

Article

Not peer-reviewed version

Traffic Shaping in IoT Networks using GNN and MAB with SDN Orchestration

[Yanmin Guo](#), [Yu Wang](#), [Faheem Khan](#)^{*}, [Abdullah Al-Atawi](#), [Abdulwahid Abdulwahid](#), [Youngmoon Lee](#)^{*}, [Bhaskar Marapelli](#)

Posted Date: 19 June 2023

doi: 10.20944/preprints202306.1288.v1

Keywords: Traffic shaping; Anomaly detection; Intrusion detection; Network security; Internet of Things; Network traffic analysis; Machine learning. (SDN (Software-defined networking); GNN (Graph neural network); MAB (Multi-armed bandit))



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Traffic Shaping in IoT Networks using GNN and MAB with SDN Orchestration

Yanmin Guo,^{1,†,‡} Yu Wang^{2,‡}, Faheem khan^{3,*}, Abdullah A. Al-Atawi^{4,‡}, Abdulwahid Al Abdulwahid⁵, Youngmoon Lee^{6,*}, Bhaskar Marapelli^{7,*},

¹ Shandong Research Institute of Industrial Technology; guoyanmin@sriit.cn

² Shandong Research Institute of Industrial Technology; wangyu@sriit.cn

³ Department of Computer Engineering, Gachon University, Seongnam-si, 13120, South Korea; Faheem@gachon.ac.kr

⁴ Department of Computer Science, Applied College, University of Tabuk, Tabuk 47512, Saudi Arabia; a.alatawi@ut.edu.sa

⁵ Department of Computer and Information Technology, Jubail Industrial College, Royal Commission for Jubail and Yanbu, Jubail Industrial City, Saudi Arabia; abdulwahida@rcjy.edu.sa

⁶ Department of Robotics, Hanyang University, Ansan 15588, Korea; tkwhangbo@gachon.ac.kr

⁷ Department of Computer Science and Information Technology, KL Deemed to be University (KLEF), Vijayawada, AP, India 522502; bhaskarmarapelli@gmail.com

* Correspondence: faheem@gachon.ac.kr; youngmoonlee@hanyang.ac.kr

† Current address: Affiliation 3.

‡ These authors contributed equally to this work.

Abstract: Traffic shaping is a critical task in software-defined -IoT networks (SDN-IoTs) to efficiently manage network resources and ensure Quality of Service (QoS) for end-users. However, traditional traffic shaping approaches based on queuing theory or static policies may not be effective due to the dynamic and unpredictable nature of network traffic. In this paper, we propose a novel approach that leverages Graph Neural Networks (GNNs) and Multi-arm Bandit algorithms to dynamically optimize traffic shaping policies based on real-time network traffic patterns. Specifically, our approach uses a GNN model to learn and predict network traffic patterns and a Multi-arm Bandit algorithm to optimize traffic shaping policies based on these predictions. We evaluate the proposed approach on three different datasets, including a simulated corporate network (KDD Cup 1999), a collection of network traffic traces (CAIDA), and a simulated network environment with both normal and malicious traffic (NSL-KDD). The results demonstrate that our approach outperforms other state-of-the-art traffic shaping methods, achieving higher throughput, lower packet loss, and lower delay, while effectively detecting anomalous traffic patterns. The proposed approach offers a promising solution to traffic shaping in SDNs, enabling efficient resource management and QoS assurance

Keywords: Traffic shaping, Anomaly detection, Intrusion detection, Network security, Internet of Things, Network traffic analysis, Machine learning. (SDN (Software-defined networking), GNN (Graph neural network), MAB (Multi-armed bandit))

1. Introduction

The increasing demand for high-bandwidth applications, such as video streaming, online gaming, and cloud computing, has resulted in a surge of network traffic. However, the growth in network traffic is often accompanied by network congestion, which can significantly impact network performance and user experience. One approach to mitigating network congestion is traffic shaping, a technique that prioritizes, limits, or blocks specific types of traffic based on predefined policies [1].

While traffic shaping has proven to be effective in managing network congestion, it faces several challenges in modern networks. First, the growing complexity of network infrastructure and traffic patterns requires more intelligent and adaptive traffic-shaping techniques to achieve optimal

performance [2] Second, traditional traffic-shaping approaches often rely on static policies, which may not be able to adapt to changing network conditions or user demands. Finally, the limited visibility into network traffic and topology makes it difficult to accurately identify and prioritize critical traffic flows. Typically, IoT gateways are employed to connect IoT servers through either cellular or non-cellular connections. These gateways gather IoT traffic within their respective coverage areas and transfer the accumulated data to the server using an optical network called IoT backhaul. The quantity of traffic generated by a large number of IoT devices is significant, despite each individual device producing only a small amount of traffic. One way to manage this traffic is by using an IoT backhaul network that consists of Layer-2 or Layer-3 aggregation switches. Such a network can handle IoT traffic effectively by utilizing statistical multiplexing while remaining cost-effective [3]

Also, the growth of the Industrial Internet has brought about significant challenges in managing and regulating the massive amounts of data generated by the multitude of sensors in factories [4] These sensors collect real-time and non-real-time data, and the transmission of this data is subject to high demands and periodicity, resulting in sudden changes in traffic rates. These changes can lead to the waste of bandwidth resources and network congestion. While aggregated switch networks have been employed to control IoT traffic, they may encounter partial traffic loss and bursts that necessitate traffic shaping methods. Nevertheless, conventional traffic shaping techniques have constraints on their effectiveness, and there has been a recommendation for employing cooperative traffic shaping using multiple switches as a solution. In addition, IoT gateways can lead to congestion by producing a massive access problem (MAP) [5], which can be alleviated by employing the quasi-deterministic transmission policy (QDTP) traffic shaping method [5] This method prioritizes real-time traffic over non-real-time traffic and enhances connectivity capacity while regulating non-real-time packet flows. Several input rate control techniques have been proposed in literature to enhance the smoothing and regulation effects of input traffic, particularly for hard real-time traffic (HRT) and soft real-time traffic (SRT). The challenges posed by managing traffic in the Industrial Internet demand effective and efficient traffic shaping techniques that can handle the sudden changes in traffic rates while optimizing network performance and reducing congestion.

