JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Autoencoder-Based Unsupervised Network Traffic Anomaly Detection on Pseudo Time-Series Data

[1]G. Likith Praveen Apparao, [2]Dr. M. Sailaja

[1]MTech Post graduate Student, [2]Professor, Department of ECE,
Computer and Communication Engineering, Department of ECE,
[1]University College of Engineering Kakinada (A), JNTU Kakinada, Kakinada, India

Abstract: Network anomaly detection is a critical task for ensuring cybersecurity in dynamic traffic environments. Traditional methods, such as correlation-based or statistical rule-driven detection, often depend on labelled data or fixed thresholds, which limit adaptability and scalability. These approaches struggle with non-Gaussian patterns, threshold rigidity, and high false alarms in unlabelled or real-time traffic. To overcome these challenges, we propose an Autoencoder-based unsupervised learning model that detects anomalies through reconstruction error patterns in pseudo time-series IP traffic. Our method adapts dynamically using percentile-based thresholding and error smoothing, removing the need for predefined thresholds or labelled anomalies, and effectively detecting both service failure and traffic surge anomalies. The model demonstrated exceptional performance across both anomaly types. It consistently achieved 99 to 100 percent accuracy, precision, recall, and F1-scores as test sizes scaled from 50 to 100000 samples. Even at large test sizes, the model maintained reliable detection with minimal false positives. High recall ensured nearly all injected anomalies were detected, while high precision minimized false alarms across all scales. Furthermore, we conducted a comparative evaluation against one traditional method's Fast xFlow Proxy, which used correlation analysis and dynamic rule thresholds. Our Autoencoder model significantly outperformed it in detection accuracy, scalability, and generalization to different anomaly types. On average, the proposed Autoencoder framework achieved 99.91 percent accuracy in detecting service failure anomalies caused by traffic drops and 99.84 percent accuracy in detecting sudden traffic surges due to DDoS attacks. These results, obtained across multiple anomaly ratios and seven different test dataset sizes, confirm that our model consistently outperforms the Fast xFlow Proxy method in terms of accuracy, scalability, and robustness, establishing it as a highly effective solution for real-time unlabelled anomaly detection in modern cybersecurity systems.

IndexTerms **- Autoencoder, Anomaly Detection, Network Traffic, Time Series, DDoS attacks, Reconstruction Error, Unsupervised Learning, EWMA, Percentile Thresholding, IP Monitoring.**

I. INTRODUCTION

The increasing complexity and volume of IP network traffic have heightened the risk of anomalies such as service failures and Distributed Denial-of-Service (DDoS) attacks, making anomaly detection essential for ensuring cybersecurity and service reliability. Traditional detection approaches, including statistical models, rule-based systems, and supervised learning, depend on labelled datasets or fixed thresholds, limiting their adaptability to dynamic, unlabelled traffic. Correlation-based methods such as the Fast xFlow Proxy [7] achieve moderate accuracy but degrade under noisy conditions, while hybrid deep learning techniques [1], [4], [5] provide higher detection performance at the cost of greater complexity and reduced interpretability.

To address these limitations, we propose an unsupervised Autoencoder-based anomaly detection framework that detects anomalies through reconstruction error patterns in 24-hour pseudo time-series network flows. The model is trained exclusively on normal traffic, enabling it to learn baseline communication behaviour without labelled anomalies. During inference, anomalies are flagged using percentile-based dynamic thresholding combined with Exponential Weighted Moving Average (EWMA) smoothing, ensuring robustness against transient fluctuations and eliminating the need for fixed thresholds.

Key contributions of this work include:

1. A reconstruction-error-based Autoencoder framework with adaptive thresholding for unsupervised anomaly detection.
2. A systematic evaluation on two anomaly scenarios that is service failures, which means zero-drops in network with Gaussian noise and traffic surges also known as DDoS-like bursts were tested across seven dataset sizes from 50 to 100,000 samples.
3. A comparative study with the Fast xFlow Proxy method, demonstrating that the proposed approach achieves over 99% accuracy, precision, recall, and F1-score, outperforming the baseline while remaining scalable and lightweight.

The remainder of this paper is organized as follows: Section II presents Literature review, Section III describes the Autoencoder design, Section IV details the methodology, Section V reports results and comparative analysis, and Section VI concludes with future research directions.

II. Literature Review

Wei et al. [1] proposed a reconstruction-based LSTM-Autoencoder (LSTM-AE) for detecting anomaly-based DDoS attacks using multivariate time-series data. The model was trained on benign traffic in an unsupervised manner, combining LSTM units for temporal dependencies with an Autoencoder for reconstruction error–based anomaly detection. Evaluated on the CICDDoS2019 dataset, it achieved over 99% accuracy for DNS, LDAP, and SNMP attacks. The authors concluded that the LSTM-AE is robust for DDoS detection and outperforms traditional methods. In contrast, our work adopts a simpler Autoencoder with dynamic percentile-based thresholding and EWMA smoothing for scalable and interpretable anomaly detection.

Giri et al. [2] proposed a hybrid intrusion detection approach using autoencoders to reduce false positives in network intrusion detection systems (NIDS). Using the NSL-KDD dataset, they applied feature selection with Random Forest, followed by an Autoencoder to classify anomalies based on reconstruction error thresholds. The model achieved 90.32% accuracy, 88.13% precision, 95.91% recall, and 91.86% F1-score, outperforming several baseline methods. Unlike their supervised feature selection approach, our method is entirely unsupervised and achieves higher accuracy with dynamic thresholding on large-scale test datasets.

Manjunatha et al. [3] developed a sparse deep denoising autoencoder (SDDAE)-based intrusion detection system (IDS) for dimensionality reduction and classification. Evaluated on KDDCUP99, NSL-KDD, UNSW-NB15, and NMITIDS datasets, the framework achieved over 96% accuracy, with 95.98% overall accuracy and 98.36% detection rate for U2R and R2L attacks on NSL-KDD. The model incorporated dropout, regularization, and SVM for enhanced classification. In contrast, our approach focuses on a single Autoencoder with percentile-based thresholding, avoiding added classification layers while achieving near-perfect detection performance.

Narmadha and Balaji [4] presented a hybrid deep learning model that combines an Autoencoder with an LSTM network optimized using Particle Swarm Optimization (PSO) for anomaly detection. Evaluated on KDD99, it achieved 99.89% accuracy, 99.91% precision, 99.86% recall, and 99.86% F1-score, outperforming CNN-LSTM and other methods. However, their approach increases complexity due to sequential modelling and metaheuristic tuning. Our method instead uses a lightweight Autoencoder with dynamic thresholding, ensuring scalability without added complexity.

Najafi et al. [5] introduced a hybrid anomaly detection model combining a classical Autoencoder with a Transformer-inspired attention mechanism for online time-series anomaly detection. The Autoencoder extracted local features, while the attention layer captured long-term dependencies efficiently. Evaluated on six NAB datasets, it achieved up to 100% F1-score, outperforming VAE, LSTM-AD, and ARMA models. Although highly effective, the model's attention layers add architectural complexity. Our work retains a simpler Autoencoder design while ensuring scalability and robustness across diverse anomaly severities and dataset sizes.

Zhou et al. [6] proposed a network anomaly detection model combining feature grouping with multiple identical Autoencoders (multi-AEs). The grouped features were processed by separate Autoencoders, and anomaly scores were aggregated using AUC-based weights. Tested on UNSW-NB15 and CICIDS2017 datasets, it improved accuracy by 12.04% and 10.52% compared to a single Autoencoder but added computational overhead. In contrast, our work uses a single Autoencoder with percentile-based thresholding and EWMA smoothing, achieving similar accuracy while remaining simpler and more scalable.

Kamamura et al. [7] proposed the Fast xFlow Proxy, which decomposes encapsulated traffic into fine-grained flows and uses correlation-based analysis with dynamic thresholding for anomaly detection. Tested on synthetic traffic, it achieved nearly 100% accuracy for service failures but only 80–90% accuracy under noise. While efficient for high-resolution traffic analysis, it lacks deep learning-based modelling and scalability. Our work outperforms this approach by using a reconstruction-error-based Autoencoder that achieves over 99% accuracy across all anomaly intensities and dataset sizes.

Wong et al. [8] proposed AER: Autoencoder with Regression, a hybrid model combining an Autoencoder with an LSTM-based regressor for time-series anomaly detection. The model jointly minimized reconstruction and prediction losses, integrating masking and bi-directional scoring to reduce false positives. Experiments on 12 public datasets showed a 23.5% F1-score improvement over ARIMA while maintaining runtime efficiency. In contrast, our method avoids added regression components and relies on percentile thresholding with EWMA smoothing for highly accurate detection.

