$$\rho_0 = (1/N) \sum_{i=1}^{N} Ri \quad (12)$$

Where N represents the total number of devices connected to the SDN network. The reputation score R_i reflects the trustworthiness and reliabi performance. By considering both positive and negative reputation scores, the overall reputation Ri captures the device's overall standing in the $\rho+$, in Eq. 11 is defined as:

$$\rho_i+ = \alpha \cdot n_i+ \quad (13)$$

Where n^+ signifies the consecutive instances where device D_i is identified as operating within normal parameters. If D_i consistently shows normal behavior over a specific duration, and its reputation score, ρ_i^+, sees a rapid increment. Conversely, if an anomaly is detected in D_i's KPI, n^+ resets to its initial state. Consequently, the reputation metric ρ_i^+ Also reverts to its initial value until D_i returns to normal operation. The weight coefficient α denotes the emphasis placed on ρ^+ during the reputation score computation.

Moreover, considering the set of Key Performance Indicators (KPIs) that device D_i needs to analyze for anomaly detection as $\{k\ pi_1, k\ pi_2, \ldots, k\ pi_m\}$, the reputation score can be determined based on the performance of D_i across these KPIs. Therefore, the negative reputation score, ρ^-, was defined as the sum of the abnormal KPI detections, calculated as follows:

$$\rho i^- = \sum_{k=1}^{m} \beta(k\ pik).\, 2^{n-(k\ pik)} \quad (14)$$

Where, $\beta(k\ pi_k)$ represents the punishment coefficient assigned to the k-th KPI, denoted as $k\ pi_k$, in the event of an abnormal detection result. The term $n^-(k\ pi_k)$ indicates the number of times that device D_i has shown abnormal detection for the specific KPI $k\ pi_k$. The variable m represents the total number of KPIs that need to be detected.

When a KPI abnormality occurs for device D_i, the negative reputation score ρ_i^- increases rapidly. Furthermore, the variable n_i^+, representing the number of consecutive normal detections, is reset to zero. As a result, the positive reputation score ρ_i^+ is also reset until all KPI detection results for device D_i return to normal. It is worth noting that the punishment coefficient $\beta(k\ pi_k)$ can be adjusted differently for each type of KPI, allowing for customization based on the specific sensitivity requirements of each KPI.

3.4 Classification of Devices Using Reputation Metrics

Devices can be grouped into three distinct classifications based on their reputation metrics. Using device D_i as a reference:

- **Normal device:** If $Ri = 100$ and the cumulative value $\sum_{k=1}^{m} n_i^-(k\ pi_k) = 0$, it indicates that device D_i is functioning optimally (or normally) without any detected KPI discrepancies.
- **Suspicious device:** For conditions where $50 \leq Ri < 100$ and $\sum_{k=1}^{m} n_i^-(k\ pi_k) > 0$ with $k \leq 2$, device D_i shows minor KPI deviations. These deviations might be due to minor device malfunctions. In response, the fog node might modify certain network settings for D_i, such as adjusting the data transmission rate or allocating a Specific bandwidth.
- **Abnormal device:** If the conditions $-\infty < Ri < 50$ or $\sum_{k=1}^{m} n_i^-(k\ pi_k) > 0$ are met, it suggests that multiple KPIs for device D_i are not functioning as expected. This could be a sign of a significant malfunction or a potential security breach. In such scenarios, the fog node communicates with the AMC to restrict access for device D_i, imposing a penalty duration defined by $P = 50 - R_i$.

By classifying devices based on these reputation metrics, the fog node can implement suitable measures to regulate device operations and bolster network safety.

4. EXPERIMENTAL WORK AND RESULTS

This section discusses the experimental work, highlighting the data set, evaluation metrics, and detection algorithms employed. The efficacy of the proposed KPI anomaly detection method will be evaluated, which leverages the synergies of the GAN-GRU hybrid model. Furthermore, the study investigates the influence of varying time windows on efficiently detecting KPI-based anomalies. In conclusion, a discussion of the synergistic benefits of merging access control techniques with anomaly detection to improve system performance.