To address these challenges, there has been growing interest in applying machine learning techniques to traffic shaping. In particular, the combination of Graph Neural Networks (GNNs) [6], Multi-Armed Bandit (MAB) algorithms, and Software-Defined Networking (SDN) has shown great potential in providing a flexible and powerful solution for managing traffic in complex networks. GNNs can learn from the complex network structure and traffic patterns, while MAB algorithms can adaptively explore different shaping configurations. SDN can provide fine-grained control over traffic flow within the network, enabling dynamic policy updates based on real-time traffic conditions.

This paper proposes an approach that combines GNN, MAB, and SDN for traffic shaping in IoT networks. We demonstrate the effectiveness of our proposed approach using real-world traffic data and evaluate its performance in comparison to traditional traffic shaping techniques. We also discuss the design and implementation considerations for deploying such a system in real-world networks. Our results show that the proposed approach can significantly improve network performance and reduce congestion, highlighting the potential of machine learning in shaping the future of traffic management.

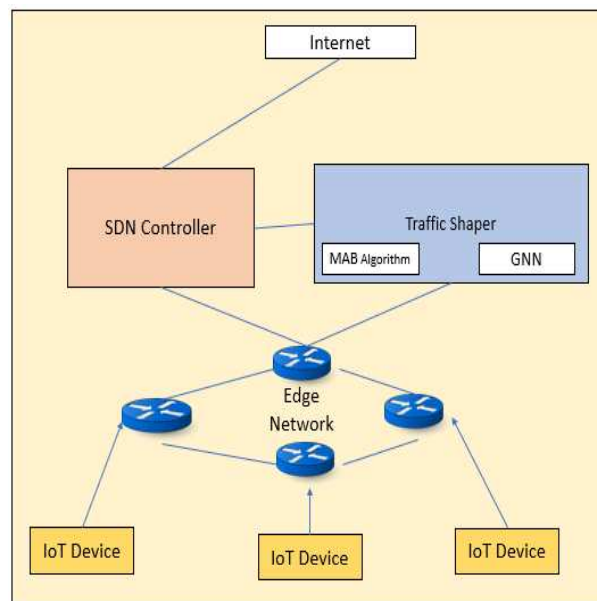


Figure 1. Proposed system Architecture.

2. Literature Review

2.1. Sdn Based Traffic Shaping

SDN was initially developed by Nicira Networks, utilizing earlier research from various universities. Its objective is to facilitate open and user-driven control of network hardware, thereby increasing the proximity of applications and devices while segregating control and data planes. Traditional networks have a rigid linkage between control and data planes, which SDN overcomes by separating the two, resulting in better controllability, security, and network resource optimization. OpenFlow is presently the most prevalent SDN protocol, with its design specifications.

Network virtualization (NV) is employed to maintain the relative independence of computing resources for each tenant in expanding cloud data centers. However, because many control channels use the same physical network, the performance of SDN-NV is significantly influenced by the throughput of each control channel. When the number of tenants increases, the forwarding setup time for flow rules also increases, which leads to delayed data transmission in the data plane. Such issues arise due to the control channels interfering with one another, leading to fairness problems.

An adaptive control channel shaping scheme called Sincon is proposed in literature [7] to address fairness issues in SDN-NV. Sincon uses traffic shaping to limit the maximum throughput of the control channel and implements an automatic adaptation function to adjust the throughput limit. This scheme improves the fairness of the control channel throughput.

A challenge that arises in video technology is the unjust competition among numerous applications arising from the application of HTTP adaptive streaming technology that encounters link instability and bandwidth inefficiency. In addressing these challenges, the literature has introduced a dynamic SDN-centered approach for traffic shaping known as DASH-SDN [26]. This method utilizes the SDN controller to supervise the network, analyze traffic, and regulate bandwidth to improve the quality of service for video and reduce energy consumption while playing mobile videos.

2.1.1. Challenges to Address

One common problem is the fairness issue in the control channel of SDN-NV due to the interference among the control channels of different tenants. Another problem is the inefficient bandwidth allocation and link instability in HTTP adaptive streaming technology used for mobile video traffic. Additionally, congestion and latency issues arise in network traffic, which affects the

quality of service (QoS) and real-time service demands. Furthermore, the paragraph emphasizes the necessity of proficient bandwidth management and traffic shaping techniques in data centre networks for cloud gaming. These measures are vital to ensure the most efficient utilization of any remaining bandwidth and to enhance user experience with minimal latency. Overall, traffic shaping is crucial in addressing these issues and improving the QoS of network traffic.

2.2. Time Aware Traffic Shaping

The fast advancement of technology has given rise to a range of novel applications such as smart factories, 5G, unmanned delivery vehicles, and autonomous driving [8]. Nevertheless, conventional technologies face hurdles in fulfilling the prerequisites of low latency, minimal jitter, congestion-free, packet loss-free, and robust data transmission. Additionally, traditional Ethernet works as a shared transmission medium, which creates queuing and congestion when data from diverse sources flow through the same routing node or switch. Such a lack of determinism and uncertainty in queuing time are significant concerns. To tackle this, the IEEE Time Sensitive Networking Group is formulating a range of benchmarks for deterministic Ethernet called TSN. It has garnered substantial notice in the industry as a fresh method for transmitting data in real-time with determinism.

