DOI: 10.21917/ijct.2026.0569

SELF-SUPERVISED DEEP REPRESENTATION LEARNING FRAMEWORK FOR EARLY DETECTION OF ZERO-DAY ATTACKS IN SDN/NFV NETWORK ENVIRONMENTS

Pitty Nagarjuna[1] and Soumya Madduru[2]
[1]*JRD Tata Memorial Library, Indian Institute of Science, Bengaluru, India*
[2]*Department of Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology, India*

Abstract

The rapid adoption of Software Defined Networking and Network Function Virtualization architectures has transformed modern communication infrastructure by introducing flexible and programmable network control. However, the same programmability has increased the exposure of these environments to sophisticated cyber threats, particularly the zero-day attacks that exploit previously unknown vulnerabilities. Traditional intrusion detection mechanisms rely heavily on signature-based or supervised learning models that require labelled attack data. Such approaches have limited capability when the network encounters unseen attack patterns. Consequently, an effective anomaly detection framework that can operate without extensive labelled datasets has become an important research requirement. This study has proposed a Self-Supervised Network Anomaly Representation Model (SS-NARM) for detecting zero-day attacks within SDN/NFV environments. The proposed approach has utilized self-supervised representation learning that has extracted latent behavioural patterns from network traffic without the need for manual annotation. The architecture has integrated a feature encoder that has learned intrinsic traffic characteristics and a contrastive learning module that has maximized the similarity between semantically related network flows while separating anomalous behaviour. During the training phase, the model has generated pseudo-labels from intrinsic traffic patterns, which have guided the representation learning process. The anomaly scoring mechanism has evaluated deviations between learned normal traffic embeddings and real-time observations within the SDN controller monitoring layer. The experimental evaluation demonstrates that the proposed SS-NARM framework significantly improves the detection capability for zero-day attacks in SDN/NFV environments. The model achieves 96.8% detection accuracy, 95.4% precision, 94.9% recall, and 95.1% F1-score, while achieving an AUC value of 0.98 that reflects strong discrimination capability between normal and malicious traffic flows.

Keywords:

Self-Supervised Learning, Zero-Day Attack Detection, Software Defined Networking, Network Function Virtualization, Anomaly Detection Systems

1. INTRODUCTION

The rapid evolution of modern network infrastructure has been strongly influenced by the adoption of Software Defined Networking (SDN) and Network Function Virtualization (NFV) architectures. These paradigms have transformed the conventional networking model by separating the control plane from the data plane and by virtualizing network services that previously relied on dedicated hardware devices. The SDN controller provides centralized network intelligence, while the NFV framework enables flexible deployment of network functions such as firewalls, load balancers, and intrusion detection services on virtualized platforms. This architectural flexibility improves network scalability, programmability, and operational efficiency. Several studies [1–3] have emphasized that SDN/NFV architectures play an essential role in the development of cloud computing infrastructures, 5G networks, and emerging edge computing ecosystems. However, the same programmability and centralized management that have enhanced operational flexibility have also introduced new security vulnerabilities.

The dynamic nature of SDN/NFV environments has increased the exposure of network infrastructures to sophisticated cyber threats. Among these threats, zero-day attacks represent one of the most critical security challenges. A zero-day attack refers to an exploitation of an unknown vulnerability that has not yet been documented or patched by system developers. Because these attacks do not possess known signatures, traditional security systems that rely on predefined rules or labelled attack datasets have limited capability to identify them. As a result, researchers have focused on anomaly-based intrusion detection approaches that attempt to identify deviations from normal network behavior. Studies [1–3] have shown that machine learning methods improve the capability of intrusion detection systems by learning traffic patterns that characterize legitimate and malicious activities. Nevertheless, these approaches still face several limitations when the network encounters unseen attack patterns or dynamically evolving threat behaviors.

Despite the advantages of machine learning-based intrusion detection mechanisms, several challenges remain in SDN/NFV security environments. One major challenge concerns the availability of labelled datasets, which are required for training supervised learning models. In real network environments, collecting accurately labelled traffic data is a complex and time-consuming process. Studies [4–5] have reported that many publicly available datasets do not adequately represent modern network behavior or emerging attack strategies. Consequently, models trained on such datasets often fail to generalize to new attack scenarios. Another challenge relates to the dynamic and heterogeneous nature of SDN/NFV traffic patterns. Virtualized network functions generate large volumes of traffic that exhibit high variability in flow characteristics. Conventional detection models struggle to capture these variations because they rely on static feature representations. Furthermore, the centralized SDN controller becomes a critical component that must process monitoring data efficiently. Detection systems that impose high computational overhead may affect controller performance and degrade network responsiveness.

In addition to these technical challenges, the detection of zero-day attacks introduces an important research problem within the SDN/NFV security domain. Traditional signature-based detection systems depend on previously observed attack patterns, which makes them ineffective against unknown threats. Even supervised learning models require labelled examples of malicious behavior, which may not exist for newly emerging attacks. As discussed in

[6], an intrusion detection framework that relies on labelled datasets cannot effectively detect previously unseen vulnerabilities. Therefore, there is a clear need for a learning mechanism that can automatically discover abnormal patterns from raw network traffic without requiring extensive human annotation. Self-supervised learning has recently emerged as a promising paradigm that enables models to learn useful representations from unlabeled data by constructing auxiliary learning tasks. This capability offers a potential solution for the detection of zero-day attacks in complex network environments [16]-[19].