4.1 Experimental Setup

To conduct the experiments, a simulation environment was set up using OMNET++, focusing on an SDN-based network with fog nodes. OMNET++ is a popular discrete event simulation framework widely used for simulating and analyzing complex network systems. Furthermore, NAB data set 1 (short for Numenta Anomaly Benchmark) was used for evaluation, which is an open anomaly detection data set that contains synthetic and real-world data. NAB encompasses various real-world time series data collected from different domains, including environmental sensors, server metrics, and industrial processes. Finally, the following three benchmark methods were used to compare with the proposed method.

1. **LSTM-VAE 2**: it utilizes both an encoder and a decoder. The encoder employs an LSTM network, complemented by two linear neural layers, to deduce the average and variance of the hidden variable z. Conversely, the decoder, mirroring the encoder's design, uses its LSTM and dual linear networks to determine the average and variance of the reformed variable x_{hat}. The process of detecting anomalies hinges on an anomaly score, derived from the log-likelihood of a given input concerning its reconstructed average and variance.
2. **TadGAN[8]**: an unsupervised anomaly detection approach built on Generative Adversarial Networks (GANs) that can effectively detect anomalies in time series data.
3. **Luminol 3**: a lightweight Python library for time series data analysis that supports two major functionalities: anomaly detection and correlation.

4.2 Anomaly Detection Performance

Table 1 compares the performance of various benchmark methods on different NAB data sets, focusing on precision, recall, and F1-score metrics. The proposed method outperforms other methods, such as LSTM-VAE, TadGAN, and Luminol, in most cases. This is because our proposed method better captures the time dependence of the data. For example, LSTM-VAE has been criticized for not considering the time dependence of the data, thus limiting its applicability to time series [9]. Moreover, the proposed method might have benefited from the use of GANs, which are effective in time series anomaly detection. Another factor contributing to the improved performance of the proposed method is its ability to adapt to different data sets. As seen in the table, the proposed method consistently performs well across all three data sets, whereas other methods, such as LSTM-VAE, show more variability in their performance.

Table 1. Comparison of the performance among various benchmark methods on different NAB data sets.

Method	Data Set 1*			Data Set 2**			Data Set 3***		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
LSTM-VAE	0.81	0.82	0.81	0.81	0.79	0.80	0.80	0.80	0.80
TadGAN	0.80	0.85	0.82	0.82	0.80	0.81	0.83	0.82	0.82
Luminol	0.78	0.80	0.79	0.79	0.77	0.78	0.80	0.78	0.79
Proposed	0.85	0.84	0.84	0.81	0.82	0.81	0.85	0.83	0.84