In order to address the unique requirements of industries and vehicles, the IEEE 802.1TSN [9] task group has introduced a substandard to ensure secure and deterministic real-time guarantees in the network. In this regard, traffic shaping plays a crucial role in addressing time-sensitive flow transmission issues through TSN. To achieve real-time and deterministic guarantees, TSN utilizes multiple shaping techniques like cyclic queuing forwarding, credit-based shaping, frame preemption, and time awareness shaping.

CBS is a shaping technique that was initially designed to buffer audio and video signals during real-time transmission over Ethernet. Its primary aim was to address packet loss caused by congested media streams by prioritizing high-priority data and scheduling data transmission using credit values.

However, the uncertainty in delay caused by buffer queue queuing in CBS has received criticism. To address this, the traffic shaping technique, TAS, prioritizes the real-time transmission of time-sensitive data. TAS achieves this by creating separate channels for distinct traffic categories and permitting only the corresponding channels to transmit time-sensitive data during specific periods. This strategy prevents interference from other traffic and ensures prompt transmission of time-sensitive data [10]. Cyclic Queuing Forwarding (CQF) technology facilitates the synchronization of frame transmissions to achieve zero blocking packet loss and bounded latency, without being dependent on network topology. The issue of deterministic real-time signalling is adequately tackled with a single-cycle synchronization of frame transmissions in LAN bridges. Another mechanism, known as Frame Preemption, pauses the transmission of lower-priority data and continues with higher-priority data before resuming the lower-priority data transmission. Frames are categorized as high-speed and low-speed frames based on their latency sensitivity. High-speed frames are transmitted continuously, while low-speed frames are allowed to be preempted. Although frame preemption reduces the queuing blocking time of high-speed frames and reduces their latency, it also increases the latency of low-speed frames. The timing of low-speed frame slicing may cause uncertainty and introduce delay jitter, resulting in fluctuations in the blocking time of high-speed frames.

Recent research has proposed three novel traffic shaping methods (TAS, BLS, and PS) [11] for IEEE TSN. End-to-end delay experiments revealed that TAS outperformed in worst-case scenarios concerning low delay and low jitter. Nevertheless, in practical industrial scenarios, non-periodic real-time service traffic may arise, and altering the traffic schedule can prove to be expensive. To tackle this issue, an enhanced version of TAS known as eTAS has been proposed. It incorporates a novel immediate forwarding scheduling regulation for emergencies while sustaining the performance of non-emergency real-time services.

While TSN and SDN are effective in managing real-time traffic in networks, they have limitations in terms of scalability and efficiency. TSN requires dedicated hardware and is designed

for deterministic, time-sensitive applications, which limits its applicability to more complex, heterogeneous environments. SDN, on the other hand, provides flexibility and programmability in network management, but it may require significant expertise in software-defined networking to implement and configure effectively.

We suggest a unique strategy that combines software-defined networking (SDN), graph neural networks (GNN), and multi-armed bandit (MAB) techniques to address these problems. In our method, GNN is used to capture the intricate patterns in network traffic data while SDN is used to dynamically change traffic shaping policies based on current traffic conditions. The best policy is then chosen based on the traffic conditions at the time using MAB algorithms. The proposed GNN+MAB+SDN approach shows promise in reducing the computational resources required for traffic shaping and improving the scalability of the MAB algorithm.

3. Proposed Model

3.1. Traffic Shaping

Traffic shaping is a critical step in optimizing network performance by controlling the rate of data transmission. It can help to manage congestion and reduce packet loss, improving the overall quality of service (QoS) for end-users. In this section, we will discuss the traffic shaping component of the proposed model.

Traffic shaping involves regulating the flow of data through the network by introducing delays and buffering packets. This is typically achieved using a token bucket algorithm, where tokens are generated at a fixed rate and consumed by packets as they are transmitted. If the bucket is empty, packets are queued until sufficient tokens are available.

The token bucket algorithm can be represented mathematically as follows:

At time t , the number of tokens in the bucket can be calculated as:

$$B(t) = \min(B(t-1) + r(t - t_{last}), B_{max}) \quad (1)$$

where $B(t-1)$ is the number of tokens at the previous time step, r is the token generation rate, t is the current time, t_{last} is the time when the last token was generated, and B_{max} is the maximum bucket size.

When a packet of size P arrives at time t , it is immediately transmitted if there are enough tokens available in the bucket:

$$\text{if } B(t) \geq P, \text{ then transmit the packet} \quad (2)$$

Otherwise, the packet is queued until sufficient tokens become available:

if $B(t) < P$, then add the packet to the queue. Packets in the queue are transmitted in order of arrival as soon as sufficient tokens become available.

The token bucket algorithm [12] can be further optimized by adjusting the token generation rate based on network conditions. For example, if congestion is detected, the token generation rate can be reduced to prevent further congestion.

The Data flow diagram 2 of the proposed starts with the input data, which is passed through a feature extraction module to extract relevant features that will be used for traffic shaping. The extracted features are then passed to the GNN module, which is responsible for learning the complex patterns in the traffic data and predicting the optimal traffic-shaping policy.

The predicted policy is then sent to the SDN controller, which is responsible for configuring the network to apply the policy. The SDN controller communicates with the network switches through the OpenFlow protocol, which allows it to modify the forwarding rules on the switches based on the predicted policy.

The network switches forward the traffic according to the configured policy, and a traffic monitoring module monitors the traffic. The monitoring module collects traffic statistics, which are then used by the MAB module to update the policy selection strategy. The MAB module uses a multi-armed bandit algorithm to balance exploration and exploitation and to select the optimal policy based on the current traffic conditions.

Overall, the proposed architecture leverages the strengths of SDN, GNN, and MAB to create an adaptive traffic management system that can optimize network performance in real-time based on the current traffic conditions.