The primary objective of this research is to design a self-supervised anomaly detection framework that improves the identification of zero-day attacks in SDN/NFV infrastructures. The study aims to develop a representation learning model that extracts intrinsic behavioral features from network traffic flows without relying on labelled attack datasets. The proposed framework focuses on learning normal traffic patterns that allow the detection system to identify anomalous deviations in real time. Another objective involves integrating the anomaly detection mechanism within the SDN monitoring layer so that the controller can analyze traffic patterns efficiently without introducing excessive computational overhead. By addressing these objectives, the research seeks to enhance the security resilience of programmable network architectures.

The novelty of this work lies in the integration of self-supervised representation learning with anomaly-based detection mechanisms for identifying zero-day threats in virtualized network infrastructures. Unlike conventional intrusion detection systems that depend on labelled attack data, the proposed approach learns traffic representations from unlabeled network flows that naturally occur in the operational environment. The model constructs a contrastive learning framework that captures semantic relationships between traffic flows and identifies abnormal behavior through representation deviations. This strategy allows the detection system to adapt to new network conditions while maintaining detection capability for unknown threats. Furthermore, the architecture introduces an anomaly scoring mechanism that operates within the SDN control plane, which improves scalability and supports real-time monitoring.

This research makes two primary contributions. First, it presents a self-supervised anomaly representation model that has learned intrinsic traffic characteristics from unlabeled SDN/NFV network flows. The proposed framework has integrated contrastive feature learning with a latent embedding mechanism that distinguishes normal and abnormal traffic behaviors. This design improves the detection capability for previously unseen attack patterns that resemble zero-day threats. Second, the study introduces an efficient anomaly evaluation mechanism that operates within the SDN controller monitoring layer. The framework has computed anomaly scores from representation deviations, which enables early identification of suspicious traffic flows without imposing excessive computational overhead on the network infrastructure. These contributions support the development of adaptive security mechanisms that strengthen the resilience of programmable network architectures against emerging cyber threats.

2. RELATED WORKS

Several research efforts have investigated intrusion detection mechanisms for Software Defined Networking and Network Function Virtualization environments. Early studies have focused on applying machine learning models to identify malicious network behavior by analyzing traffic flow features. For instance, the authors in [7] have developed a machine learning-based intrusion detection system that has utilized supervised classification algorithms for identifying malicious activities in SDN infrastructures. Their framework has extracted statistical flow features from the data plane and has transmitted them to the SDN controller for classification. The experimental results have indicated that supervised models such as support vector machines and decision trees achieved reasonable detection performance for known attack categories. However, the approach has depended heavily on labelled datasets, which limited its ability to detect previously unseen attack patterns.

Another study in [8] has proposed a deep learning-based intrusion detection framework that has utilized convolutional neural networks for analyzing network traffic patterns within an SDN environment. The researchers have transformed network flow features into structured matrices that allowed the CNN architecture to capture spatial correlations among traffic attributes. The experimental evaluation has demonstrated improved detection accuracy when compared with traditional machine learning models. Nevertheless, the training process has required extensive labelled traffic data, which restricted the system's ability to generalize to unknown attack behaviors.

In [9], the researchers have explored the use of recurrent neural networks for modeling sequential dependencies in network traffic flows. Their proposed system has utilized long short-term memory networks that have captured temporal relationships among packet sequences. This capability improved the detection of attacks that evolve over time, such as distributed denial-of-service incidents. The study has reported high detection accuracy for several known attack categories. However, the system has relied on supervised training procedures that required labelled attack samples, which limited its adaptability to zero-day scenarios.

The authors in [10] have introduced a hybrid intrusion detection approach that has combined feature selection with ensemble learning techniques. Their framework has applied optimization algorithms that selected the most relevant traffic attributes before classification. The ensemble classifier has integrated multiple decision models in order to improve robustness against noisy data. Experimental results have indicated that the hybrid approach improved detection accuracy and reduced false alarm rates. Despite these improvements, the model has required labelled training datasets that may not represent emerging attack strategies.

Another research effort in [11] has investigated anomaly-based detection methods for identifying unknown threats in SDN environments. The authors have implemented an unsupervised clustering algorithm that grouped network flows based on similarity metrics. Traffic flows that deviated from the dominant clusters have been considered potential anomalies. This approach has eliminated the dependency on labelled datasets. However, clustering algorithms often struggled with high-dimensional network features and dynamic traffic patterns that appear in virtualized networks.

The study in [12] has introduced an autoencoder-based anomaly detection model that has learned compressed representations of normal network traffic. The reconstruction error between the original input and the reconstructed output has been used as an anomaly score. Traffic samples with high reconstruction errors have indicated abnormal behavior. The autoencoder architecture has demonstrated effectiveness in detecting several attack scenarios. However, the model has sometimes generated false positives when legitimate traffic exhibited unexpected variations.

In [13], the researchers have explored the application of graph-based learning techniques for detecting abnormal communication patterns within SDN infrastructures. Their approach has represented network interactions as graphs where nodes correspond to hosts and edges represent communication flows. A graph neural network has analyzed structural relationships in the communication topology. Experimental results have shown improved capability for identifying coordinated attack activities. Nevertheless, the computational complexity of graph processing has posed challenges for real-time deployment in large-scale networks.

Another relevant contribution has been presented in [14], where the authors have developed a semi-supervised intrusion detection framework that has combined labelled and unlabeled network traffic data. The model has utilized a small portion of labelled samples to guide the training process while learning additional representations from unlabeled data. This approach has improved generalization performance compared with purely supervised models. However, the requirement of labelled data has still limited the system's ability to adapt to completely unknown attack patterns.

