Enhancing Network Security with a Multi-Modal Auto-Encoder for Netflow Traffic Analysis

Ravi Veerabhadrappa, Poornima Athikatte Sampigerayappa
Research Scholar, Department of CSE, Siddaganga Institute of Technology, Affiliated to VTU Belagavi, India
E-mail: ravi@sit.ac.in, aspoornima@sit.ac.in

Keywords: Network traffic anomaly, multi-modal auto encoder, long short term memory networks

Received: June 14, 2025

In today's landscape of encrypted network communications, traditional intrusion detection systems (IDS) face significant challenges in analyzing traffic effectively. Their limited visibility into packet contents complicates the detection of diverse and evolving attack vectors. The integration of various data sources and flow monitoring tools further exacerbates these issues, making it difficult to form a coherent picture of network security. To address this, a novel framework is proposed that incorporates a multimodal Autoencoder (MMAE) in conjunction with an LSTM model. This approach aims to create and merge latent spaces derived from multiple datasets, enhancing feature aggregation in federated learning scenarios. The MMAE helps reduce dimensionality and align features from data generated by the NetFlow tool. Extensive evaluations were conducted using five benchmark datasets, including NF-UNSW-NB15 and NF-BoT-IoT, to develop a consolidated latent space. The latent spaces were then fused using techniques like concatenation, averaging, and weighted sums. Results from the LSTM classifier revealed a remarkable classification accuracy of 98.5% for the latent space aggregated through the Concat and Weighted sum methods. The proposed framework demonstrates promising potential for distributed anomaly detection in scenarios like Federated IDS. It allows for the efficient merging of similar NetFlow datasets while maintaining privacy and improving aggregation quality.

Povzetek: Članek obravnava izzive zaznavanja anomalij v šifriranem NetFlow prometu, kjer klasični IDS izgubijo vidnost. Predlaga multimodalni avtomatski kodirnik (MMAE) za združevanje latentnih prostorov več NetFlow naborov ter LSTM za časovno klasifikacijo. Združeni latentni prostori (Concat, Weighted sum) omogočajo zanesljivejše porazdeljeno zaznavanje napadov v federiranih IDS-sistemih.

1 Introduction

The rise of technology, primarily driven by big data, cloud computing, and end-to-end information security, has fundamentally transformed how communication occurs over the Internet. As enterprise networks increasingly rely on the cloud and distributed environments, the need for end-to-end encryption schemes becomes critical. While protocols like SSL, TLS, and QUIC secure these communications, they also introduce vulnerabilities. Current threats such as malware, ransomware, replay attacks, and DDoS attacks exploit weaknesses in these encryption protocols [1, 2]. As these threats evolve, so too do the challenges of Cybersecurity monitoring and anomaly detection. Traditional network security approaches often rely on single sources, such as flow metadata, packet metadata, or logs, which are insufficient in addressing sophisticated attacks within encrypted traffic. For instance, Figure 1 illustrates the percentage of attack vectors consisting of DNS and DDoS attacks in recent years.

Moreover, the widespread adoption of encryption protocols such as TLS 1.3 and QUIC complicates network flow analysis. These protocols enhance security by encrypting nearly all communication metadata, reducing the visibility of conventional monitoring tools. This lack of visibility creates a blind spot that attackers can exploit by embedding threats within encrypted sessions, highlighting the urgent need for more effective threat detection mechanisms. Given the high-dimensional nature of encrypted traffic features—such as handshake patterns, session timings, and behavioral fingerprints—traditional intrusion detection systems (IDS) struggle with scalability and noise management. Recent research has underscored the need to integrate multiple forms of analysis and develop more robust and adaptive IDS capable of effectively addressing these evolving threats [3]. As new technologies emerge and the nature of attacks changes, continuous adaptation in security protocols is essential for mitigating potential risks in enterprise environments.