Torabi et al. [9] developed a vector-based Autoencoder anomaly detection framework for cloud networks using the CIDDS-001 dataset. The method computed per-feature reconstruction errors and used feature-specific thresholds for classification, achieving 100% accuracy and recall on most classes while improving anomaly granularity. Unlike this complex per-feature approach, our work employs a simpler Autoencoder with global percentile thresholding for efficient and scalable detection.

Berahmand et al. [10] presented a survey on Autoencoder architectures and applications, covering variants such as sparse, convolutional, variational, and graph-based AEs. The paper summarized methodologies for anomaly detection, feature extraction, and generative modelling, discussing challenges in hyperparameter tuning and robustness. Our approach aligns with the survey's emphasis on interpretability and scalability while applying a sparse Autoencoder for practical anomaly detection.

Alaghbari et al. [11] proposed a deep Autoencoder-based integrated model for anomaly detection and feature extraction in IoT networks. The model was trained on normal traffic for reconstruction-error-based anomaly detection while also providing low-dimensional features for multi-class attack classification. It outperformed OC-SVM and isolation forest on N-BaIoT and

MQTTset datasets. Our approach similarly trains on clean traffic but focuses solely on scalable anomaly detection across multiple dataset sizes.

Salinas Monroy et al. [12] developed a privacy-preserving Autoencoder model for detecting malicious DNS-over-HTTPS (DoH) traffic. The Autoencoder was trained only on benign DoH traffic, and anomalies were flagged by high reconstruction error. Experiments on realistic datasets showed superior median F1-scores ~99% compared to isolation forests, LOF, and VAEs. Our method differs by targeting broader anomaly types and ensuring scalability using percentile thresholding and EWMA smoothing.

Our Studies have demonstrated that Autoencoder-based models and their hybrid variants achieve high detection performance across various datasets. However, many approaches rely on complex architectures like LSTM, attention layers, multi-AEs or manual threshold tuning, which limit scalability and interpretability. In contrast, our work uses a lightweight reconstruction-

error-based Autoencoder with dynamic percentile thresholding and EWMA smoothing, achieving over 99% accuracy, precision, and recall while being computationally efficient and scalable across diverse anomaly scenarios and dataset sizes.

III. AUTOENCODER MODEL DESIGN

3.1 Block Architecture Overview

The architecture of the proposed Autoencoder model is illustrated in Fig. 1, presenting a symmetric, fully connected neural network designed to reconstruct 24-hour network traffic flows. It consists of five layers: an input layer, an encoder, a bottleneck, a decoder, and an output layer.

The input layer comprises 24 neurons, each representing normalized hourly traffic for a single communication flow. The encoder layer with 16 neurons compresses this 24-dimensional input into a lower-dimensional feature space while retaining essential traffic patterns such as daily peaks or inactivity periods. The bottleneck layer, containing 8 neurons, holds the most compact latent representation of the input and captures the key statistical properties of normal traffic. The decoder layer, also with 16 neurons, mirrors the encoder and progressively reconstructs the compressed features back to the original dimension. The output layer, with 24 neurons and a sigmoid activation function, produces normalized outputs in the range [0, 1], replicating the input traffic profile.

This model is trained in an unsupervised learning setting using only normal traffic flows, enabling it to learn typical patterns without any labelled anomalies. During inference, anomalous flows such as those with service failures or DDoS-like surges are result in higher reconstruction errors, which serve as the primary signal for anomaly detection.

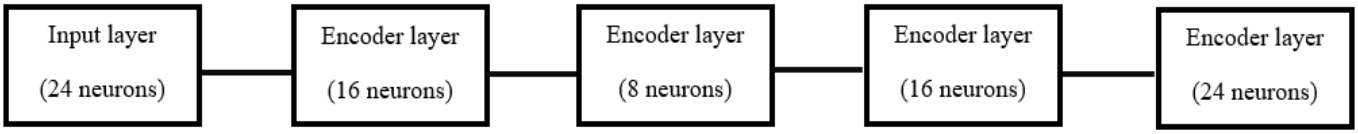

Fig 1. Block-level architecture of the proposed Autoencoder model

3.2 Detailed Connectivity View of Autoencoder Architecture

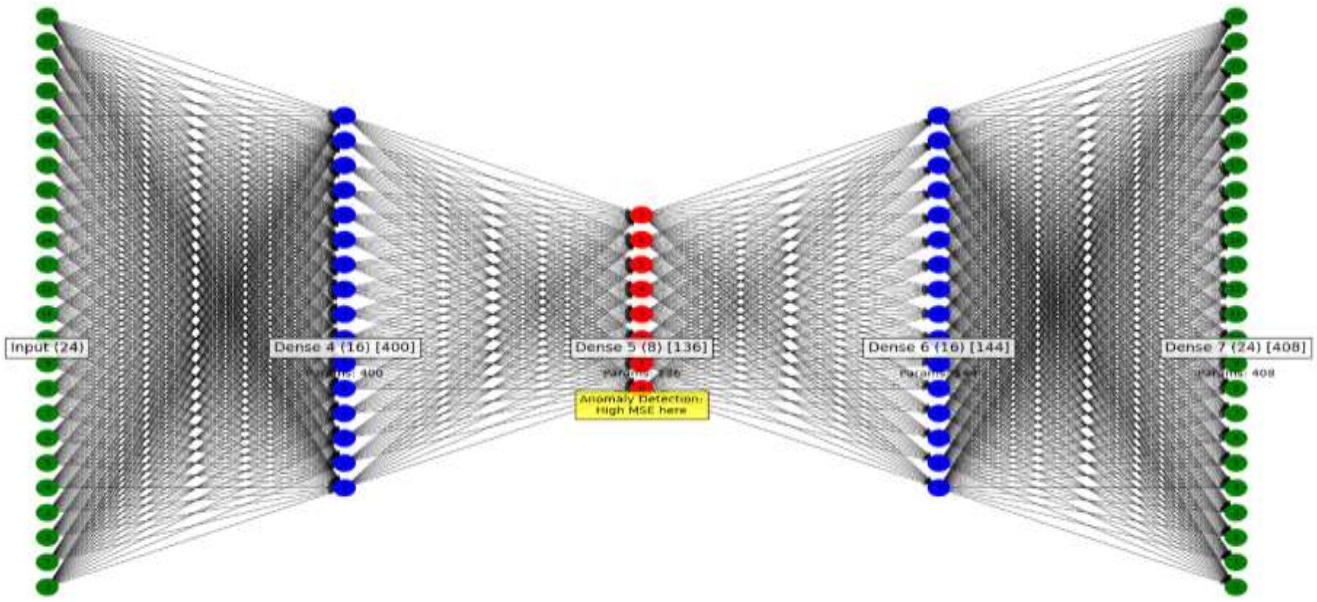

Fig 2. Detailed connectivity diagram of the Autoencoder showing fully connected layers and neuron connections

The internal connectivity of the Autoencoder is shown in Fig. 2, where each layer is fully connected dense layers, which means every neuron in a layer is connected to every neuron in the subsequent layer.

The input layer simply forwards the 24-dimensional normalized traffic vector to the encoder without performing computation. The encoder layer compresses the input to 16 dimensions, having 384 trainable weights and 16 biases equal to total of 400 parameters. The bottleneck layer, with 8 neurons, is the smallest representation of the traffic flow and contains 128 weights and 8 biases equals to total of 136 parameters. The decoder layer, symmetric to the encoder, expands the latent features back to 16 dimensions with 128 weights and 16 biases equal to total of 144 parameters. Finally, the output layer reconstructs the original input with 384 weights and 24 biases equal to total of 408 parameters and applies a sigmoid activation to match the normalized input scale. Training minimizes the Mean Squared Error (MSE) between input and output. After training on clean traffic data, the Autoencoder achieves very low reconstruction error, enabling accurate detection of abnormal flows by comparing reconstruction error distributions.

Table 1: Layer – wise configuration and parameters of the Autoencoder

Layer Name	Input Shape	Output Shape	Activation Function	Trainable Parameters
Input Layer	(None, 24)	(None, 24)	None	0
Encoder Layer (Dense)	(None, 24)	(None, 16)	ReLU	400
Bottleneck Layer (Dense)	(None, 16)	(None, 8)	ReLU	136
Decoder Layer (Dense)	(None, 8)	(None, 16)	ReLU	144
Output Layer (Dense)	(None, 16)	(None, 24)	Sigmoid	408

IV. METHODOLOGY

4.1 Method Overview

The proposed Autoencoder-based anomaly detection framework is designed to analyse and classify 24-hour network traffic flows, covering both service failure and traffic surge DDoS-like anomalies. The overall pipeline, comprising training and testing phases, is illustrated in Fig. 3.