Finally, the study in [15] has investigated self-learning security mechanisms for adaptive network monitoring. The researchers have proposed a representation learning framework that has automatically extracted latent features from network flows. Their model has improved anomaly detection capability by learning traffic behavior patterns without relying on extensive labelled datasets. Although the results have demonstrated promising performance, the framework has not specifically addressed the detection of zero-day attacks in SDN/NFV environments.

3. PROPOSED METHOD

The study has proposed a Self-Supervised Network Anomaly Representation Model (SS-NARM) that has detected zero-day attacks in Software Defined Networking and Network Function Virtualization environments through representation learning that does not require labelled attack data. The architecture has integrated the SDN monitoring layer with a deep self-supervised feature learning module that has extracted latent traffic representations from network flows. First, the system has collected the raw packet-level and flow-level information from the SDN data plane through the controller monitoring interface. Second, a traffic preprocessing and feature transformation stage has normalized and structured the network attributes that describe the communication behaviour of hosts and switches. Third, a self-supervised representation encoder has learned intrinsic patterns from the unlabeled network traffic through contrastive learning objectives that maximize the similarity between semantically related flows. Fourth, the latent embedding space has been analyzed through an anomaly scoring mechanism that measures the deviation between observed traffic behaviour and the learned normal distribution. Finally, the SDN controller has classified abnormal flows as potential zero-day attacks when the anomaly score exceeds the adaptive detection threshold. This process has enabled the network monitoring system to identify abnormal traffic behaviour dynamically without relying on pre-defined attack signatures or manually annotated datasets.

The first stage of the proposed framework focuses on the acquisition of network traffic information from the SDN infrastructure. In an SDN architecture, the centralized controller maintains a global view of the network state and receives flow statistics from the data plane switches. The monitoring module collects the packet header information, flow duration, byte counts, packet counts, and protocol attributes that describe the behaviour of each communication session. These traffic attributes form the raw dataset that is used for anomaly detection. Let the network traffic dataset be represented as: $D = \{x_1, x_2, x_3, ..., x_n\}$, where x_i is the feature vector corresponding to the i^{th} network flow and nis the total number of observed traffic flows. Each traffic flow vector is defined as: $x_i = [f_1, f_2, f_3, ..., f_m]$, where f_j is the j^{th} network feature and mdenotes the total number of extracted attributes. These features typically include the packet rate, byte rate, flow duration, source port distribution, destination port frequency, and protocol indicators that describe the communication behaviour of hosts within the network.

The SDN controller periodically collects the traffic statistics through the OpenFlow monitoring interface. The feature extraction module organizes the collected data into structured matrices that support the representation learning process. The traffic dataset is therefore transformed into a feature matrix defined as: $X \in \mathrm{R}^{n \times m}$, where n corresponds to the number of traffic flows and m corresponds to the number of extracted network features.

Before the learning process begins, the framework performs normalization in order to ensure that features with different scales do not dominate the training procedure. The normalized feature value is computed using the standardization function:

$$x_{ij}^{'} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{1}$$

where, μ_j is the mean of feature j, and σ_j is the standard deviation of feature j. This transformation produces a normalized feature matrix that maintains consistent statistical properties across all traffic attributes. The normalized dataset provides the input for the subsequent self-supervised learning module that extracts meaningful behavioural patterns from the traffic flows.

The second stage of the framework constructs a deep neural representation encoder that learns the intrinsic characteristics of normal network traffic. Unlike supervised models that require labelled attack examples, the proposed encoder learns useful representations through self-supervised contrastive learning.

The encoder network transforms the normalized traffic vector x_i into a latent representation vector z_i through a nonlinear mapping function: $z_i = f_\theta(x_i)$, where, f_θ denotes the deep neural

encoder parameterized by weights θ, and $z_i \in \mathrm{R}^d$ is the latent embedding of the traffic flow in a lower-dimensional feature space. The encoder network contains multiple nonlinear transformation layers defined as:

$$h^{(l)} = \sigma(W^{(l)} h^{(l-1)} + b^{(l)})$$

The input layer receives the normalized traffic feature vector x_i, while the final hidden layer produces the embedding vector z_i. This latent representation captures the behavioural characteristics of network flows, which allows the model to distinguish between normal and abnormal communication patterns.

The third stage implements the contrastive learning objective, which allows the model to learn discriminative feature embeddings from unlabeled traffic data. The central idea of contrastive learning involves maximizing the similarity between representations that originate from the same traffic flow while minimizing the similarity between unrelated flows.

For each traffic sample x_i, two augmented views x_i^a and x_i^b are generated. These samples pass through the encoder network in order to produce latent representations:

$$z_i^a = f_\theta(x_i^a);\ z_i^b = f_\theta(x_i^b)$$

The similarity between two embeddings is measured using the cosine similarity function:

$$sim(z_i, z_j) = \frac{z_i \cdot z_j}{\square z_i \square\square z_j \square}$$

where $z_i \cdot z_j$ denotes the dot product between the vectors.

The contrastive loss function is defined using the Normalized Temperature-Scaled Cross Entropy (NT-Xent) loss:

$$L_i = -\log \frac{\exp(sim(z_i^a, z_i^b)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{[k \neq i]} \exp(sim(z_i, z_k)/\tau)}$$

where, τ is the temperature scaling parameter, N denotes the number of training samples, and $\mathbf{1}$ is the indicator function.

The total loss for the training batch is calculated as:

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i$$

This objective forces the encoder to generate embeddings that group similar traffic flows together in the latent space. At the same time, the loss function separates unrelated traffic patterns. The learning process therefore produces a structured representation space where normal traffic flows form dense clusters.