The challenges presented by concept drifts and ongoing technological changes necessitate the adoption of innovative approaches that can effectively assess and monitor encrypted network traffic. With the increasing complexity of network communications, there is a clear motivation to advance our methodologies for detecting and responding to threats. To tackle these challenges, various network monitoring tools, such as CIC-Flow[4], Netflow [5], NFStream [6], generate flow-level metadata that can be used to train learning models for automated intrusion detection [7, 8]. Deep learning models, particularly Autoencoders, can play a significant role in effectively capturing the complex relationships among features within these datasets. By re-

ducing dimensionality and enhancing robustness, these advanced deep learning models can analyze encrypted traffic more effectively [9, 10].

Furthermore, recent developments in Autoencoders provide new avenues for combining and processing information from heterogeneous data sources [11, 12, 13]. The design of these models enables multi-dataset-based training, generating low-dimensional shared representations that can significantly enhance flow-based classifier robustness in handling zero-day attacks [15, 16, 17]. This combination of flow metadata from various sources is pivotal for creating comprehensive defense strategies against the evolving threat landscape. Traditional intrusion detection systems

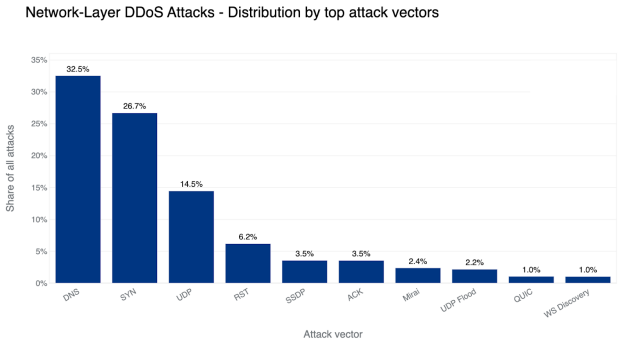


Figure 1: Top DDoS attack vectors in heterogeneous networks (Image courtesy: CloudFlare)

(IDS) have established foundational methods for identifying potential security threats using single-source flow metadata. However, they face significant limitations, particularly in their ability to adapt to diverse and heterogeneous datasets. The reliance on architectures like CNNs, RNNs or standalone LSTMs often results in challenges when analyzing encrypted traffic, such as feature imbalance, noise sensitivity, and poor generalizability. Consequently, these systems struggle to extract meaningful representations essential for effective anomaly detection [18].

On the other hand, Distributed or federated intrusion detection systems are emerging as a promising solution to address these limitations while promoting data privacy and collaboration among organizations. By allowing multiple nodes—such as separate organizations or network segments—to contribute to the detection process without sharing sensitive data, federated IDS leverage the concept of latent space merging. This approach integrates the latent features generated by each node using techniques such as autoencoders, encapsulating valuable insights into network behavior and potential anomalies.

By combining the strengths of traditional IDS with the collaborative framework of federated learning, organizations can enhance their detection capabilities. Conventional methods can serve as a baseline for feature extraction and initial anomaly detection. At the same time, federated learning allows for the continuous refinement of models based on a broader view of network activity. Techniques such as latent space fusion help balance the contributions of each node, thereby improving the accuracy and adaptability of the overall system. Thus, the integration of traditional and federated IDS approaches fosters a more robust cybersecurity framework that addresses both the need for effective anomaly detection and the imperative of maintaining data privacy.

This paper introduces the following key contributions:

1. **Multimodal Autoencoder (MMAE)**: A new architecture for harmonizing and reducing high-dimensional features from multiple datasets.
2. **Latent Space Fusion**: Application of concatenation, averaging, and weighted techniques to combine extracted features for robust representations.
3. **Temporal Classification via LSTM model**: Leveraging the fused latent features to model temporal dependencies and enhance detection accuracy.
4. **Multi-Dataset Generalization**: Achieving high accuracy up to 98.96% across five benchmark NetFlow datasets, significantly outperforming existing models.

The remaining part of the paper is as follows: Section 2 presents surveys related to work in Multimodal learning and traditional IDS approaches. Section 3: Presents the proposed MMAE-LSTM framework, with detailed descriptions of architecture, preprocessing, and algorithms. Section 4: Discusses experimental setup, hyperparameter tuning, and baseline comparisons. Section 5: Presents evaluation results across individual and combined datasets, accompanied by statistical analyses and comparisons to state-of-the-art methods. Section 6: Concludes the study with insights and suggestions for future enhancements.