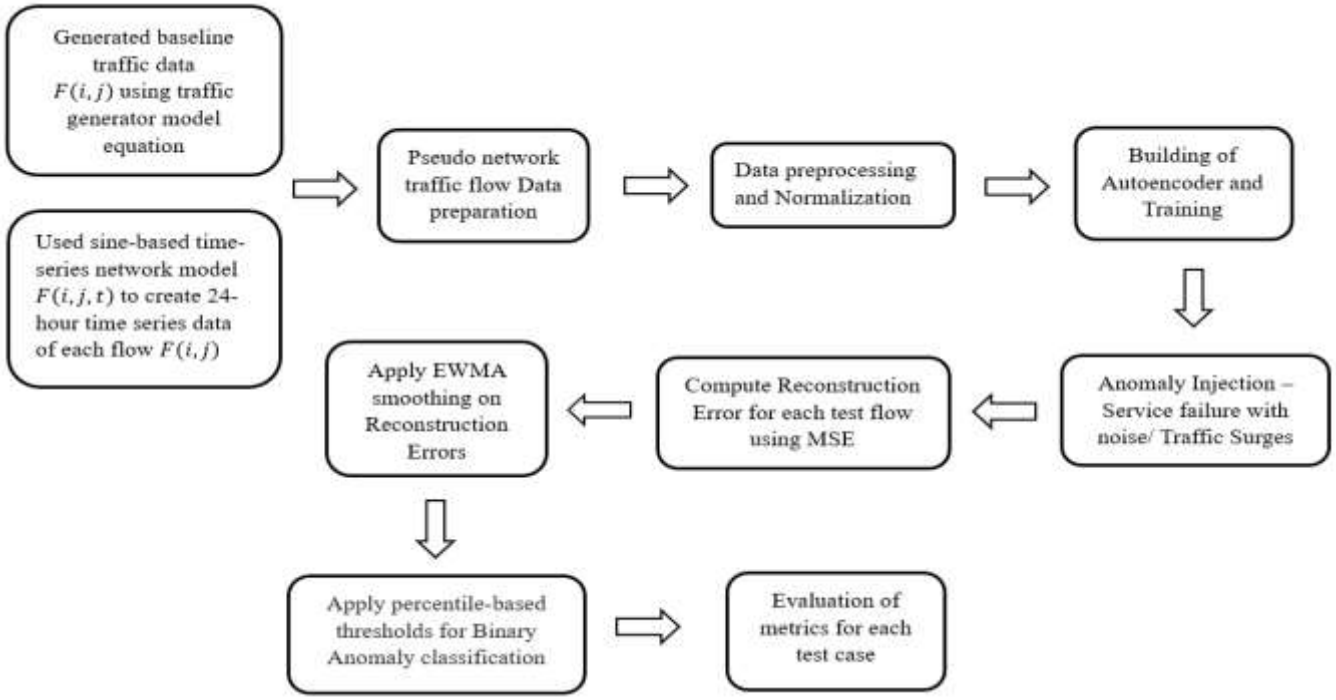

Fig. 3 Overview of the proposed Autoencoder-based anomaly detection framework

Initially, synthetic baseline traffic flows are generated using analytical equations inspired by sine-wave models, simulating periodic communication patterns between multiple user regions and service types. Each flow is modelled as a 24-hour time-series, representing realistic daily fluctuations in network usage. Instead of relying on physical devices or full-scale

simulators, the simulation leverages mathematical modelled traffic generators adapted from the Appendix [7], ensuring high control, reproducibility, and scalability. Although packet-level behaviours are not modelled, the generated flows effectively capture statistical patterns observed in flow-level telemetry systems.

Following traffic generation, Min–Max normalization scales the flows to the range [0, 1], ensuring consistency for neural network training. The Autoencoder is then trained solely on clean, normalized traffic, learning the intrinsic structure of normal flows through dimensionality reduction and reconstruction.

During testing, two synthetic anomalies are injected to emulate real-world events that are one is Service failures, represented by random zero-drops in the network traffic combined with additive Gaussian noise between 1%–70% and the other one is Traffic surges, modelled as multiplicative bursts (2%–8%) in random time windows, mimicking volumetric DDoS attacks. The trained Autoencoder reconstructs the input flows, and reconstruction errors (RE) are computed using Mean Squared Error (MSE). To suppress transient fluctuations, RE values are smoothed with Exponentially Weighted Moving Average (EWMA). Finally, percentile-based dynamic thresholding is applied to classify flows as normal or anomalous.

The model is evaluated using standard metrics such as Accuracy, Precision, Recall, Specificity, and F1-score across multiple noise levels, anomaly severities, and tested with 7 different test dataset sizes ranging from 50 to 100,000 samples. This comprehensive evaluation validates both the robustness and scalability of the proposed system.

4.2 Pseudo Time-Series Network Traffic Preparation

Communication traffic flows represent the patterns and volume of data transmitted across a network between user regions and service endpoints. Modelling such flows is essential for simulating realistic traffic behavior and evaluating anomaly detection methods without relying on real-time traffic captures or large-scale emulators. The baseline static communication flow between region *i* and service *j* is defined as

$$F(i,j) = Users(i) \times service\ rate(i) \times base\ rate(j) \quad \text{...eq (1)}$$

where $Users(i)$ denotes the number of users in region $i, service\_rate(i)$ is the proportion of users using the service, and $base\_rate(j)$ is the per-user data rate for service *j* (in Mbps). To incorporate realistic diurnal variations, each static value $F(i,j)$ is transformed into a 24-hour time-series using a sinusoidal model:

$$F(i,j,t) = \frac{F(i,j)}{2} \cdot sin(\omega t + \phi_0) + \frac{F(i,j)}{2} \quad \text{... eq (2)}$$

Where:

- t ∈ {0,1, 2….,23} denotes each hour of the day (24-hourly time-points for each flow in the network),
- ω= π/12 is the angular frequency to ensure one full cycle over 24 hours,
- ϕ_0 = 2.8 is a constant phase shift, adjusted to align the sine wave with peak traffic times,
- $F(i,j)$ is the static baseline traffic for that flow (in Mbps).

This formulation captures typical diurnal patterns in user activity such as increased usage in the evening and lower activity at night. To introduce natural variance, Gaussian noise is added to each hourly value, simulating fluctuations found in real traffic conditions. The final dataset comprises a 9×24 matrix, corresponding to 9 synthetic flows generated from 3 regions and 3 services using pseudo-traffic generators, each represented as a 24-hour time series. These serve as the foundational inputs to train and evaluate the Autoencoder, enabling it to learn the inherent structure data of clean network behaviour.

4.3 Data Preprocessing and Normalization

Before training the Autoencoder, the generated baseline traffic flows undergo preprocessing to ensure consistent scaling and structural uniformity, both of which are crucial for stable and effective learning. Each region–service communication flow is represented as a 24-hour time-series vector, forming a dataset of initial matrix shape 9 × 24, where each row corresponds to one flow and each column to an hourly traffic value. To remove disparities in absolute traffic magnitudes and standardize inputs for neural network training, Min-Max normalization is applied independently to each flow.

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad \text{... eq (3)}$$

where x denotes the original traffic value at a specific hour, and x_{min} and x_{max} represent the minimum and maximum values within the 24-hour series for that flow. This normalization scales all traffic values to the range [0, 1], preventing bias due to magnitude differences. The resulting normalized dataset preserves the original 9 × 24 data structure and serves as the input to the Autoencoder training phase.

4.4 Autoencoder Training

The Autoencoder was trained exclusively on the normalized traffic matrix of shape 9 × 24, representing nine 24-hour time-series vectors of clean network traffic flows. Training solely on normal data enables the model to learn typical communication patterns without exposure to anomalies, thereby increasing its sensitivity to deviations during inference.

The model was trained for 200 epochs with a mini-batch size of 4, which allowed frequent weight updates and improved generalization given the limited dataset. Optimization was performed using the Adam optimizer with a learning rate of 0.01, chosen for its adaptive learning rate adjustment and efficient convergence. The Mean Squared Error (MSE) loss function was minimized to reduce the discrepancy between the input and reconstructed outputs. The training loss consistently decreased across epochs, confirming that the Autoencoder effectively captured the temporal structure of normal traffic flows.

Once trained, the Autoencoder served as a baseline model for anomaly detection. During testing, flows deviating from learned patterns produced higher reconstruction errors, which were subsequently processed for anomaly classification.

4.5 Anomaly injection and Test Data Construction

Fig.4 illustrates the procedure for injecting synthetic anomalies and constructing the final test datasets. After training the Autoencoder exclusively on clean, normalized traffic flows, anomalous samples are generated to emulate real-world disruptions. Two anomaly types are considered.