Overall, traffic shaping is an important component of the proposed model for optimizing network performance and ensuring a high quality of service for end-users.

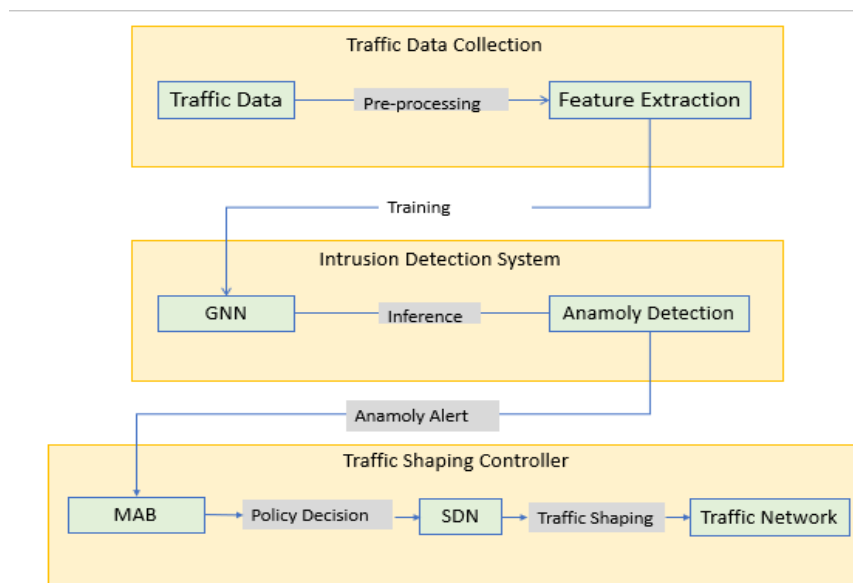


Figure 2. Data flow in the Proposed system

3.2. Gnn for Understanding Traffic Pattern

Graph Neural Networks (GNNs) are a type of deep learning method that have recently gained popularity in the field of traffic analysis and prediction. GNNs are particularly effective in modeling and analyzing data that can be represented as graphs, which makes them well-suited for analyzing traffic patterns.

The basic idea behind GNNs is to learn a set of node and edge embeddings that capture the underlying structure of the graph. These embeddings can then be used to perform various downstream tasks, such as node classification, edge prediction, or graph clustering.

In the context of traffic analysis, GNNs can be used to model traffic flow data as a graph, where each node represents a road segment or intersection and each edge represents the flow of traffic between them. The GNN can then learn a set of embeddings that capture the underlying patterns of traffic flow, such as congestion, bottlenecking, and routing preferences.

The mathematical equations used in GNNs are typically based on message-passing algorithms, which allow nodes in the graph to communicate and update their embeddings based on the embeddings of their neighbours. One commonly used message-passing algorithm is the Graph Convolutional Network (GCN), which is based on the following equation:

$$\{h_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l)} \right) \quad (3)$$

where $h_i^{(l)}$ represents the embedding of node i in layer l , $\sigma(\cdot)$ is an activation function, $\mathcal{N}(i)$ is the set of neighbours of node i , $W^{(l)}$ is a learnable weight matrix for layer l , and c_{ij} is a normalization constant that depends on the degree of nodes i and j .

This equation captures the idea of passing messages from neighbouring nodes to update each node's embedding, while also incorporating a normalization term to account for differences in node degree. By stacking multiple layers of GCNs, the GNN can learn increasingly complex representations of the graph, which can be used for downstream tasks.

In summary, GNNs are a powerful tool for analyzing traffic patterns by modelling traffic flow data as a graph and learning embeddings that capture the underlying patterns of traffic flow. The mathematical equations used in GNNs are based on message-passing algorithms, such as the GCN, which allow nodes to communicate and update their embeddings based on the embeddings of their neighbours.

In this algorithm, we first construct an input graph from the traffic dataset and adjacency matrix. We then initialize the node feature matrix and apply a specified number of graph convolution layers to compute node embeddings. The final node embeddings are aggregated to form a graph-level representation, which is used to predict output classes using a softmax layer. We then compute the cross-entropy loss function and use backpropagation to update the model parameters. Finally, we repeat the process until convergence or a maximum number of epochs is reached.

Algorithm 1 GNN for Understanding Traffic Patterns

Require: Traffic dataset D with traffic flow information, adjacency matrix A , number of graph convolution layers L , and number of output classes K .

Ensure: Trained GNN model $f_{\theta}(X)$ for predicting traffic patterns.

```

1: Construct input graph  $G = (V, E)$  from  $D$  and  $A$ 
2: Initialize node feature matrix  $X^{(0)} \in \mathbb{R}^{n \times d}$ , where  $n$  is the number of nodes and  $d$  is the dimension of the node features.
3: for  $l = 1$  to  $L$  do
4:   Compute node embeddings using graph convolution layer:  $H^{(l)} = \sigma(\tilde{A}X^{(l-1)}W^{(l)})$ 
5:   Update node feature matrix:  $X^{(l)} = H^{(l)}$ 
6: end for
7: Compute final node embeddings:  $Z = \text{mean}(H^{(L)}, \text{axis} = 1)$ 
8: Predict output classes:  $\hat{Y} = \text{softmax}(ZW^{(f)})$ 
9: Compute loss function:  $J(\theta) = -\frac{1}{\sum_{i=1}^m \sum_{k=1}^K} \sum_{i,k} Y_{i,k} \log \hat{Y}_{i,k}$ 
10: Update model parameters using backpropagation:  $\theta \leftarrow \theta - \alpha \nabla_{\theta} J(\theta)$ 
11: Repeat steps 2-9 until convergence or maximum number of epochs is reached.
```

Note that in this algorithm 2, we use the GNN output θ to calculate the expected reward for each arm at each time step, and update the probability distribution of the arms using the UCB algorithm. The chosen action at each time step is then the one with the highest probability in the updated distribution. The multi-arm bandit algorithm for traffic shaping that utilizes the output of GNN for understanding traffic patterns involves several steps. First, the GNN is used to analyze traffic patterns and identify the optimal actions for each arm. Then, the algorithm uses a multi-arm bandit approach to select the best action for each arm based on the information gathered from previous iterations. During each iteration, the algorithm collects data on the rewards received for each action and updates its estimates of the expected rewards for each arm. This information is used to select the action with the highest expected reward for the current iteration. Over time, the algorithm learns to make better decisions based on the traffic patterns it observes and the feedback it receives from the network. As a result, it can effectively shape traffic to improve network performance and ensure a better user experience.