2 Related work

This section highlights the literary works of various authors that have contributed to the development of traditional IDS systems, enabling them to detect attacks using standalone datasets. Section 2.1 summarizes the conventional IDS, Multi-modal based, and Multi-dataset approaches with challenges and limitations of the works. Section 2.2 summarizes the gaps identified among these techniques.

2.1 Traditional IDS, multi-dataset and multi-modal approaches

Table 1 summarizes essential studies that have employed these cutting-edge methodologies, illustrating the potential value of multi-modal and multi-datasets training to network attack classification. These studies form a basis for further investigation into more adaptive and robust intrusion detection systems.

Habibi et al. [19] introduced a novel Intrusion Detection System (IDS) called IMD-IDS, which was trained on multiple datasets derived from both host and network flows. This model integrates attack data from a single dataset source to create a robust network classifier capable of handling previously unseen attack vectors. Evaluation results indicate that the models exhibit overfitting due to the class imbalance within the training datasets. Shaukat et al. [20] address the challenges related to feature selection and the training of machine learning models. They discuss various machine learning algorithms, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes, and Random Forest. The authors point out that factors such as limited labeled datasets and high data volumes hinder the

practical training of models for unknown attacks. Selecting appropriate classification metrics and datasets can significantly impact the quality of model training. They also emphasize the importance of feature extraction and classification tasks using deep learning models.

Popoola et al. [21] proposed a hybrid model called LAE-BLSTM, which combines Long Short-Term Memory (LSTM) networks with Autoencoder (AE) models. When training deep learning models, the authors emphasize the advantages of using AEs for input dimensionality reduction. The LAE model achieves a remarkable 91.89% reduction in input dimensions, which can significantly reduce both training time and memory usage. Additionally, AEs can capture both spatial and temporal information from flow data while maintaining reduced input dimensionality. The authors emphasize the importance of hybrid models in adapting to new attacks and enhancing robustness by allowing for real-time updates of model parameters.

Table 1: Review of multi-modal and multi-dataset classification works.

Sl. No	Authors	Multi-modal	Multi-datasets
1	Habibi et al., [19]	No	Yes
2	Shaukat et al., [20]	No	No
3	Popoola et al., [21]	No	No
4	Sarhan et al., [22]	No	No
5	Nguyen et al., [23]	No	No
6	Huang et al., [24]	Yes	Yes
7	Torre et al., [25]	No	No
8	Bovenzi et al., [26]	No	No
9	Fox et al., [27]	No	No
10	Zhu et al., [28]	No	Yes
11	Ghani et al., [29]	No	Yes
12	Manocchio et al., [30]	No	No
13	Wang et al., [31]	Yes	No
14	Neloy et al., [32]	No	No
15	Torabi et al., [33]	No	No
16	Irfan et al., [34]	Yes	No
17	L. Yu et al., [35]	Yes	No
18	Kiflay et al., [36]	Yes	No
19	Palakurti et al., [37]	No	No
20	Proposed Work	Yes	Yes

Sarhan et al. [22] examined various feature selection methods for building intrusion detection systems using machine learning. The authors emphasized the importance of feature selection and highlighted the challenges posed by biased results in specific datasets due to chosen feature selection criteria. To develop a network classifier, they conducted model training evaluations on the NF-UNSW-NB15, NF-CSE-CIC-IDS2018, and NF-ToN-IoT datasets. Nguyen et al. [23] examine the issue of domain adaptability in intrusion detection systems (IDS), noting that traditional machine learning models often struggle with unseen data. Deep learning models can be trained to handle unknown or new attacks more effectively. Their proposed approach utilizes the BERT model to extract meaningful representations, enhancing machine learning models' training for prediction and classification tasks. The authors discuss the necessity for deep learning models to manage extensive network flows in real-time, which can improve their robustness and generalizability to new zero-day attacks.