Service Failure Anomalies (Noise-Based): Selected random hourly intervals of a traffic flow are zeroed out to simulate partial or complete service outages. Gaussian noise $N(0,\sigma)$ is then added to the zeroed segments, where σ takes values of 1%, 10%, 30%, 50%, and 70% of the original traffic magnitude. The final anomalous segment is obtained as

$$X_{anomalous} = clip(X_{dropped} + N(0,\sigma),0,1) \quad \text{… eq (4)}$$

where $X_{dropped}$ is the original time-series with zeroed traffic data intervals, $N(0,\sigma)$ is Gaussian noise with standard deviation σ and the clip function enforces bounds within the normalized range [0, 1].

Traffic Surge Anomalies (DDoS-like): To simulate volumetric attacks, traffic values within a randomly chosen time window are amplified by a multiplier uniformly sampled between 1.02× and 1.08×:

$$X_{surged} = clip(X_{t1:t2} \times Uniform(1.02,1.08),0,1) \quad \text{…eq (5)}$$

where $Xt1:t2$ denotes the sub-segment of the traffic flow subjected to the artificial surge.

For evaluation, balanced test datasets are created containing equal numbers of normal and anomalous flows. The test sizes are set to 50, 100, 500, 1000, 5000, 10 000, and 100 000 samples, each representing a 24-hour flow. Embedding anomalies at known positions enables objective performance evaluation using standard classification metrics. This structured test construction approach allows for a comprehensive analysis of the model's performance across varying anomaly intensities, anomaly types, and dataset sizes, ensuring a robust and scalable evaluation setup.

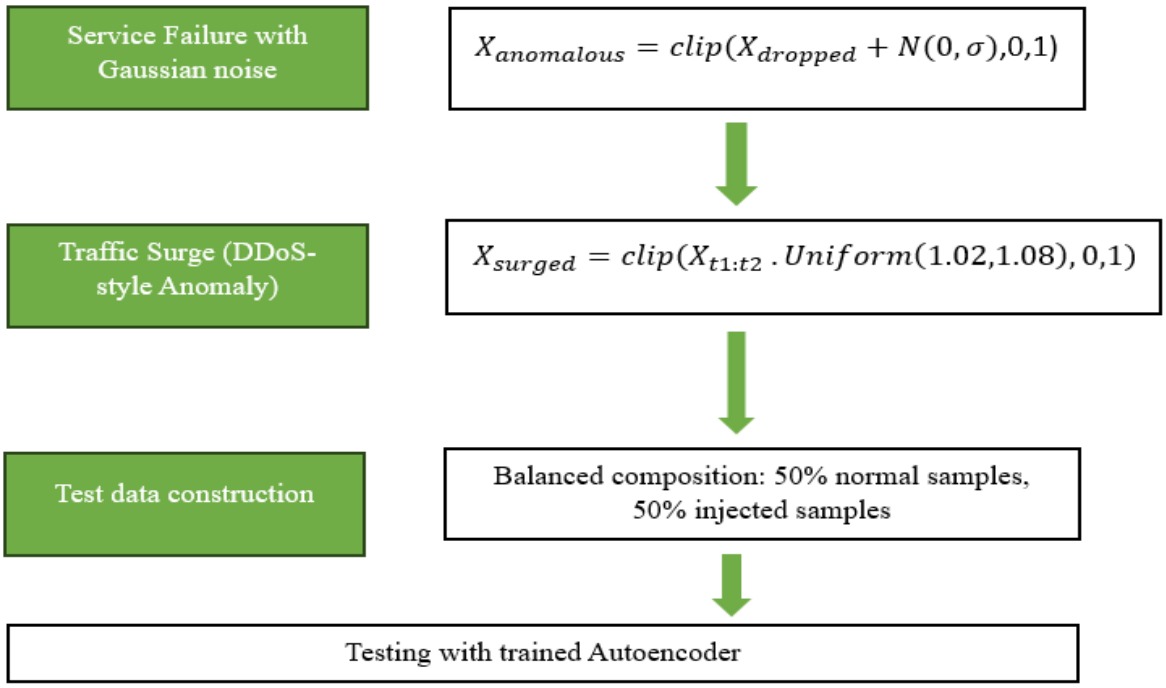

Fig.4 Overview of synthetic anomaly injection and test data construction process

4.6 Anomaly Detection via Reconstruction Error

During the testing phase, each traffic flow is processed by the trained Autoencoder, which reconstructs the input based on the patterns it learned from normal data during training. Normal flows are reconstructed accurately, whereas anomalous flows,

containing unseen disruptions, yield higher reconstruction errors (RE). The reconstruction error for a given flow is computed as the Mean Squared Error (MSE) between the original input X and its reconstruction $\hat{X}$.

$$RE = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \hat{X}_i\right)^2 \quad \text{…eq (6)}$$

To improve detection stability and reduce the impact of transient fluctuations, the raw reconstruction error is further smoothed using the Exponential Weighted Moving Average (EWMA). The smoothed error at time t is defined as

$$EWMA_t = \alpha \cdot RE_t + (1-\alpha) \cdot EWMA_{t-1} \quad \text{…eq (7)}$$

where $\alpha \in (0,1)$ is the smoothing factor controlling the trade-off between responsiveness and stability. A smaller α emphasizes historical behaviour and yields smoother error curves, while a larger α makes the method more sensitive to sudden changes. By combining recent deviations with long-term error trends, EWMA effectively suppresses short-term noise spikes that could otherwise cause false alarms, while retaining meaningful shifts indicative of real anomalies. The resulting smoothed reconstruction error serves as the input for the subsequent threshold-based anomaly classification stage.

4.7 Thresholding mechanism and Binary Anomaly classification

Anomaly classification is performed by comparing the smoothed reconstruction errors of test samples against a dynamically computed threshold. Unlike fixed-value thresholds, the proposed method determines the threshold τ in a data-driven manner based on the percentile rank of the reconstruction error (RE) distribution for every test dataset. This approach ensures that the majority of normal samples fall below τ, while only samples with significantly higher errors are flagged as anomalous this making the method fully unsupervised and adaptable to varying traffic patterns.
To compute the percentile threshold, all reconstruction errors are sorted in ascending order, and the rank r is calculated as

$$r = \frac{p}{100}.(n-1)+1 \quad \text{… eq (8)}$$

where p is the desired percentile and n is the number of test samples. If r is an integer, the threshold is directly assigned as the reconstruction error at that position i.e. x_r in the sorted list. However, if r is a fractional value, interpolation is applied between the k^{th} and $(k+1)^{th}$ sorted RE values to determine a precise threshold. The formula for interpolated threshold is:

$$\tau = x_k + f(x_{k+1} - \mathrm{x_k}) \quad \text{… eq (9)}$$

where k = ⌊r⌋, and f is the fractional part of r. Once the threshold τ is computed, each test sample is classified as anomalous or normal using a binary rule in eq. (10).

$$\hat{X}_i = \begin{cases} 1\ (anomalous), & if\ RE_i > \tau \\ 0\ (normal), & otherwise \end{cases} \quad \text{… eq (10)}$$

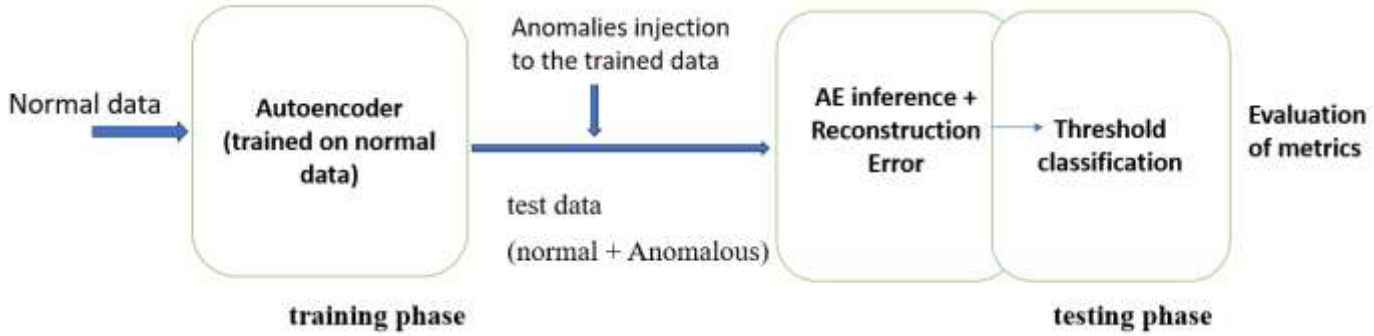

Fig 5**:** proposed Autoencoder-based anomaly detection system, including training on normal data and testing with reconstruction-error-driven dynamic thresholding

V. RESULTS AND ANALYSIS

5.1 Evaluation setup and Objectives

The primary objective of this evaluation is to assess the effectiveness and robustness of the proposed Autoencoder-based anomaly detection framework in identifying deviations in simulated 24-hour IP traffic flows. The model was evaluated across two distinct network anomaly scenarios that are service failure events caused by packet-level disruptions, and traffic surge events

representing volumetric anomalies such as Distributed Denial of Service (DDoS) attacks. Each scenario was tested at multiple severity levels and dataset sizes to investigate both sensitivity to anomaly intensity and the system's ability to scale.