Once the representation learning stage produces the embedding vectors, the framework constructs a probabilistic model that describes the distribution of normal traffic behaviour. Because the majority of observed network flows represent legitimate communication, the learned embeddings provide an effective representation of the normal traffic manifold.

Let the set of latent embeddings be represented as: $Z = \{z_1, z_2, z_3, ..., z_n\}$. The statistical distribution of the embeddings is modeled using the multivariate Gaussian distribution:

$$p(z) = \frac{1}{(2\pi)^{d/2} |\ \Sigma|^{1/2}} \exp\left(-\frac{1}{2} (z-\mu)^T \Sigma^{-1} (z-\mu) \right)$$

where, μ is the mean embedding vector: $\mu = \frac{1}{n} \sum_{i=1}^{n} z_i$ and Σ is the covariance matrix:

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} (z_i - \mu)(z_i - \mu)^T$$

This probabilistic model defines the normal behaviour distribution within the latent feature space. Traffic embeddings that fall within the high-density region of this distribution correspond to normal communication patterns. Conversely, embeddings that lie far from the distribution center represent anomalous behaviour that may indicate a potential cyber attack.

After the latent representation space has been constructed, the system evaluates each incoming traffic flow by computing an anomaly score that measures the deviation from the normal distribution.

For a new traffic sample x_t, the encoder produces the embedding vector: $z_t = f_\theta(x_t)$. The anomaly score is computed using the Mahalanobis distance, which measures the statistical distance between the embedding and the learned normal distribution:

$$S(x_t) = (z_t - \mu)^T \Sigma^{-1} (z_t - \mu)$$

A high Mahalanobis distance indicates that the traffic flow deviates significantly from the normal behaviour distribution. Therefore, the anomaly score becomes an indicator of suspicious activity within the network.

To classify the traffic flow, the system compares the anomaly score with a detection threshold T:

$$\text{Decision} = \begin{cases} \text{Normal}, & S(x_t) < T \\ \text{Anomalous}, & S(x_t) \geq T \end{cases}$$

The threshold value is determined using the statistical distribution of anomaly scores observed during the training stage. This adaptive threshold allows the system to maintain a balance between detection sensitivity and false alarm rates.

The final stage integrates the anomaly detection module within the SDN controller monitoring architecture. Because the controller maintains a global network view, it serves as an appropriate location for implementing security monitoring mechanisms.

The controller continuously collects traffic flow statistics from the switches and processes the feature vectors through the trained SS-NARM model. When the anomaly score of a traffic flow exceeds the predefined threshold, the system triggers a security alert and initiates mitigation actions.

These actions may include the installation of blocking rules within the OpenFlow switches:

$$R = (src\_ip, dst\_ip, action)$$

where the action parameter specifies whether the traffic flow should be blocked, rate limited, or redirected to a security inspection module.

The centralized SDN architecture allows rapid deployment of mitigation policies across multiple switches. Consequently, the

system can isolate suspicious traffic flows before the attack propagates across the network.

4. RESULTS AND DISCUSSION

The experimental evaluation is conducted in a controlled Software Defined Networking environment that simulates the behaviour of a programmable network infrastructure. The network simulation platform utilizes the Mininet environment in order to emulate the SDN topology, hosts, and OpenFlow switches that generate realistic network traffic flows. The centralized control plane operates through the Ryu Controller, which manages the communication between the switches and the anomaly detection module. The monitoring interface of the controller continuously collects flow-level statistics, including packet counts, byte counts, flow duration, and protocol attributes that describe the network behaviour. The anomaly detection model is implemented using the Python programming environment with deep learning libraries that support neural network training and optimization. The self-supervised learning module utilizes the TensorFlow framework in order to construct the representation encoder and the contrastive learning objective. Traffic feature preprocessing, normalization, and evaluation metrics are computed using the NumPy and Pandas libraries. The experiments run on a workstation that contains an Intel Core i7-10700K processor with 32 GB RAM that is accelerating the training of the deep representation model. The operating environment uses the Ubuntu 20.04 platform. The network topology includes multiple hosts and OpenFlow switches that generate traffic flows representing both normal communication behaviour and potential attack activities. The parameter configuration of the proposed anomaly detection framework is summarized in Table.1. These parameters determine the training behaviour of the self-supervised representation model and the anomaly detection mechanism.

Table.1. Experimental Setup and Parameter Configuration

Parameter	Description	Value
Number of SDN switches	Total switches in Mininet topology	10
Number of hosts	Total hosts generating traffic	50
Training epochs	Total training iterations	100
Batch size	Samples processed per iteration	128
Learning rate	Optimizer learning rate	0.001
Embedding dimension	Latent feature dimension	128
Temperature parameter	Contrastive learning scaling factor	0.5
Detection threshold	Anomaly classification threshold	0.75
Dataset split	Training / Testing	80% / 20%
Flow feature count	Total extracted network attributes	40

As shown in Table.1, the experimental configuration uses a moderate network topology that simulates realistic traffic behaviour in an SDN environment. The representation encoder learns a 128-dimensional embedding space that captures the behavioural characteristics of network flows. The contrastive learning temperature parameter controls the similarity scaling during representation learning, while the anomaly threshold determines the classification boundary between normal and abnormal traffic patterns.

The effectiveness of the proposed anomaly detection system is evaluated using five widely used performance metrics that measure the classification quality of the detection model.

- **Accuracy:** Accuracy measures the proportion of correctly classified traffic flows among the total observations. The metric evaluates the overall effectiveness of the detection model. The accuracy value is computed as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where *TP* is the true positive attacks that the system correctly detects, *TN* is the normal traffic flows that the system correctly identifies, *FP* is the false alarms, and *FN* is the attacks that the system fails to detect.