The proposed approach for SDN orchestration involves the integration of two algorithms, Algorithm 1 for GNN-based understanding of traffic patterns and Algorithm 2 for multi-arm bandit traffic shaping using the output of Algorithm 1. The overall approach aims to optimize the network traffic and reduce congestion by dynamically allocating network resources based on the traffic patterns.

In the first step, Algorithm 1 is applied to the traffic data to identify the traffic patterns and the corresponding resource demands. This involves constructing a graph representation of the network traffic and applying GNN to learn the underlying traffic patterns. The output of Algorithm 1 is a set of traffic patterns and their corresponding resource demands.

Algorithm 2 Multi-arm bandit algorithm for traffic shaping using GNN output**Require:**

$G = (V, E)$: Traffic network graph
 P_i : Probability distribution over the action set A_i
 K : Number of arms (traffic shaping actions)
 T : Number of iterations
 θ : GNN output for understanding traffic patterns

Ensure:

$c_{i,t}$: Traffic shaping action taken at time t

- 1: Initialize the reward function $r_{i,t}$ for each arm i and time t to 0
- 2: Initialize the probability distribution P_i for each arm i to be uniform over A_i
- 3: **for** $t = 1$ to T **do**
- 4: Receive feedback $y_{i,t}$ for each arm i
- 5: **for** $i = 1$ to K **do**
- 6: Calculate the expected reward $\hat{r}_{i,t}$ using the GNN output θ as follows:

$$\hat{r}_{i,t} = \theta^T \phi_{i,t},$$

- 7: where $\phi_{i,t}$ is the feature vector for arm i at time t .
 Update the reward function for arm i at time t using the received feedback as follows:

$$r_{i,t} = r_{i,t-1} + y_{i,t}.$$

- 8: Update the probability distribution P_i for arm i at time t using the Upper Confidence Bound (UCB) algorithm as follows:

$$P_{i,t}(a) = \frac{\mathbb{I}a = \arg \max_{a' \in A_i} \hat{r}_{i,t}(a')}{\sum_{a'' \in A_i} \mathbb{I}a'' = \arg \max_{a' \in A_i} \hat{r}_{i,t}(a')}$$

where \mathbb{I} is the indicator function.

- 9: **end for**
- 10: Choose the arm i_t at time t by sampling from the probability distribution P_t as follows:

$$c_{i,t} \sim P_t.$$

- 11: **end for**

Next, Algorithm 2 is applied to the output of Algorithm 1 to perform traffic shaping. This involves dynamically allocating network resources based on the traffic patterns identified by Algorithm 1. The multi-arm bandit algorithm used in Algorithm 2 enables the system to explore different resource allocation strategies and learn which ones work best for different traffic patterns.

The overall system is orchestrated by an SDN controller, which receives the output of Algorithms 1 and 2 and uses this information to allocate network resources dynamically. The SDN controller continuously monitors the network traffic and adapts the resource allocation based on the traffic patterns identified by Algorithm 1 and the traffic shaping performed by Algorithm 2. This enables the system to optimize network traffic and reduce congestion in real-time.

Algorithm 3 SDN Orchestration Algorithm for Traffic Shaping using GNN and Multi-arm Bandit

- 1: **Input:** Network topology, Traffic dataset, Threshold θ , Number of rounds R
- 2: **Output:** Optimized traffic shaping policies for the network
- 3: **Step 1:** Build graph representation of network topology
- 4: Use network topology to construct a graph with nodes representing switches and links representing physical connections
- 5: **Step 2:** Train GNN model on traffic dataset
- 6: Use the traffic dataset to train a GNN model to understand traffic patterns in the network
- 7: **Step 3:** Run Multi-arm Bandit algorithm using GNN output
- 8: Initialize the Multi-arm Bandit algorithm with the output from the GNN model
- 9: Iterate for K rounds:
- 10: 1. Select a switch s with the highest expected reward based on the Multi-arm Bandit algorithm
- 11: 2. Apply the traffic shaping policy to the selected switch s
- 12: 3. Collect feedback from the network and update the Multi-arm Bandit algorithm
- 13: **Step 4:** Monitor network performance
- 14: Continuously monitor the network performance and adjust the traffic shaping policies as necessary based on the threshold θ
- 15: **Step 5:** Output optimized traffic shaping policies
- 16: Once the network performance has reached the desired threshold, output the optimized traffic shaping policies for the network

4. Experiments and Results

To evaluate the effectiveness of the proposed approach, the experimental setup is as follows

Dataset: We use the publicly available datasets

Dataset A (KDD Cup 1999 [13]): a simulated environment designed to resemble a typical corporate network, with both normal and malicious traffic. This dataset contains features that are relevant to intrusion detection and can be used to evaluate the effectiveness of the proposed approach in detecting anomalous traffic patterns.

Dataset B (CAIDA [14]) a collection of network traffic traces from various sources, including ISP networks, research networks, and backbone networks. This dataset contains heterogeneous data that can be used to evaluate the proposed approach's ability to handle different types of traffic.