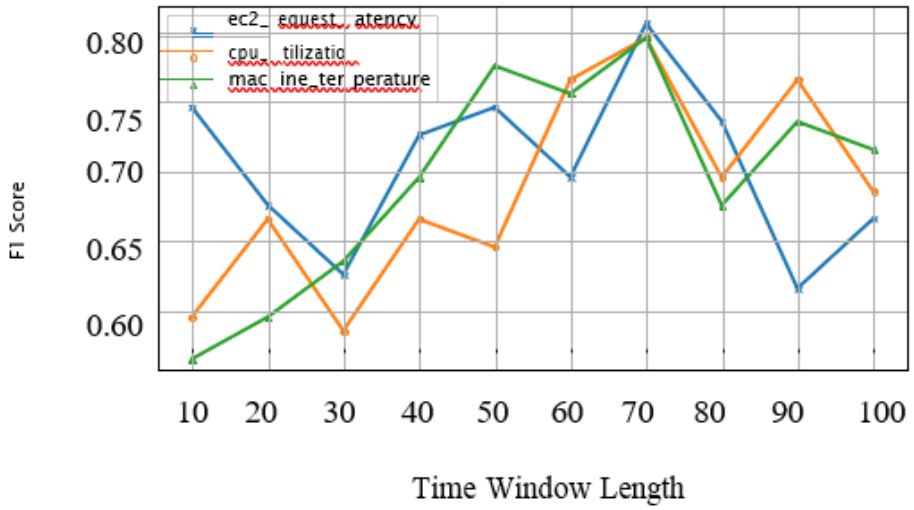
*ec2 request latency system failure

**CPU utilization as misconfiguration

*** Machine temperature system failure

4.3 Evaluating the Impact of Time Window Length on Anomaly Detection

The length of varying time windows can have varying effects on anomaly detection performance. To observe these changes, the time windows were divided into different lengths and the F1 scores were achieved by our proposed model.
On three different NAB data sets: ec2 request latency system failure, CPU utilization as misconfiguration, and machine temperature system failure. Ten experiments were conducted, each having different time window lengths ranging from 10 to 100. As illustrated in Figure 4, selecting a very short time window can hinder the model's ability to discern the underlying patterns in the KPI sequence. Conversely, an overly extended time window might cause the model to miss certain anomalies. In essence, both extremes can compromise the accuracy of anomaly detection. Based on our experimental results, optimal F1 scores were observed when the time window duration was fixed at 70 across all three datasets, which signifies optimal detection capabilities.



Time Window Length

Figure 4. The plot shows that the time window length affects the F1-score of the proposed method when used on the three different NAB data sets.

4.4 Evaluating the Reputation-Based Access Restriction Management Mechanism

To evaluate the effectiveness of the proposed reputation-based access restriction management mechanism, experiments were conducted using a simulated SDN network consisting of 50 devices. The reputation score of each device was determined based on its KPI detection performance, as described in the section. The reputation score was used to restrict access to the network, with devices having a higher reputation score being granted greater access privileges.

The proposed scheme was compared with a role-based access control (RBAC) model [10] and a centralized reputation management scheme [11]. The RBAC model grants access privileges based on the role of the user, while the centralized reputation management scheme identifies and isolates malicious controllers in a distributed SDN environment. The three models were evaluated based on their ability to prevent unauthorized access to the network and their impact on network performance. The reputation score of each device was calculated using Eq.11 and Eq.14.

As shown in 5, the proposed model outperformed the other two models. In the proposed model, the reputation score of each device was determined based on its KPI detection performance, capturing the device's trustworthiness and reliability through positive and negative reputation scores. In contrast, the RBAC model assigns access privileges based on user roles, while the centralized reputation management scheme identifies and isolates malicious controllers in a distributed SDN environment. Our proposed model effectively prevented unauthorized access to the network while ensuring reliable and secure operations by considering both positive and negative reputation scores. Conversely, the RBAC model may encounter challenges such as role explosions, requiring unforeseen expenses to support the access control system. As the number of roles within an organization increases, additional resources become necessary for implementing this access model. The centralized reputation management scheme faced limitations in large-scale networks due to its centralized nature, potentially resulting in performance issues and a single point of failure.

5 RELATED WORK

In recent years, various research efforts have been conducted in the field of access control mechanisms in SDN as well as KPI-based anomaly detection. This section provides an overview of the related work in this area.

5.1 SDN-enabled Cloud-Fog Collaborative Networks for Smart Cities

The combination of SDN and fog computing has been explored to enhance the efficiency and scalability of network management in distributed environments. The authors in [12] highlighted the potential of SDN-enabled fog computing to enable low-latency and real-time applications through distributed intelligence. The collaborative nature of cloud-fog networks allows for the delegation of tasks between cloud data centers and fog nodes, leading to reduced response times and enhanced resource utilization. Shi et al. emphasized the growing security concerns associated with fog devices, especially IoT devices, due to their limited capabilities and exposure to external threats [3]. These concerns underscore the need for effective security mechanisms, including anomaly detection, to safeguard fog.