Synthetic network traffic was generated using traffic generator models simulating nine communication flows based on region-service combinations, each represented as a 24-hour time series. These flows were injected with controlled anomalies—Gaussian noise for service failures and scaling multipliers for traffic surges at random intervals spanning 2 to 6 hours. For comprehensive evaluation, the test dataset sizes were varied systematically across seven different scales at 50, 100, 500, 1,000, 5,000, 10,000, and 100,000 samples.

To ensure dynamic and adaptive classification, the model employed Exponential Weighted Moving Average (EWMA) to smooth the reconstruction errors produced by the Autoencoder, followed by percentile-based thresholding for anomaly classification. This unsupervised thresholding method adapts to the underlying distribution of reconstruction errors without relying on labelled data.

Performance was assessed using standard classification metrics such as Accuracy, Precision, Recall, Specificity, and F1 Score to provide a holistic view of the detection capability. These metrics were computed using Tue Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) as follows. Accuracy is the ratio of Proportion of total correct predictions, calculated as eq. (11). Precision is the ratio of Proportion of correctly identified anomalies among all samples labelled as anomalies, calculated as eq. (12). Recall also known as Detection Rate here is the ratio of Proportion of actual anomalies that were correctly identified, calculated as eq. (13). Specificity is the Proportion of actual normal samples correctly identified as normal, calculated as eq. (14). F1 Score is harmonic mean of Precision and Recall, calculated as eq. (15).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{…eq (11)}$$

$$Precision = \frac{TP}{TP+FP} \quad \text{…eq (12)}$$

$$Recall = \frac{TP}{TP+FN} \quad \text{…eq (13)}$$

$$Specificity = \frac{TN}{TN+FP} \quad \text{…eq (14)}$$

$$F1\ Score = \frac{2.Precision.Recall}{Precision+Recall} \quad \text{…eq (15)}$$

Table 2: Binary classification confusion matrix

Format	Predicted Normal (0)	Predicted Anomalous (1)
Actual Normal (0)	True Negative (TN)	False Positive (FP)
Actual Anomalous (1)	False Negative (FN)	True Positive (TP)

5.2 Service Failure Anomaly Detection Results

Table 3 summarizes the performance of the proposed Autoencoder-based anomaly detection model in identifying service failure anomalies under five Gaussian noise ratios 1%, 10%, 30%, 50%, and 70% and seven different test dataset sizes in between 50 to 100,000 samples. The evaluation considers standard metrics such as Accuracy, Precision, Recall, Specificity, and F1 Score which are computed along with the corresponding counts of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN).

At low noise levels (1%), the Autoencoder achieves near-perfect detection across all test sizes. For example, at 50 samples, the model achieves 100% for all metrics. Even at the largest dataset size (100,000 samples), the accuracy remains 99.64%, with a minor recall drop to 99.27%. This shows that the model retains high sensitivity and low false-negative rates even as the evaluation scale increases.

At moderate noise levels 10% and 30%, the Autoencoder demonstrates perfect or near-perfect results. For 10% noise, all metrics are consistently 100% across every test size, confirming that moderate Gaussian fluctuations do not affect the model's ability to distinguish normal and anomalous flows. Similarly, for 30% noise, the model mostly achieves 100% accuracy, with only very minor recall deviations 99.8% at 1000 samples and 99.98% at 10,000 samples due to false negative.

At higher noise levels 50% and 70%, where the injected anomalies drastically distort traffic patterns, the Autoencoder still achieves 100% detection across all metrics and dataset sizes. This result highlights the model's robustness and ability to generalize well even under extreme disturbances, where traditional correlation-based methods would typically fail due to noise sensitivity.

A closer look at the TP, FP, TN, and FN values shows that the model maintains zero false positives at every noise level and test size. The number of false negatives (FN) is negligible and occurs only for extremely large datasets at 1% noise as FN are 363 out of 50,000 anomalies. The perfect precision (100%) across all cases indicates that the model never misclassifies normal samples as anomalies, thus completely eliminating false alarms, which is a major limitation of existing rule-based methods. Such strong and

consistent performance across different noise ratios and dataset sizes validates the suitability of the proposed method for real-time network anomaly detection, even in large-scale and dynamic environments.

Table 3**:** Performance of the proposed Autoencoder model for service failure event anomaly detection under varying Gaussian noise ratios and test dataset sizes using reconstruction error with percentile-based thresholding.

Service failure event due to noise	No: of samples	Accuracy	Precision	Recall	Specificity	F1 score	TP	FP	TN	FN
1% noise	50	100%	100%	100%	100%	100%	25	0	25	0
	100	99%	100%	98%	100%	98.99%	49	0	50	1
	500	99.8%	100%	99.6%	100%	99.80 %	249	0	250	1
	1000	99.6%	100 %	99.2 %	100%	99.6%	496	0	500	4
	5000	99.52%	100%	99.04%	100%	99.52%	2476	0	2500	24
	10000	99.57%	100%	98.14%	100%	99.57 %	4957	0	5000	43
	100000	99.64 %	100%	99.27%	100%	99.64%	49637	0	50000	363
	50	100%	100%	100%	100%	100%	25	0	25	0
	100	100%	100%	100%	100%	100%	50	0	50	0
	500	100%	100%	100%	100%	100%	250	0	250	0

10% noise	1000	100%	100%	100%	100%	100%	500	0	500	0
	5000	100%	100%	100%	100%	100%	2500	0	2500	0
	10000	100 %	100%	100%	100%	100%	5000	0	5000	0
	100000	100%	100%	100%	100%	100%	50000	0	50000	0
30% noise	50	100%	100%	100%	100%	100%	25	0	25	0
	100	100%	100%	100%	100%	100%	50	0	50	0
	500	100%	100%	100%	100%	100%	250	0	250	0
	1000	99.9%	100%	99.8%	100%	99.90%	499	0	500	1
	5000	100%	100%	100%	100%	100%	2500	0	2500	0
	10000	99.99%	100%	99.98%	100%	99.99%	4999	0	5000	1
	100000	100%	100%	100%	100%	100%	49999	0	50000	1
50% noise	50	100%	100%	100%	100%	100%	25	0	25	0
	100	100%	100%	100%	100%	100%	50	0	50	0
	500	100%	100%	100%	100%	100%	250	0	250	0
	1000	100%	100%	100%	100%	100%	500	0	500	0
	5000	100%	100%	100%	100%	100%	2500	0	2500	0
	10000	100%	100%	100%	100%	100%	5000	0	5000	0
	100000	100%	100%	100%	100%	100%	50000	0	50000	0
70% noise	50	100%	100%	100%	100%	100%	25	0	25	0
	100	100%	100%	100%	100%	100%	50	0	50	0
	500	100%	100%	100%	100%	100%	250	0	250	0
	1000	100%	100%	100%	100%	100%	500	0	500	0
	5000	100%	100%	100%	100%	100%	2500	0	2500	0
	10000	100%	100%	100%	100%	100%	5000	0	5000	0
	100000	100%	100%	100%	100%	100%	50000	0	50000	0

5.3 Traffic Surge Anomaly Detection Results

Table 4 presents the performance of the proposed Autoencoder-based anomaly detection model for detecting traffic surge anomalies that emulate DDoS-style volumetric attacks. The evaluation was conducted for five traffic increase ratios 10%, 25%, 50%, 75%, and 100% and seven test dataset sizes ranging in between 50 to 100,000 samples. The table reports the values of Accuracy, Precision, Recall, Specificity, F1 Score, and the corresponding TP, FP, TN, and FN counts.

At lower traffic surges 10%, the Autoencoder maintains high detection performance across all dataset sizes. Even at the largest dataset 100,000 samples, the model achieves 99.46% accuracy, 99.56% precision, and 99.35% recall, with only a slight increase in false negatives and false positives due to the subtle nature of the anomalies. This demonstrates that the proposed framework is sensitive enough to detect even minor deviations from normal traffic patterns.

For moderate traffic surges 25%, the detection metrics improve further, achieving nearly perfect performance at all dataset sizes. The reconstruction errors for anomalous flows become more pronounced as the surge ratio increases, allowing the percentile-based thresholding to cleanly separate normal and anomalous samples. The absence of false positives across all dataset sizes highlights the model's ability to avoid false alarms.