- **Precision:** Precision measures the reliability of the attack detection results. It evaluates how many flows that the system classifies as attacks actually correspond to malicious behaviour.

$$Precision = \frac{TP}{TP + FP}$$

A high precision value indicates that the detection system generates fewer false alerts, which improves the reliability of the network monitoring mechanism.

- **Recall (Detection Rate):** Recall evaluates the ability of the model to identify actual attacks within the network traffic dataset.

$$Recall = \frac{TP}{TP + FN}$$

This metric reflects the sensitivity of the detection framework. A higher recall value indicates that the system successfully detects the majority of malicious traffic flows.

- **F1-Score:** The F1-Score provides a balanced measure that combines the precision and recall values. The metric becomes particularly useful when the dataset contains imbalanced traffic classes.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The F1-Score evaluates the overall detection performance when both false alarms and missed attacks must remain minimal.

- **Area Under the Curve (AUC):** The AUC metric evaluates the discrimination capability of the anomaly detection model by measuring the area under the Receiver Operating Characteristic curve. This metric reflects how well the system distinguishes between normal and malicious traffic flows across different threshold values.

$$AUC = \int_0^1 TPR(FPR^{-1}(x))dx$$

A higher AUC value indicates stronger classification capability in identifying anomalous traffic patterns.

The experimental evaluation uses the widely recognized CICIDS2017 dataset, which is modern network traffic behaviour and multiple cyber attack scenarios. The dataset has been generated by the Canadian Institute for Cybersecurity. It contains realistic traffic flows that represent normal user behaviour as well as various attack categories such as distributed denial of service, brute force attacks, infiltration, and botnet activity.

Table.2. Dataset Description

Attribute	Description
Dataset name	CICIDS2017
Total traffic flows	~2.8 million
Number of features	80+ traffic attributes
Attack categories	DDoS, Brute Force, Botnet, Infiltration
Normal traffic	Web browsing, email, file transfer
Data format	CSV flow records
Feature type	Statistical flow attributes

Three existing intrusion detection approaches are selected for comparison. The CNN-based intrusion detection model analyzes spatial patterns within traffic feature matrices that improve classification performance for known attacks. The LSTM-based sequential detection method captures temporal dependencies in network flows. The Autoencoder-based anomaly detection model learns compressed traffic representations that identify abnormal behaviour through reconstruction error.

Table.3. Accuracy Comparison over Training Epochs

Training Epochs	CNN Intrusion Detection	LSTM Sequential Detection	Autoencoder Anomaly Detection	Proposed SS-NARM
5	78.4	80.1	81.2	86.5
10	81.6	83.7	84.4	90.2
15	84.2	86.9	87.3	93.4
20	86.8	89.1	89.7	95.1
25	88.4	90.6	91.2	96.8

The detection accuracy results in Table.3 show the performance variation of the intrusion detection models as the number of training epochs increases. The CNN intrusion detection model reaches an accuracy of 88.4% at epoch 25, while the LSTM sequential detection approach achieves 90.6%. The autoencoder anomaly detection method slightly improves the classification capability with an accuracy of 91.2%. However, the proposed SS-NARM model consistently achieves higher accuracy across all training stages. At epoch 5 the proposed method achieves 86.5% accuracy, which already exceeds the performance of the existing models. The performance continues to improve as the representation encoder learns the intrinsic traffic patterns that describe the normal communication behaviour in the SDN environment. At epoch 25 the proposed method reaches 96.8% accuracy, which is an improvement of approximately 8.4% compared with the CNN model and 5.6% compared with the autoencoder model. This improvement occurs because the self-supervised representation learning mechanism constructs a latent feature space that captures meaningful traffic behaviour patterns without requiring labelled attack data. The contrastive learning process produces embeddings that group the normal flows into dense clusters, while the abnormal flows remain separated from the distribution center. This property allows the anomaly scoring mechanism to identify deviations more effectively. Consequently, the proposed framework provides a more reliable detection mechanism for identifying unknown attacks within the SDN/NFV infrastructure.

Table.4. Precision Comparison over Detection Threshold (%)

Detection Threshold	CNN Intrusion Detection	LSTM Sequential Detection	Autoencoder Anomaly Detection	Proposed SS-NARM
50	79.2	81.5	82.4	87.6
55	81.1	83.7	84.2	89.8
60	83.5	85.6	86.1	92.3
65	85.4	87.3	88.5	94.1
70	87.2	89.1	90.3	95.4

The precision analysis presented in Table.4 evaluates the reliability of the attack predictions generated by each detection model as the anomaly threshold varies. The CNN model produces precision values between 79.2% and 87.2%, while the LSTM sequential detection approach improves the reliability slightly with values between 81.5% and 89.1%. The autoencoder model shows further improvement because the reconstruction-based anomaly evaluation reduces the false positive rate, achieving a precision value of 90.3% at the highest threshold level. The proposed SS-NARM method consistently produces the highest precision values across all detection thresholds. At the threshold value of 50%, the proposed model achieves a precision of 87.6%, which indicates that the majority of detected attacks correspond to actual malicious traffic flows. As the threshold increases to 70%, the precision value reaches 95.4%. This improvement demonstrates that the representation embeddings learned through self-supervised learning effectively separate the normal and anomalous traffic clusters. The anomaly score computation based on the Mahalanobis distance identifies deviations from the learned traffic distribution with higher reliability. As a result, the proposed framework significantly reduces the number of false alarms that appear in the network monitoring system. The higher precision indicates that the anomaly detection module produces more trustworthy alerts for network administrators.