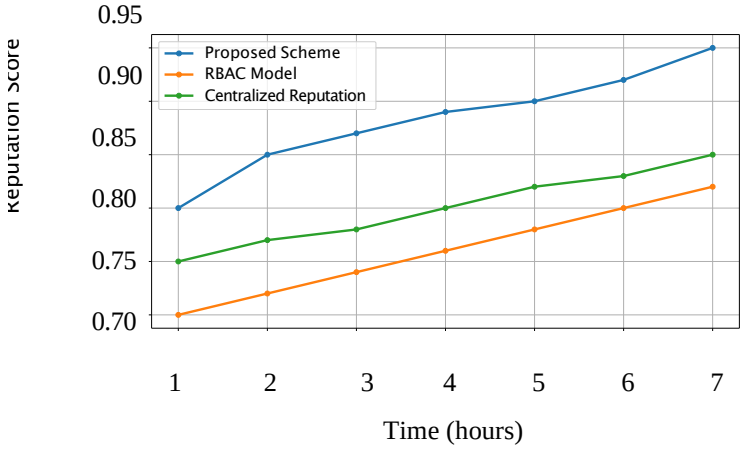


Figure 5. Reputation Score of the proposed model compared to RBAC and centralized reputation management scheme

Devices and the overall network.

Furthermore, in the context of smart cities, SDN-enabled cloud-fog collaborative networks have been leveraged to optimize traffic loads across distributed data centers, thereby enhancing the efficiency of Internet of Everything (IoE) services [20]. The authors in [19] proposed a multi-layer routing solution, Multi-Layer Advanced Networking Environment (Multi-LANE), specifically designed for fog-based deployment environments. This solution provides novel models for forwarding mechanisms specialized for fog computing scenarios. In another study, the authors proposed a load-balancing strategy for smart cities based on SDN-enabled cloud-fog networks, which significantly reduced latency and improved the quality of service [21]. These studies highlight the potential of SDN-enabled cloud-fog collaborative networks in enhancing the efficiency and security of smart city applications.

5.2 Access Control Mechanisms in SDN Networks

The SDN environment makes it essential to establish robust access control mechanisms. Matias et al. [6] proposed FlowNAC, an access control scheme that enables fine-grained control over user access based on their requested targets. However, this scheme suffers from time-consuming authentication processes and relies on third-party authorization, which introduces potential security vulnerabilities. While traditional access control methods, such as role-based access control, have been used, Li et al. pointed out their limitations in the context of SDN networks [5]. SDN's dynamic and programmable nature demands more flexible and fine-grained access control strategies that can adapt to changing network conditions. In [7], the authors presented a method based on an access control mechanism, which leverages the input with OpenFlow tables to achieve granular access control. However, a single authentication method is used, which lacks scalability when there are numerous endpoints to manage.

5.3 Deep Learning-based Anomaly Detection

Deep learning techniques have demonstrated promising results in various anomaly detection tasks due to their ability to automatically extract intricate features from high-dimensional data. Recurrent Neural Networks (RNNs) have been employed to capture temporal dependencies in time-series data [13]. Additionally, Generative Adversarial Networks (GANs) have shown the potential to generate synthetic data and identify anomalies through adversarial learning [14]. Specifically, the Gated Recurrent Unit (GRU) has been used to improve the performance of RNNs in sequential data processing [15].

5.4 The Proposed Method

In this paper, a novel method for anomaly detection in SDN-enabled cloud-fog collaborative networks for smart cities is proposed, leveraging the GAN-GRU model. The proposed method aims to combine the computational power of cloud data centers with the low-latency capabilities of fog nodes. By partitioning the smart cities network into small regions and distributing the detection tasks between the cloud and fog, efficient and timely KPI anomaly detection is achieved. Moreover, the GAN-GRU model's ability to process and analyze big data ensures accurate anomaly detection, while the collaborative architecture addresses the real-time requirements of IoT devices. Therefore, the proposed method holds promise in providing robust security solutions for SDN-enabled cloud-fog networks. The subsequent sections of this paper will delve into the details of our proposed method and present experimental results to validate its effectiveness in detecting malicious activities in SDN-enabled cloud-fog collaborative networks.