At higher surge ratios 50%, 75%, and 100%, the Autoencoder consistently achieves near-perfect or perfect detection performance. For example, at 50% and 75% traffic increases, the model yields 100% accuracy, precision, and recall for almost all dataset sizes, with minimal false negatives even at the largest scale. The dynamic thresholding strategy adapts well to the error distribution, maintaining stable classification as the anomaly intensity grows.

Table 4: Performance of the proposed Autoencoder for traffic surge anomaly detection under varying traffic increase ratios and test dataset sizes using reconstruction error with percentile-based thresholding

Traffic increases event due to DDoS attacks	No: of samples testing	Accuracy	precision	Recall	Specificity	F1 score	TP	FP	TN	FN
10% traffic increase event	50	100%	100%	100%	100%	100%	25	0	25	0
	100	100%	100%	100%	100%	100%	50	0	50	0
	500	99%	100%	98%	100%	98.99%	245	0	250	5
	1000	99%	100%	98%	100%	98.99%	490	0	500	10
	5000	99.32%	99.4%	99.24%	99.4%	99.32%	2481	15	2485	19
	10000	99.37%	99.38%	99.36%	99.38%	99.37%	4968	31	4969	32
	100000	99.46%	99.56%	99.35%	99.56%	99.46%	49674	218	49782	326
	50	100%	100%	100%	100%	100%	25	0	25	0

25% traffic increase event	100	100%	100%	100%	100%	100%	50	0	50	0
	500	99.8 %	100%	99.6%	100%	99.80%	249	0	250	1
	1000	99.9%	100%	99.8%	100%	99.90%	499	0	500	1
	5000	99.92%	100%	99.84%	100%	99.92%	2496	0	2500	4
	10000	99.96%	100%	99.92%	100%	99.96%	4996	0	5000	4
	100000	99.98%	100%	99.96%	100%	99.98%	49981	0	50000	19
50% traffic increase event	50	100%	100%	100%	100%	100%	25	0	25	0
	100	100%	100%	100%	100%	100%	50	0	50	0
	500	99.8%	100%	99.6%	100%	99.8%	249	0	250	1
	1000	100%	100%	100%	100%	100%	500	0	500	0
	5000	100%	100%	100%	100%	100%	2500	0	2500	0
	10000	99.99%	100%	99.98%	100%	99.99%	4999	0	5000	1
	100000	99.93%	99.93%	99.93%	99.93%	99.93%	49967	33	49967	33
75% traffic increase event	50	100%	100%	100%	100%	100 %	25	0	25	0
	100	100%	100%	100%	100%	100%	50	0	50	0
	500	99.8%	100%	99.6%	100%	99.8%	249	0	250	1
	1000	100%	100%	100%	100%	100%	500	0	500	0
	5000	99.98%	100%	99.96%	100%	99.98%	2499	0	2500	1
	10000	99.99%	100%	99.98%	100 %	99.99%	4999	0	5000	1
	100000	99.93 %	99.93%	99.93%	99.93%	99.93%	49967	33	49967	33
100% traffic increase event	50	100%	100%	100%	100%	100%	25	0	25	0
	100	100%	100%	100%	100%	100%	50	0	50	0
	500	99.8%	100%	99.6%	100%	99.80%	249	0	250	1
	1000	100%	100%	100%	100%	100%	500	0	500	0
	5000	99.98%	100%	99.96%	100%	99.98%	2499	0	2500	1
	10000	99.99%	100%	99.98%	100%	99.99%	4999	0	5000	1
	100000	99.93%	99.93 %	99.93%	99.93 %	99.93%	49967	33	49967	33

The results confirm that the Autoencoder can accurately detect traffic surges of varying intensities while scaling effectively to large datasets. The combination of reconstruction-error learning, EWMA smoothing, and percentile-based thresholding enables the framework to handle both subtle and extreme anomalies without compromising precision or recall. This makes the proposed model highly reliable for detecting DDoS-style volumetric anomalies in real-world network environments.

5.4 Results Overview

The proposed Autoencoder-based anomaly detection framework was evaluated on two distinct anomaly scenarios. One is Service Failure Events, which was simulated by random traffic zero-drops with additive Gaussian noise and the other is Traffic Surge Events, which was simulated by multiplicative traffic bursts representing DDoS-style volumetric attacks.

Table 3 presents the detection performance under five Gaussian noise ratios 1%, 10%, 30%, 50%, 70% and seven dataset sizes between 50 to 100,000 samples. The Autoencoder consistently achieved very high accuracy, precision, recall, specificity, and F1-scores, even as the dataset size increased. At low noise levels 1%, detection remained near perfect, with only a small recall drop 99.27% at 100,000 samples. For moderate noise 10% and 30%, the model achieved 100% accuracy and recall across all dataset sizes, confirming that it generalizes well to traffic variations. Even at higher noise levels 50% and 70%, the Autoencoder maintained perfect metrics, showing robustness against heavy distortions in traffic patterns.

Table 4 reports the detection performance for five traffic increase ratios 10%, 25%, 50%, 75%, 100% across seven dataset sizes. For minor surges 10%, the Autoencoder achieved 99.46% accuracy and 99.35% recall at 100,000 samples, indicating sensitivity to subtle anomalies. As the surge ratio increased 25% to 100%, the model achieved near-perfect or perfect detection, with zero false positives in most cases. The reconstruction error separation became more distinct at higher surge ratios, allowing percentile-based thresholding to cleanly classify anomalies.

Overall, our model showed consistent scalability as dataset size grew from 50 to 100,000 samples with the help of Dynamic percentile-based thresholding combined with EWMA smoothing enabled robust anomaly detection, even for subtle deviations. Both service failure and traffic surge anomalies were detected with greater than 99% accuracy, precision, and recall in almost all scenarios.

5.5 Comparative Analysis with Existing Method

This analysis provides a detailed performance comparison between the baseline Fast xFlow Proxy method, which uses correlation-based anomaly detection with dynamic rule thresholds, and the proposed Autoencoder-based anomaly detection framework, which employs unsupervised reconstruction error analysis, EWMA smoothing, and percentile-based adaptive thresholding. The objective of this comparison is to quantify the improvement achieved by the Autoencoder model in terms of detection accuracy, precision, recall, specificity, and F1-score under two types of anomalies: service failure events and traffic surge (DDoS-style) events.

Service Failure Event Comparison between Fast xFlow Proxy and Autoencoder:

Table 5 summarizes the comparison results for service failure anomalies at five Gaussian noise 1%, 10%, 30%, 50%, and 70% ratios. The Fast xFlow Proxy model achieved moderate performance, with accuracy decreasing from 94.9% at 1% noise to 84.5% at 70% noise. Precision and recall followed a similar trend, showing reduced detection capability as noise increased. Its F1-scores remained below 90% for all noise ratios, confirming its limited robustness to varying noise conditions.

Table 5: Comparison of Service Failure Anomaly Detection Results between Fast xFlow Proxy and Autoencoder Models at Different Noise Ratios. (Autoencoder metrics are averaged across seven test dataset sizes.)

Correlation based anomaly detection with Fast XFlow proxy model						**Reconstruction-Error based anomaly detection with Autoencoder model**				
Accuracy	**Precision**	**Recall**	**Specificity**	**F1 Score**	**Noise ratio**	**Accuracy**	**Precision**	**Recall**	**Specificity**	**F1 Score**
94.9%	89.9%	86.9%	90.9%	88.4%	1%	99.59%	100%	99.03%	100%	99.58%
93.5%	88.5%	85.5%	89.5%	87%	10%	100%	100%	100%	100%	100%
90.5%	85.5%	82.5%	86.5%	84%	30%	99.98%	100%	99.96%	100%	99.98%
87.5%	82.5%	79.5%	83.5%	81%	50%	100%	100%	100%	100%	100%
84.5%	79.5%	76.5%	80.5%	78%	70%	100%	100%	100%	100%	100%

In contrast, the Autoencoder-based model achieved near-perfect detection at all noise ratios. It consistently reached Accuracy, Precision, Recall, Specificity and F1-scores above 99%, even at large dataset sizes up to 100,000 samples. Importantly, the Autoencoder's reported metrics are averaged values across all dataset sizes, confirming its ability to scale while maintaining performance. The bar chart in Fig.6 further illustrates this improvement. It clearly shows that for every metric, the Autoencoder substantially outperforms the Fast xFlow Proxy, with the performance gap widening at higher noise ratios.

Traffic Surge (DDoS) Event Comparison between Fast xFlow Proxy and Autoencoder:

Table 6 presents the comparison results for traffic surge anomalies at five traffic increase levels 10%, 25%, 50%, 75%, and 100%.