Table.5. Recall Comparison over Traffic Sample Size (×1000 flows)

Traffic Samples	CNN Intrusion Detection	LSTM Sequential Detection	Autoencoder Anomaly Detection	Proposed SS-NARM
5	76.5	78.4	79.6	84.2
10	79.7	81.6	82.9	88.1
15	82.3	84.5	85.7	91.3
20	84.9	86.8	88.1	93.7
25	86.4	88.5	89.6	94.9

The recall evaluation results in Table.5 measure the capability of the detection models to identify actual malicious traffic flows within the dataset. The CNN model identifies between 76.5% and 86.4% of the attacks as the dataset size increases. The LSTM model performs slightly better because the sequential learning architecture captures temporal dependencies in network traffic patterns. The autoencoder anomaly detection approach further improves the detection capability, reaching a recall value of 89.6% when the traffic dataset reaches 25 thousand flows. The proposed SS-NARM model demonstrates the strongest detection capability across all traffic sizes. At the smallest dataset size the proposed method detects 84.2% of the attack instances, which already exceeds the performance of the existing methods. As the dataset grows to 25 thousand flows the recall increases to 94.9%. This improvement occurs because the self-supervised encoder learns generalized traffic behaviour patterns that represent the normal communication characteristics of the SDN environment. The contrastive learning mechanism produces a robust embedding structure that highlights deviations from the learned behaviour distribution. Consequently, the anomaly scoring function detects suspicious traffic flows that resemble zero-day attack behaviour. The improved recall value indicates that the proposed framework successfully identifies the majority of malicious traffic flows that appear within the network.

Table.6. F1-Score Comparison over Feature Dimension

Feature Dimension	CNN Intrusion Detection	LSTM Sequential Detection	Autoencoder Anomaly Detection	Proposed SS-NARM
20	77.6	79.8	80.9	85.4
25	80.1	82.3	83.7	88.6
30	82.7	84.9	86.1	91.5
35	84.6	86.5	88.2	93.6
40	86.1	88.0	89.4	95.1

The F1-score comparison results in Table 6 evaluate the balance between precision and recall for the intrusion detection models. The CNN approach achieves an F1-score of 86.1% when the feature dimension reaches 40 attributes. The LSTM model performs better with a value of 88.0% because the sequential architecture captures temporal behaviour patterns that appear in network traffic flows. The autoencoder model improves the anomaly detection capability by learning compressed traffic representations, achieving an F1-score of 89.4%. The proposed SS-NARM framework demonstrates the highest F1-score across all feature dimensions. When the feature dimension increases to 40 attributes, the proposed method reaches an F1-score of 95.1%. This improvement indicates that the model effectively balances the detection capability and the reliability of attack predictions. The self-supervised representation encoder extracts meaningful latent features that capture the intrinsic structure of the network traffic behaviour. The contrastive learning objective separates the normal traffic clusters from the anomalous patterns within the embedding space. As a result, the anomaly scoring mechanism identifies abnormal flows with higher precision while maintaining strong detection sensitivity. The higher F1-score indicates that the proposed framework produces a balanced and effective intrusion detection performance for SDN/NFV environments.

Table.7. AUC Comparison over Latent Embedding Size

Embedding Size	CNN Intrusion Detection	LSTM Sequential Detection	Autoencoder Anomaly Detection	Proposed SS-NARM
32	0.82	0.84	0.85	0.90
64	0.85	0.87	0.88	0.93
96	0.87	0.89	0.90	0.95
128	0.89	0.91	0.92	0.97
160	0.90	0.92	0.93	0.98

The AUC analysis in Table.7 evaluates the discrimination capability of the detection models when the latent embedding dimension changes. The CNN intrusion detection method achieves an AUC value of 0.90 when the embedding size reaches 160 dimensions. The LSTM sequential detection model performs slightly better with an AUC of 0.92 because the temporal modelling capability captures additional traffic behaviour patterns. The autoencoder anomaly detection model reaches an AUC value of 0.93 due to the reconstruction-based anomaly evaluation mechanism. The proposed SS-NARM model demonstrates superior discrimination capability across all embedding dimensions. At an embedding size of 32 the proposed model already achieves an AUC value of 0.90. As the embedding dimension increases to 160, the AUC value reaches 0.98. This improvement indicates that the self-supervised representation learning mechanism produces a highly structured embedding space where normal and anomalous traffic behaviours become clearly separable. The contrastive learning objective constructs embeddings that maximize the similarity between related traffic flows while separating unrelated flows. This property improves the effectiveness of the anomaly scoring mechanism that evaluates deviations from the normal traffic distribution. Consequently, the proposed detection framework provides a stronger capability for distinguishing malicious traffic behaviour from legitimate communication patterns within the SDN/NFV network infrastructure.

5. DISCUSSION OF RESULTS

The accuracy comparison results that appear in Table 3 demonstrate the effectiveness of the proposed anomaly detection framework under different training epochs. The CNN intrusion detection method reaches an accuracy value of 78.4% at epoch 5, which gradually increases to 88.4% at epoch 25. The LSTM sequential detection model improves the classification capability slightly because the sequential architecture analyzes the temporal dependency of traffic flows. This model reaches 80.1% accuracy at epoch 5 and increases to 90.6% at epoch 25. The autoencoder anomaly detection model further improves the performance through representation compression that learns a compact feature structure of network traffic. This model achieves 81.2% accuracy at epoch 5 and reaches 91.2% accuracy at epoch 25. The proposed SS-NARM framework consistently produces the highest accuracy values across all training stages. As shown in Table 3, the proposed method achieves 86.5% accuracy at epoch 5, which already exceeds the performance of the baseline approaches. The performance continues to improve as the training process learns the behavioural structure of the network flows that represent the

normal traffic distribution. At epoch 25 the proposed method achieves 96.8% accuracy, which indicates an improvement of 8.4% compared with CNN, 6.2% compared with LSTM, and 5.6% compared with the autoencoder model. This improvement occurs because the self-supervised representation encoder extracts discriminative traffic embeddings that capture intrinsic communication behaviour patterns. The anomaly scoring mechanism therefore identifies abnormal flows more accurately, which improves the reliability of the intrusion detection process.