6 CONCLUSION

This paper addressed the challenges of anomaly detection in SDN-enabled cloud-fog collaborative networks, particularly focusing on the security concerns of IoT devices and real-time data processing in smart cities.

The proposed method combines a cloud-fog collaborative architecture with deep learning, utilizing the GAN-GRU model for efficient and accurate KPI anomaly detection.

The main contributions of this work are threefold. Firstly, introduced a novel cloud-fog collaborative architecture that leverages the strengths of cloud data centers and fog nodes. By distributing detection tasks between the cloud and fog, our approach achieves real-time and efficient KPI anomaly detection. The collaborative intelligence between the cloud and fog reduces the computational burden on the cloud while ensuring timely responses for detecting security threats.

Secondly, proposed a partitioning strategy that divides the network into subregions, enabling distributed detection tasks. The cloud is responsible for model training and maintaining a global view of network behavior, while fog nodes execute detection tasks locally on the data collected from devices within their respective subregions. This distributed approach not only enhances processing efficiency but also minimizes data transmission delays, addressing the scalability and real-time requirements of IoT devices.

Thirdly, applied the GAN-GRU model for KPI anomaly detection, showcasing its effectiveness in capturing complex patterns in large-scale network data. The use of deep learning and adversarial learning enables our method to achieve superior performance compared to traditional machine learning techniques. The proposed method consistently outperforms benchmark algorithms on different NAB data sets, demonstrating its capability to handle diverse real-world scenarios.

Moreover, explored the impact of time window length on the anomaly detection performance, identifying an optimal window length of 70 for achieving the best detection results. This analysis highlights the importance of carefully choosing time window lengths for effective anomaly detection in SDN networks. Additionally, a reputation-based access restriction management mechanism was introduced to enhance network security. By assigning reputation scores based on device KPI detection performance, our approach ensures that devices with higher reputation scores are granted greater access privileges. Experimental evaluations revealed the effectiveness of this mechanism in preventing unauthorized access while maintaining reliable and secure network operations. Future research can further explore the application of the proposed method in larger and more complex network environments and investigate potential extensions to other types of anomaly detection tasks.