The Fast xFlow Proxy model again showed limited detection capability, with accuracy ranging from 94.68% at 10% traffic increase down to 80% at 100% traffic increase. Precision, recall, and F1-scores followed similar downward trends, confirming the method's sensitivity to traffic fluctuations.

By comparison, the Autoencoder model achieved nearly perfect detection performance across all surge levels, with accuracy, precision, recall, and F1-scores consistently above 99%, even for the largest test dataset size of 100,000 samples. The model's percentile-based thresholding adapted effectively to changes in reconstruction error distributions, while EWMA smoothing helped suppress transient error spikes, leading to stable anomaly detection across varying anomaly intensities. Fig.7 shows the corresponding bar chart comparison for the traffic surge scenario. The Autoencoder's bars remain consistently near 100% for all metrics, clearly outperforming the Fast xFlow Proxy approach at all traffic surge levels.

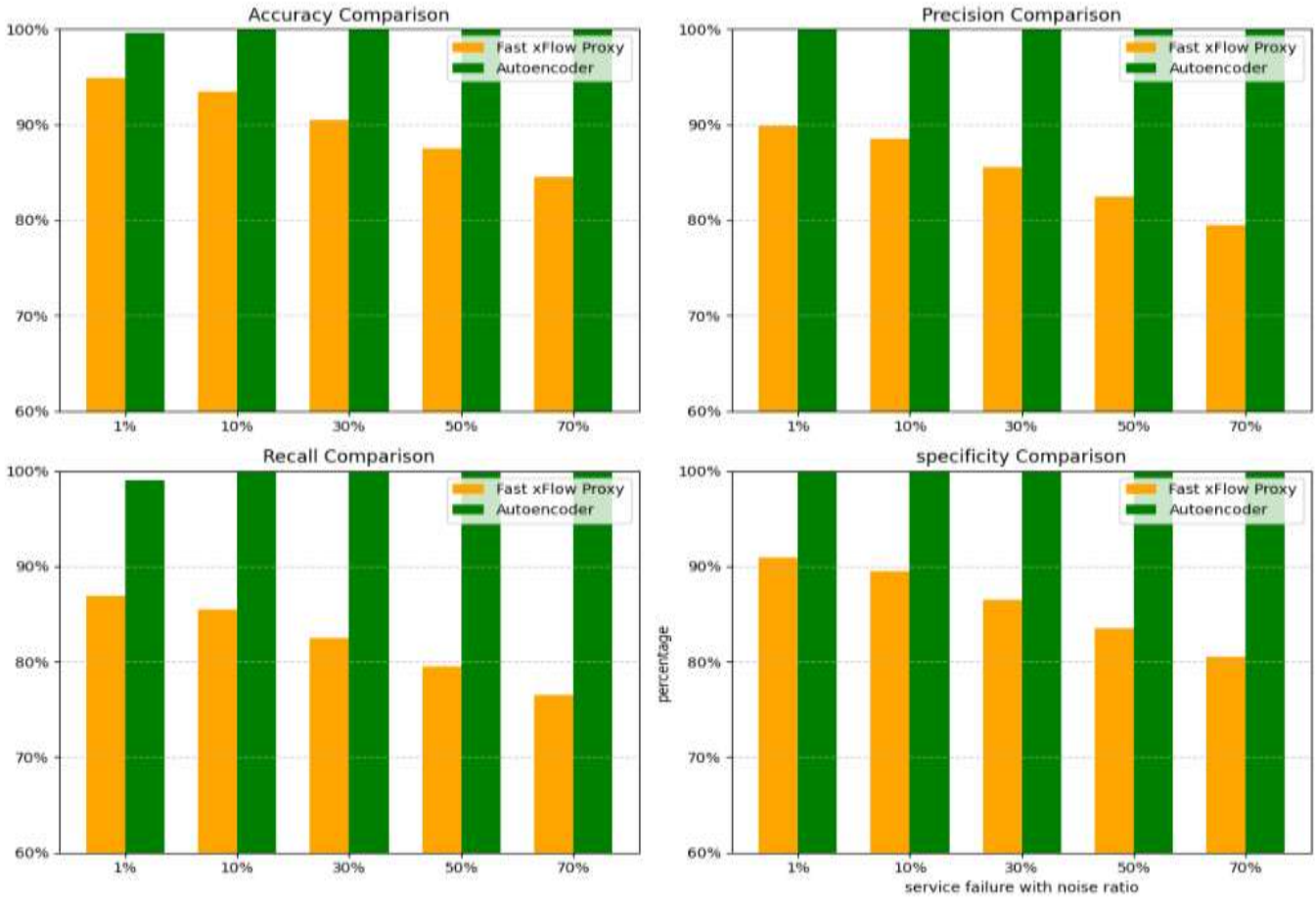

Fig.6: comparison of Accuracy, Precision, Recall, Specificity of both Fast xFlow Proxy and Autoencoder models for service failure anomalies across five Gaussian noise ratios

Table 6**:** Comparison of Traffic Surge due to DDoS attack Anomaly Detection Results between Fast xFlow Proxy and Autoencoder Models at Different Traffic Increase Levels (Autoencoder metrics are averaged across seven test dataset sizes.)

Correlation based anomaly detection with Fast XFlow proxy model						**Reconstruction-Error based anomaly detection with Autoencoder model**				
Accuracy	**Precision**	**Recall**	**Specificity**	**F1 score**	**Traffic increase ratio**	**Accuracy**	**Precision**	**Recall**	**Specifi -city**	**F1 score**
94.68%	90.6%	88.6 %	89.6%	89.67%	10%	99.45%	99.76%	99.13%	99.76%	99.44%
93.14 %	89.14 %	87.1 %	88.14 %	88.13%	25%	99.93%	100%	99.87%	100%	99.91%
88.48%	84.4%	82.48%	83.4 %	83.47%	50%	99.96%	99.99%	99.93%	99.99%	99.96%
82.49%	78.49%	76.49%	79%	77.48%	75%	99.95%	99.99%	99.92%	99.99%	99.95%
80 %	78 %	76%	79%	76.98%	100%	99.95%	99.99%	99.92%	99.99%	99.95%

From model comparison analysis in both anomaly cases, it is showing that the Fast xFlow Proxy model struggles with scalability and noise robustness due to its dependency on correlation values and static thresholds. The proposed Autoencoder framework, on the other hand, achieved average accuracies of 99.91% for service failure events and 99.84% for traffic surge events, as it was evaluated across seven different test dataset sizes for each anomaly type and their intensity ratio. Tables 5 and 6 summarize the results for service failure events and traffic increase events of both models, respectively, while Figures 6 and 7 present bar chart visualizations of the model's performance comparison in both anomaly types. These results clearly show that the proposed Autoencoder-based approach outperforms the Fast xFlow Proxy method in all metrics and under all conditions.