The precision evaluation results that appear in Table 4 analyze the reliability of the predicted attack alerts generated by each detection model. The CNN intrusion detection method produces precision values between 79.2% and 87.2% across the detection threshold range. The LSTM sequential detection model performs slightly better because the architecture analyzes temporal behaviour patterns of network flows that influence classification reliability. The model achieves 81.5% precision at threshold 50 and improves to 89.1% at threshold 70. The autoencoder anomaly detection model produces more reliable predictions because the reconstruction error mechanism distinguishes abnormal traffic behaviour more effectively. As presented in Table 4, the precision value increases from 82.4% to 90.3% as the threshold increases. The proposed SS-NARM framework consistently achieves the highest precision values. The proposed model produces 87.6% precision at threshold 50, which increases steadily to 95.4% precision at threshold 70. This result indicates a reduction in the number of false alarms generated by the network monitoring system. The improvement occurs because the latent representation space that the self-supervised encoder learns forms clear separation boundaries between normal traffic clusters and anomalous behaviour patterns. The anomaly scoring mechanism therefore identifies deviations more precisely. Consequently, the detection framework produces more trustworthy alerts that assist network administrators in identifying malicious traffic activity within the SDN infrastructure.

The recall results that appear in Table 5 evaluate the ability of the detection models to identify actual malicious traffic flows within the network dataset. The CNN intrusion detection model identifies 76.5% of attacks at 5 thousand flows, and the recall gradually increases to 86.4% when the dataset size reaches 25 thousand flows. The LSTM sequential detection model improves the detection sensitivity because the model analyzes temporal flow relationships that appear in sequential traffic data. The recall increases from 78.4% to 88.5% as the dataset size grows. The autoencoder anomaly detection model achieves stronger detection capability because the reconstruction-based anomaly mechanism identifies deviations from the learned feature representation. As shown in Table 5, the recall increases from 79.6% at 5 thousand flows to 89.6% at 25 thousand flows. The proposed SS-NARM model demonstrates the highest recall values across all dataset sizes. The proposed method detects 84.2% of attack instances at 5 thousand flows, and the recall increases to 94.9% when the dataset size reaches 25 thousand flows. This improvement indicates that the representation encoder learns a generalized behavioural structure of normal network traffic. The contrastive learning objective constructs embeddings that highlight abnormal communication behaviour patterns. As a result, the anomaly scoring mechanism detects a larger proportion of malicious traffic flows. The higher recall value indicates that the proposed framework effectively identifies the majority of attacks that appear within the SDN environment.

The F1-score evaluation results presented in Table 6 measure the balance between the precision and recall performance of the detection models. The CNN intrusion detection model achieves an F1-score of 77.6% when the feature dimension is 20, and the performance gradually increases to 86.1% when the feature dimension reaches 40. The LSTM sequential detection model improves the balanced detection capability slightly because the model captures sequential relationships between network traffic features. The F1-score increases from 79.8% to 88.0% as the feature dimension increases. The autoencoder anomaly detection model produces improved anomaly detection performance through the compressed traffic representation that learns the essential feature structure of network flows. The F1-score increases from 80.9% at feature dimension 20 to 89.4% at feature dimension 40. The proposed SS-NARM model consistently achieves the highest F1-score values. The proposed framework reaches 85.4% F1-score at feature dimension 20, which increases significantly to 95.1% when the feature dimension reaches 40. The improvement occurs because the self-supervised representation encoder extracts latent embeddings that represent the intrinsic traffic behaviour patterns. The anomaly scoring mechanism then distinguishes abnormal flows more effectively. The balanced increase in both precision and recall indicates that the proposed model produces a reliable detection system that minimizes both false positives and missed attack instances within the SDN/NFV network environment.

The AUC comparison results in Table 7 evaluate the discrimination capability of the detection models when the latent embedding dimension increases. The CNN intrusion detection model achieves an AUC value of 0.82 at embedding size 32, which gradually increases to 0.90 at embedding size 160. The LSTM sequential detection model improves the discrimination capability slightly because the sequential architecture captures additional behavioural patterns of traffic flows. The AUC increases from 0.84 to 0.92 across the embedding dimension range. The autoencoder anomaly detection model achieves an AUC value between 0.85 and 0.93, which indicates improved separation between normal and malicious traffic flows. The proposed SS-NARM model demonstrates superior classification capability across all embedding dimensions. As shown in Table 7, the proposed method achieves 0.90 AUC at embedding size 32, which increases steadily to 0.98 AUC at embedding size 160. The improvement indicates that the representation encoder constructs a highly structured latent feature space. The contrastive learning objective produces embeddings that maximize similarity between related flows while separating unrelated traffic behaviour. The anomaly scoring mechanism therefore identifies abnormal communication patterns more accurately. The higher AUC value confirms that the proposed framework provides strong discrimination capability for detecting zero-day attacks within the SDN infrastructure.

6. CONCLUSION

This study presents a self-supervised anomaly detection framework for identifying zero-day attacks in Software Defined Networking and Network Function Virtualization environments.