REFERENCES

[1] MD Islam, Mojammel Hossain, Mohammed AlMukhtar, “A Survey on SDN & SDCN Traffic Measurement: Existing Approaches and Research Challenges,” *arXiv preprint arXiv:2206.14236*, 2022.
[2] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig, ”A Comprehensive Survey on Software-Defined Networking,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014, IEEE.
[3] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, Lanyu Xu, ”Edge Computing: Vision and Challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016, IEEE.
[4] Victor Heorhiadi, Seyed Kaveh Fayaz, Michael K Reiter, Vyas Sekar, ”SNIPS: A Software-Defined Approach for Scaling Intrusion Prevention Systems via Offloading,” *Information Systems Security: 10th International Conference, ICISS 2014*, pp. 9–29, 2014, Springer.
[5] Wenjuan Li, Weizhi Meng, Lam For Kwok, ”A Survey on OpenFlow-based Software Defined Networks: Security Challenges and Countermeasures,” *Journal of Network and Computer Applications*, vol. 68, pp. 126–139, 2016, Elsevier.
[6] Jon Matias, Jokin Garay, Alaitz Mendiola, Nerea Toledo, Eduardo Jacob, ”FlowNAC: Flow-Based Network Access Control,” *2014 Third European Workshop on Software Defined Networks*, pp. 79–84, 2014, IEEE.
[7] Diogo Menezes Ferrazani Mattos, Otto Carlos Muniz Bandeira Duarte, ”AuthFlow: Authentication and Access Control Mechanism for Software Defined Networking,” *Annals of Telecommunications*, vol. 71, pp. 607–615, 2016, Springer.
[8] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, Kalyan Veeramachaneni, ”Tadgan: Time Series Anomaly Detection Using Generative Adversarial Networks,” *2020 IEEE International Conference on Big Data (Big Data)*, pp. 33–43, 2020, IEEE.
[9] Yun Zhao, Xiuguo Zhang, Zijing Shang, Zhiying Cao, ”DA-LSTM-VAE: Dual-Stage Attention-Based LSTM-VAE for KPI Anomaly Detection,” *Entropy*, vol. 24, no. 11, p. 1613, 2022, MDPI.
[10] Priyanka Kamboj, Gaurav Raj, ”Analysis of Role-Based Access Control in Software-Defined Networking,” *Proceedings of Fifth International Conference on Soft Computing for Problem Solving: SocProS 2015, Volume 1*, pp. 687–697, 2016, Springer.
[11] Bilal Karim Mughal, Sufian Hameed, Ghulam Muhammad Shaikh, ”A Centralized Reputation Management Scheme for Isolating Malicious Controller(s) in Distributed Software-Defined Networks,” *arXiv preprint arXiv:1711.11005*, 2017.
[12] Je´ferson Campos Nobre, Allan M de Souza, Denis Rosa´rio, Cristiano Both, Leandro A Villas, Eduardo Cerqueira, Torsten Braun, Mario Gerla, ”Vehicular Software-Defined Networking and Fog Computing: Integration and Design Principles,” *Ad Hoc Networks*, vol. 82, pp. 172–181, 2019, Elsevier.
[13] Zachary C Lipton, John Berkowitz, Charles Elkan, ”A Critical Review of Recurrent Neural Networks for Sequence Learning,” *arXiv preprint arXiv:1506.00019*, 2015.
[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, ”Generative Adversarial Nets,” *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[15] Kyunghyun Cho, Bart Van Merrie¨nboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, ”Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[16] P William, Anurag Shrivastava, Hemant Chauhan, Pooja Nagpal, Prabhdeep Singh, and others, ”Framework for Intelligent Smart City Deployment via Artificial Intelligence Software Networking,” *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 455–460, 2022, IEEE.

[17] Ning Chen, Tie Qiu, Laiping Zhao, Xiaobo Zhou, Huansheng Ning, ”Edge Intelligent Networking Optimization for Internet of Things in Smart City,” *IEEE Wireless Communications*, vol. 28, no. 2, pp. 26–31, 2021, IEEE.

[18] Mehdi Gheisari, Guojun Wang, Shuhong Chen, Hamidreza Ghorbani, ”IoT-SDNPP: A Method for Privacy-Preserving in Smart City with Software Defined Networking,” *Algorithms and Architectures for Parallel Processing: 18th International Conference, ICA3PP 2018*, pp. 303–312, 2018, Springer.

[19] Paolo Bellavista, Carlo Giannelli, Dmitrij David Padalino Montenero, ”A Reference Model and Prototype Implementation for SDN-Based Multi-Layer Routing in Fog Environments,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1460–1473, 2020, IEEE.

[20] Ji-Young Kwak, Chunglae Cho, YongYoon Shin, Sunhee Yang, ”IntelliTC: Intelligent Inter-DC Traffic Controller for the Internet of Everything Service Based on Fog Computing,” *IET Communications*, vol. 14, no. 2, pp. 193–205, 2020, Wiley Online Library.

[21] Subhranshu Sekhar Tripathy, Diptendu Sinha Roy, Rabindra K Barik, ”M2FBalancer: A Mist-Assisted Fog Computing-Based Load Balancing Strategy for Smart Cities,” *Journal of Ambient Intelligence and Smart Environments*, vol. 13, no. 3, pp. 219–233, 2021, IOS Press.