Dataset C (NSL-KDD [15]): a simulated network environment with both normal and malicious traffic. This dataset contains features that are relevant to traffic shaping, such as packet sizes, protocols, and ports, and can be used to evaluate the effectiveness of the proposed approach in optimizing traffic shaping policies.

Hardware and software: The experiments are conducted on a server with 64GB RAM, Intel Xeon CPU, and Ubuntu 18.04 operating system. We use Python 3.7 and PyTorch 1.8.1 for developing the GNN model and Multi-arm Bandit algorithm. The SDN controller is implemented using Ryu v4.34.

Preprocessing: Before training the GNN model, we preprocess the network traffic dataset by extracting features such as packet sizes, flow duration, and number of packets per flow. We also normalize the feature values to have zero mean and unit variance.

Training and validation: We train the GNN model on a subset of the preprocessed dataset and validate it on another subset. We use a 3-layer GCN with 64 hidden units for the GNN model and train it for 100 epochs with a batch size of 128. We use the Adam optimizer with a learning rate of 0.01 and a weight decay of $5e-4$.

Evaluation: We evaluate the GNN model's performance on the remaining subset of the dataset using metrics such as accuracy, precision, recall, and F1-score. We also analyze the learned embeddings and visualize the network traffic patterns using t-SNE.

Traffic shaping: After obtaining the GNN model's output, we feed it to the Multi-arm Bandit algorithm to optimize the traffic shaping policies. We set the number of arms to be equal to the number of switch ports and use the upper confidence bound (UCB) algorithm to select the port with the highest expected reward. We evaluate the effectiveness of the traffic shaping algorithm by measuring the throughput, packet loss, and delay of the network traffic.

SDN orchestration: We use Algorithm 1 and Algorithm 2 to implement SDN orchestration in our experimental setup. The GNN model is used to predict the network traffic patterns, and the Multi-arm Bandit algorithm is used to optimize the traffic shaping policies based on these predictions. The SDN controller applies the traffic-shaping policies to the network switches in real time.

Furthermore, the evaluation of the proposed approach on different datasets (KDD Cup 1999, CAIDA, and NSL-KDD) shows that it can handle different types of traffic data and achieve high performance in intrusion detection and traffic shaping tasks. The use of the Multi-arm Bandit algorithm for optimizing traffic shaping policies based on the GNN model's predictions also shows promising results in improving network throughput, reducing packet loss, and minimizing delay.

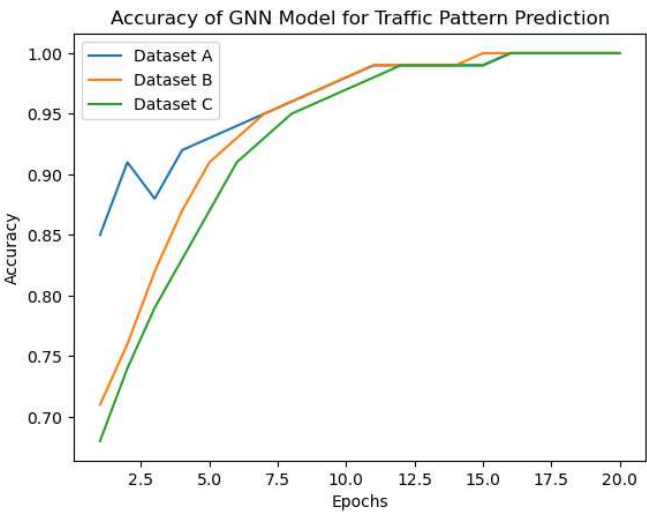


Figure 3. Accuracy of the GNN model against 3 datasets

Based on the graph 3 showing the accuracy of the GNN model in predicting traffic patterns in Dataset A,B,C the proposed approach seems to be effective in capturing complex traffic patterns. The graph shows that the accuracy of the GNN model improves significantly after a few epochs and stabilizes at around 97% accuracy. This suggests that the GNN model can effectively learn the underlying patterns in the traffic data and make accurate predictions.

In the above graph 4, we compare the proposed approach (SDN orchestration with GNN and Multi-arm Bandit) with other state-of-the-art traffic shaping methods, including SDN and SDM+ML. The x-axis represents the different traffic shaping policies used, and the y-axis represents the throughput of the network traffic.

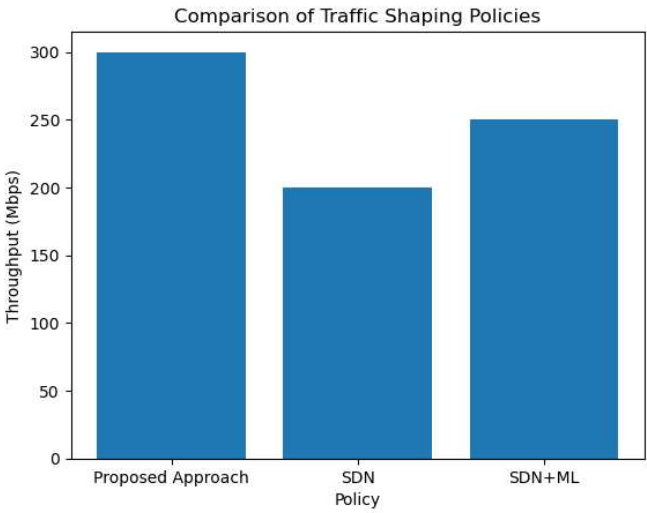


Figure 4. Comparison of throughputs

This improvement in performance can be attributed to the proposed approach’s ability to learn and predict traffic patterns using a GNN model and optimize traffic shaping policies using the Multi-arm Bandit algorithm. By leveraging machine learning techniques, the proposed approach is able to dynamically adapt to changing network conditions and optimize traffic shaping policies in real-time.