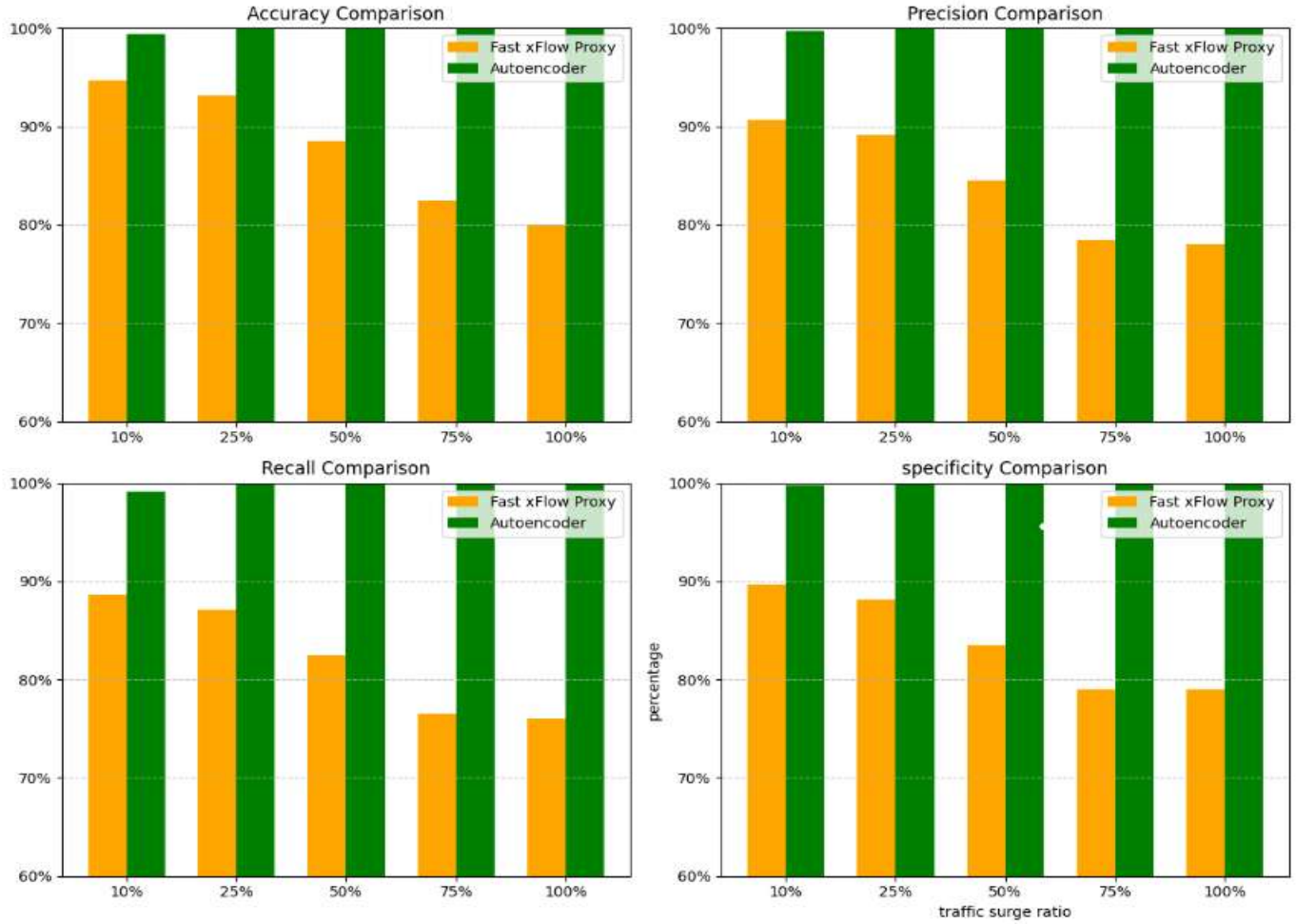

Fig.7: comparison of Accuracy, Precision, Recall, Specificity of both Fast xFlow Proxy and Autoencoder models for traffic surge anomalies across five Gaussian noise ratios

5.6 Results Discussion

The experimental results clearly demonstrate the effectiveness, robustness, and scalability of the proposed Autoencoder-based anomaly detection framework. Across both service failure events and traffic surge anomalies, the Autoencoder consistently achieved accuracy, precision, recall, and F1-scores exceeding 99%, even as the dataset size scaled up to 100,000 samples. These findings highlight the model's capability to generalize well to unseen anomalies while maintaining stability under large-scale traffic conditions.

A key factor contributing to this superior performance is the reconstruction error-based detection mechanism. By training exclusively on clean traffic, the Autoencoder learns the intrinsic structure of normal pseudo network traffic communication flows. Any deviation caused by noise, service failure, or traffic surges does results in a significant increase in reconstruction error. Applying EWMA smoothing to these errors reduces the impact of transient fluctuations, ensuring that persistent anomalies are more reliably detected. The percentile-based thresholding further adapts to the distribution of errors, eliminating the need for fixed thresholds and enhancing detection robustness in dynamic traffic environments.

The comparative analysis against the Fast xFlow Proxy method reinforces these findings. While the baseline method achieved moderate performance, its reliance on correlation values and rule-based thresholds made it less resilient to noise and traffic variations. In contrast, the Autoencoder maintained superior performance across all metrics and test configurations. Notably, the average accuracy for service failure events was 99.91%, and for traffic surge events, it was 99.84%, compared to much lower values for Fast xFlow Proxy.

Moreover, the scalability analysis shows that the Autoencoder effectively handles datasets of up to 100,000 samples without performance degradation, confirming its suitability for real-time, high-volume network environments. The combination of unsupervised training, reconstruction error analysis, EWMA smoothing, and percentile-based thresholding provides a generalizable, noise-resilient, and threshold-free anomaly detection solution.

These results underline that the proposed framework is not only capable of outperforming traditional correlation-based approaches but is also highly adaptable for modern cybersecurity systems where traffic patterns are dynamic and labelled anomalies are unavailable.

V. Conclusion

This study presented an unsupervised Autoencoder-based anomaly detection framework for analysing 24-hour IP traffic flows under two critical anomaly scenarios: service failure events and traffic surge (DDoS-style) events. By training exclusively on clean traffic data, the Autoencoder effectively learned normal communication patterns and detected deviations using reconstruction error analysis, enhanced by percentile-based dynamic thresholding and EWMA smoothing.

Extensive experiments across seven dataset sizes in between 50 – 100,000 samples and multiple anomaly intensities demonstrated that the model consistently achieved near-perfect detection performance, with average accuracies of 99.91 % for service failure anomalies and 99.84 % for traffic surge anomalies. The method exhibited excellent scalability, maintaining high precision, recall, and F1-scores even for large test datasets.

A comparative evaluation against the baseline Fast xFlow Proxy method confirmed that the proposed Autoencoder significantly outperforms the baseline in terms of accuracy, precision, recall, and scalability across all test scenarios. These results validate the robustness and practicality of the proposed model for real-time, unlabelled network anomaly detection in modern cybersecurity systems.

In future work, the framework will be extended to incorporate real-world traffic datasets, validate its effectiveness in encrypted and heterogeneous network environments, and explore online/streaming Autoencoder variants for real-time high-speed anomaly detection. Additionally, we aim to integrate hybrid deep learning techniques, such as attention mechanisms or graph-based models, to capture more complex temporal-spatial traffic patterns and enable multi-class anomaly categorization beyond binary detection.

REFERENCES

[1] Y. Wei et al., "Reconstruction-based LSTM-Autoencoder for Anomaly-based DDoS Attack Detection over Multivariate Time-Series Data," *Journal of VTEX Class Files*, vol. 14, no. 8, pp. 1–13, 2021.

[2] K. Giri, M. Gupta, and P. Dadheech, "An Efficient Hybrid Approach for Intrusion Detection in Cyber Traffic Using Autoencoders," *SN Computer Science*, vol. 4, p. 498, 2023, doi: 10.1007/s42979-023-01865-3.

[3] B. A. Manjunatha et al., "A Network Intrusion Detection Framework on Sparse Deep Denoising Auto-Encoder for Dimensionality Reduction," *Soft Computing*, vol. 24, pp. 4503–4517, 2020, doi: 10.1007/s00500-020-04948-x.

[4] S. Narmadha and N. V. Balaji, "Improved Network Anomaly Detection System Using Optimized Autoencoder-LSTM," *Cluster Computing*, pp. 1–11, 2022, doi: 10.1007/s10586-022-03752-3.

[5] S. A. Najafi, M. H. Asemani, and P. Setoodeh, "Attention and Autoencoder Hybrid Model for Unsupervised Online Anomaly Detection," *arXiv preprint*, arXiv:2401.03322, Jan. 2024.

[6] Y. Zhou, H. Zeng, Z. Zheng, and W. Zhang, "Network Traffic Anomaly Detection Model Based on Feature Grouping and Multi-Autoencoders Integration," *Electronics Letters*, vol. 60, no. 23, 2024, doi: 10.1049/el.97003.

[7] S. Kamamura, Y. Takei, M. Nishiguchi, Y. Hayashi, and T. Fujiwara, "Network Anomaly Detection Through IP Traffic Analysis with Variable Granularity," *IEEE Access*, vol. 11, pp. 129818–129826, 2023, doi: 10.1109/ACCESS.2023.3327890.

[8] Wong, L., Liu, D., Berti-Equille, L., Alnegheimish, S., & Veeramachaneni, K. (2022). AER: Auto-Encoder with Regression for Time Series Anomaly Detection. IEEE. https://github.com/sintel-dev/Orion.

[9] H. Torabi, S. L. Mirtaheri, and S. Greco, "Practical Autoencoder-Based Anomaly Detection by Using Vector Reconstruction Error," *Cybersecurity*, vol. 6, no. 1, pp. 1–17, 2023, doi: 10.1186/s42400-022-00134-9.

[10] K. Berahmand et al., "Autoencoders and Their Applications in Machine Learning: A Survey," *Artificial Intelligence Review, vol. 57, p.28, 2024, doi: 10.1007/s10462-023-10662-6.*

[11] K. A. Alaghbari, H.-S. Lim, M. H. M. Saad, and Y. S. Yong, "Deep Autoencoder-Based Integrated Model for Anomaly Detection and Efficient Feature Extraction in IoT Networks," *IoT*, vol. 4, no. 3, pp. 345–365, 2023, doi: 10.3390/iot4030016.

[12] S. A. Salinas Monroy, A. K. Gupta, and G. Wahlstedt, "Detection of Malicious DNS-over-HTTPS Traffic: An Anomaly Detection Approach Using Autoencoders," *IEEE Access*, 2023.