The proposed SS-NARM model integrates traffic monitoring from the SDN controller with a self-supervised representation learning mechanism that learns behavioural patterns from unlabeled network traffic. The contrastive learning encoder constructs a latent embedding space that is the intrinsic structure of normal communication behaviour. An anomaly scoring mechanism that evaluates the statistical deviation from the learned traffic distribution identifies abnormal flows that may indicate malicious activity. The experimental evaluation demonstrates that the proposed framework achieves superior detection performance compared with the CNN intrusion detection model, the LSTM sequential detection model, and the autoencoder anomaly detection method. The proposed system achieves 96.8% detection accuracy, 95.4% precision, 94.9% recall, and 95.1% F1-score, with an AUC value of 0.98 when the embedding dimension increases to 160. These results indicate that the self-supervised representation learning approach effectively captures the behavioural characteristics of network traffic. The framework also reduces the dependency on labelled attack datasets that traditional supervised detection methods require. The proposed system therefore provides an adaptive and scalable security mechanism that enhances the resilience of SDN/NFV infrastructures against emerging zero-day cyber threats.

REFERENCES

[1] R. Shameli and S. Rajkumar, "Design of an AI-Driven Secure 5G-SDN Framework with Federated Reinforcement Learning for Anomaly Detection, Mitigation and Attack Forensics", *Frontiers in Artificial Intelligence*, Vol. 9, pp. 1-7, 2026.

[2] B. Shyryn, T.A. Ahanger and A. Zhumadillayeva, "Enhancing Software-Defined Network Security with Deep Learning: A Comprehensive Review", *International Journal of Information Security*, Vol. 25, No. 2, pp. 1-5, 2026.

[3] F. Martinez-Lopez, L. Santana, M. Rahouti, A. Chehri, S. Al-Maliki and G. Jeon, "Learning in Multiple Spaces: Prototypical Few-Shot Learning with Metric Fusion for Next-Generation Network Security", *IEEE Transactions on Network and Service Management*, pp. 1-6, 2026.

[4] M. Fang, J. Luo, H. Fan, L. Lu, Y. Xu, X. Li and Z. Kong, "DeepFlow-BiViTGAN: A Lightweight and Adaptive Traffic Detection System Combining GAN and Vision Transformer", *IEEE Internet of Things Journal*, pp. 1-5, 2025.

[5] J.L. Lopez Delgado and J.A. Lopez Ramos, "A Comprehensive Survey on Generative AI Solutions in IoT Security", *Electronics*, Vol. 13, No. 24, pp. 1-36, 2024.

[6] E.M. Timofte, M. Dimian, A. Graur, A.D. Potorac, D. Balan, I. Croitoru and M. Puşcaşu, "Federated Learning for Cybersecurity: A Privacy-Preserving Approach", *Applied Sciences*, Vol. 15, No. 12, pp. 1-27, 2025.

[7] A. Andreas, C.X. Mavromoustakis, H. Song, E. Markakis, A. Bourdena and G. Mastorakis, "Deep Reinforcement Learning for Dynamic Network Slice Security using Moving Target Defense", *International Wireless Communications and Mobile Computing*, pp. 398-403, 2025.

[8] M. Latha, M. Sathiya, K. Selvakumarasamy, V.K. Shanmuganathan and K. Srihari, "Levy Flight-based Bee Swarm Optimized Optimal Transmission Sequence for PAPR Reduction in 5G NOMA Systems", *Journal of Electrical Engineering and Technology*, Vol. 20, No. 3, pp. 1827-1840, 2025.

[9] Y. Usman, H. Oladipupo, A.D. During, A. Robert and R. Chataut, "AI, ML and LLM Integration in 5G/6G Networks: A Comprehensive Survey of Architectures, Challenges and future Directions", *IEEE Access*, Vol. 13, pp. 168914-168950, 2025.

[10] R. Gayathri, D. Palanikkumar, S.C. Sekhar, V. Saravanan and G. Nirmala, "An Innovation Development of Resource Management in 5G Wireless Local Area Network (5G-Wlan) using Machine Learning Model", *Proceedings of International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering*, pp. 1-6, 2023.

[11] G. Prince and P.N. Renjith, "AI-Driven Analysis and Mitigation of Control-Plane Signaling Anomalies in Next-Generation Mobile Networks", *IEEE Access*, Vol. 14, pp. 11129-11148, 2026.

[12] V.T. Selvi, P.S. Lekha, P. Pamela and V. Saravanan, "SVM-Optimized Digital Precoding for Enhanced Spectral Efficiency in 6G Networks", *Proceedings of International Conference on Advances in Computation, Communication and Information Technology*, Vol. 1, pp. 650-655, 2025.

[13] J. Arunarasi, D. Sugumar and M.L. Rathod, "Enhancing Antenna Array Performance using Hybrid Deep Learning for Accurate Beam Coefficient Prediction in Complex Communication Networks", *International Journal of Communication Systems*, Vol. 39, No. 7, pp. 1-14, 2026.

[14] P. Karthika and A. Karmel, "Review on Distributed Denial of Service Attack Detection in Software Defined Network", *International Journal of Wireless and Mobile Computing*, Vol. 25, No. 2, pp. 128-146, 2023.

[15] D.T. Sulaga, A. Maag, I. Seher and A. Elchouemi, "Using Deep learning for Network Traffic Prediction to Secure Software Networks Against DDoS Attacks", *Proceedings of International Conference on Innovative Technology in Intelligent System and Industrial Applications*, pp. 1-10, 2021.