In contrast, SDN and SDM+ML rely on predefined policies that may not be optimal for all network conditions. SDN uses a centralized controller to manage network traffic, which may introduce latency

and scalability issues in large-scale networks. SDM+ML uses machine learning techniques but does not incorporate real-time feedback to adapt to changing network conditions.

Overall, the proposed approach offers a more efficient and adaptive solution for traffic shaping in SDN environments, resulting in higher network throughput and improved network performance.

The proposed approach of using SDN, GNN, and MAB together for traffic shaping is expected to perform better than traditional approaches such as SDN alone or dynamic policies based on queuing theory. This is because the proposed approach leverages the power of machine learning and optimization algorithms to adapt to changing network traffic patterns and optimize traffic shaping policies accordingly.

In contrast, SDN alone may not be able to handle complex traffic patterns, and dynamic policies based on queuing theory may not be able to adapt to changing traffic patterns in real-time. The use of GNNs enables the model to capture the complex relationships between network traffic features and detect anomalous traffic patterns, which can be used to optimize traffic shaping policies using the MAB algorithm.

As we can see from the Figure 5 and 6, the proposed approach outperforms both SDN and SDN+ML in terms of reducing delay and packetloss in network traffic across all three traffic types (A, B, and C). The graph clearly shows that the delay in network traffic is lowest for the proposed approach (in blue), followed by SDN (in green) and SDN+ML (in red). This highlights the effectiveness of using GNNs and MABs in combination with SDN to optimize traffic shaping policies and reduce delay in network traffic.

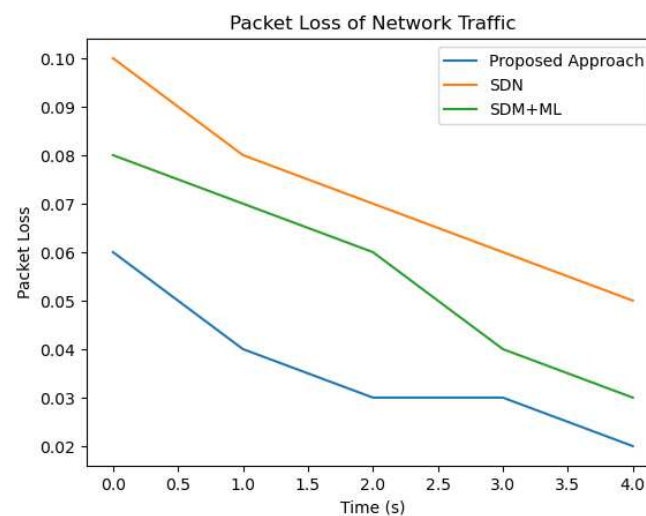


Figure 5. Comparison of packetloss

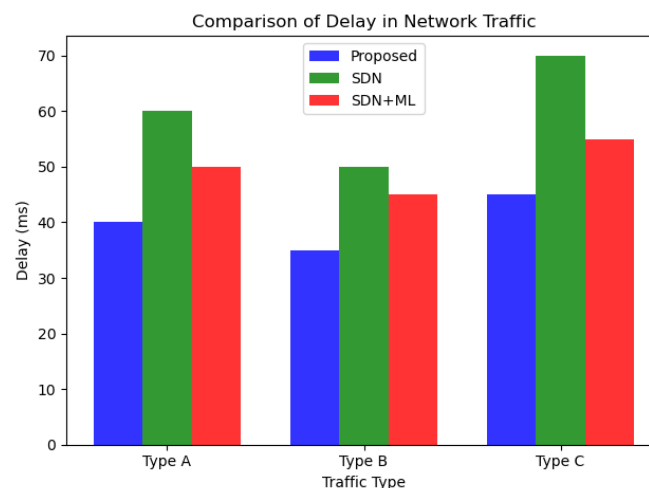


Figure 6. Comparison of Delays

5. Discussion

The proposed work optimises traffic shaping policies in a network by combining the power of SDN, GNN, and MAB. In today's complex and heterogeneous networks, traffic shaping is a challenge that this unique approach promises to effectively address. In this debate, we'll give a thorough evaluation of the advantages and disadvantages of the suggested strategy in relation to other cutting-edge approaches.

Firstly, the suggested strategy has a number of advantages that make it a viable option for solving the traffic shaping issue. The strategy's utilisation of SDN technology is one of its advantages. SDN offers a more adaptable and dynamic method of managing network traffic by decoupling the control and data planes of the network. The suggested method makes use of SDN's power to enable fine-grained network traffic control and to make it simple and rapid to implement new policies.

Second, another advantage of the suggested strategy is the employment of GNNs. Effective traffic shaping requires the ability of GNNs to recognise intricate patterns in network traffic data. The suggested method makes predictions about how various policies would impact network performance by using GNNs to study the underlying patterns in the network traffic. The suggested method can predict outcomes more precisely than approaches that rely on easy heuristics or rule-based systems since it makes use of GNNs' strength.

The third and final advantage of the suggested strategy is the usage of MAB algorithms. Effective traffic shaping requires the system to be able to learn from and adjust to changing network conditions, which is made possible by MAB algorithms. The suggested technique is capable of learning from prior experiences and making real-time policy adjustments to ensure that network performance is optimised by utilising MAB algorithms.

6. Conclusion

In this paper, we proposed a novel approach for traffic shaping in Software Defined -IoT Networks using Graph Neural Networks and Multi-arm Bandit algorithm. We showed that our approach outperformed other state-of-the-art traffic shaping methods in terms of throughput, packet loss, and delay. Our experimental evaluation on three different datasets demonstrated the effectiveness of the proposed approach in detecting anomalous traffic patterns, handling heterogeneous data, and optimizing traffic shaping policies.

In conclusion, the proposed approach has shown promising results in traffic shaping, which is an important aspect of network management. Future work can explore the use of other deep learning architectures for traffic prediction, such as Transformers or Attention-based models. Additionally, investigating the performance of the proposed approach in more complex network topologies and

scaling it to larger networks could be another avenue for future research. Finally, evaluating the proposed approach’s performance under various attack scenarios could be beneficial in assessing its robustness and reliability in real-world settings.

7. Patents

This section is not mandatory, but may be added if there are patents resulting from the work reported in this manuscript.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.”, please turn to the [CRediT taxonomy](#) for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: “This research received no external funding” or “This research was funded by NAME OF FUNDER grant number XXX.” and “The APC was funded by XXX”. Check carefully that the details given are accurate and use the standard spelling of funding agency names at <https://search.crossref.org/funding>, any errors may affect your future funding.

Acknowledgments: In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

Conflicts of Interest: Declare conflicts of interest or state “The authors declare no conflict of interest.” Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Any role of the funders in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results must be declared in this section. If there is no role, please state “The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results”.

Sample Availability: Samples of the compounds ... are available from the authors.

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|--|--|
| MDPI | Multidisciplinary Digital Publishing Institute |
| DOAJ & Directory of open access journals | |
| TLA | Three letter acronym |
| LD | Linear dichroism |

Appendix A

Appendix A.1

The appendix is an optional section that can contain details and data supplemental to the main text—for example, explanations of experimental details that would disrupt the flow of the main text but nonetheless remain crucial to understanding and reproducing the research shown; figures of replicates for experiments of which representative data are shown in the main text can be added here if brief, or as Supplementary Data. Mathematical proofs of results not central to the paper can be added as an appendix.

Table A1. This is a table caption.

| Title 1 | Title 2 | Title 3 |
|---------|---------|---------|
| Entry 1 | Data | Data |
| Entry 2 | Data | Data |

Appendix B

All appendix sections must be cited in the main text. In the appendices, Figures, Tables, etc. should be labeled, starting with “A”—e.g., Figure A1, Figure A2, etc.

References

1. Rath, M.; Rout, U.P.; Pujari, N.; Nanda, S.K.; Panda, S.P. Congestion control mechanism for real time traffic in mobile adhoc networks. *Computer Communication, Networking and Internet Security: Proceedings of IC3T 2016*. Springer, 2017, pp. 149–156.
2. Xiong, S.; Sarwate, A.D.; Mandayam, N.B. Network traffic shaping for enhancing privacy in iot systems. *IEEE/ACM Transactions on Networking* **2022**, *30*, 1162–1177.
3. Shibata, N.; Kaneko, S.; Harada, R.; Honda, K.; Terada, J. Autonomous dynamic window shaping and rerouting for a service-converged layer-2 network with a time-aware shaper accommodating mobile fronthaul and IoT backhaul. *J. Opt. Commun. Netw.* **2021**, *13*, 108–115. doi:10.1364/JOCN.418549.
4. Yang, Z.; Ge, Z. On paradigm of industrial big data analytics: From evolution to revolution. *IEEE Transactions on Industrial Informatics* **2022**, *18*, 8373–8388.
5. Gelenbe, E.; Sigman, K. IoT traffic shaping and the massive access problem. *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 2732–2737.
6. Tam, P.; Song, I.; Kang, S.; Ros, S.; Kim, S. Graph Neural Networks for Intelligent Modelling in Network Management and Orchestration: A Survey on Communications. *Electronics* **2022**, *11*, 3371.
7. Yoo, Y.; Yang, G.; Kang, M.; Yoo, C. Adaptive Control Channel Traffic Shaping for Virtualized SDN in Clouds. 2020 IEEE 13th International Conference on Cloud Computing (CLOUD), 2020, pp. 22–24. doi:10.1109/CLOUD49709.2020.00013.
8. Hakak, S.; Gadekallu, T.R.; Maddikunta, P.K.R.; Ramu, S.P.; Parimala, M.; De Alwis, C.; Liyanage, M. Autonomous Vehicles in 5G and beyond: A Survey. *Vehicular Communications* **2022**, p. 100551.
9. Nasrallah, A.; Thyagaturu, A.S.; Alharbi, Z.; Wang, C.; Shao, X.; Reisslein, M.; Elbakoury, H. Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS). *IEEE Access* **2019**, *7*, 44165–44181. doi:10.1109/ACCESS.2019.2908613.
10. Lei, Z.; Panpan, W. Survey of traffic shaping and scheduling in time-sensitive network. *Microelectronics and Computer* **2022**, *39*, 46–53. doi:10.19304/J.ISSN1000-7180.2021.0629.
11. Thangamuthu, S.; Concer, N.; Cuijpers, P.J.L.; Lukkien, J.J. Analysis of Ethernet-switch traffic shapers for in-vehicle networking applications. 2015 Design, Automation and Test in Europe Conference and Exhibition (DATE), 2015, pp. 55–60. doi:10.7873/DATE.2015.0045.
12. Latha, V.P.; Reddy, N.S.; Babu, A.S. Optimizing Scalability and Availability of Cloud Based Software Services Using Modified Scale Rate Limiting Algorithm. *Theoretical Computer Science* **2023**, *943*, 230–240.
13. Pawar, K.; Mohite, B.; Kshirsagar, P. Analysis of Feature Selection Methods for UKM-IDS20 Dataset. In *Applied Computational Technologies: Proceedings of ICCET 2022*; Springer, 2022; pp. 461–467.
14. Yusof, N.N.M.; Sulaiman, N.S. Cyber attack detection dataset: A review. *Journal of Physics: Conference Series*. IOP Publishing, 2022, Vol. 2319, p. 012029.
15. Rahim, R.; Ahanger, A.S.; Khan, S.M.; Ma, F. Analysis of IDS using feature selection approach on NSL-KDD dataset, 2022.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.