Huang et al. [24] introduce a feature fusion technique called MFFAN: Multiple Features Fusion with Attention Networks, which can extract features from pcap files. Their model extracts features at various levels, including Byte, Packet, and Statistical. It employs a one-dimensional Convolutional Neural Network (1d-CNN) for Byte-level feature extraction, a trained CNN for Packet-level features, and Long Short-Term Memory (LSTM) models to generate encodings from these features. The feature fusion module creates a combined feature space that feeds into a fully connected layer for classification tasks. The authors report achieving a classification accuracy of 99% on two distinct datasets, ISCXIDS2012 and CICIDS2017. However, they also identify challenges related to feature interpretability and adaptability to evolving threats, which pose significant obstacles in model training.

Torre et al. [25] emphasize the necessity of utilizing deep learning models, specifically CNN and LSTM, to analyze cybersecurity attacks. Their work extensively discusses the application of sequence models, including RNNs, Autoencoders, deep neural networks (DNNs), and neural networks (NNs) for attack analysis. The authors provide insights into data representation through Autoencoders, which aid in reducing feature space and efficiently training sequence models to manage long-range dependencies, particularly when analyzing lengthy network flows. Using Autoencoders and transformer-based models can significantly streamline feature space reduction and enhance the extraction of feature sets for practical model training.

Bovenzi et al. [26] focus on the role of Autoencoders in creating compressed feature representations that can be employed for training deep learning models. The authors discuss the advantages of using Autoencoders in conjunction with deep learning models to develop robust classifier models to detect label-flipping and data poisoning attacks. They highlight the importance of generalization in model training and the need for explainability and scalability. Additionally, the work addresses the necessity for multi-dataset generalization to establish a resilient classifier. Fox et al. [27] investigate the design of deep learning models such as 1D-CNN, 2D-CNN, and MLP to detect anomalous network flows early. The authors note the overfitting in models when applied to specific datasets, such as the USTC-TFC2016 dataset, which contains a significant amount of non-representative benign TCP traffic. Their research emphasizes the importance of model-agnostic input representations and the need for specific representations that align with the model's requirements. Evaluation results across multiple datasets indicate an accuracy exceeding 97% when using model-specific representations.

Zhu et al. [28] discuss various deep learning models, including the Cost Matrix Time Space Neural Network (CMTSNN), BiLSTM, and 1D-CNN, in the context of ToN, BoT, and ISCX VPN datasets. They demonstrate that integrating deep learning models with Autoencoders for input representation can alleviate overfitting associated with imbalanced datasets and improve adaptability to new, unseen attack datasets by employing Autoencoders as an unsupervised representation learning technique. Evaluation results highlight the advantages of utilizing the CMTSNN

model, which shows superior classification metrics across the datasets. Ghani et al. [29] discuss the limitations of traditional input representations such as PCA, t-SNE, and UMAP techniques, particularly in scenarios involving highly imbalanced data and data that exhibit linear properties. The authors propose using autoencoders for dimensionality reduction and feature space representation, noting that autoencoders can significantly enhance the performance of deep learning models when training on multiple datasets.

Manocchio et al. [30] proposed a novel flow transformer-based architecture to handle long-term dependencies in flow metadata-based classification. The proposed dataset was tested on publicly available CICIDS-17, UNSW-NB15, and KDD datasets. Embeddings from transformer models can significantly reduce model size by 50% without deliberating on model performance. Additionally, Flow-Transformer improves inference and training times, making it practical for real-world deployment. Wang et al. [31] present a multi-modal perception model designed to aggregate information on various attacks, facilitating the dynamic detection of attack technologies and aiding in the reconstruction of attack maps. Their work emphasizes the use of multi-modal data fusion techniques and their effectiveness in detecting behaviors from various data sources.

Neloy et al. [32] discuss the benefits of utilizing Autoencoders in anomaly detection. The authors highlight how AEs can improve various models' training time and efficiency. They used the MNIST dataset to evaluate their findings and noted the limitations of AEs, such as their sensitivity to noisy inputs and challenges in interpreting the latent space. The choice of an AE should depend on the specific architectural requirements, the tasks at hand, and resource availability. Torabi et al. [33] discuss the analysis of malware families in the context of Internet of Things (IoT) environments. The authors highlight the lack of a comprehensive framework for analyzing network traffic containing malware. They stress the need for a diverse dataset featuring a variety of malware families to support the development of robust classifiers that can address evolving threats. Deep learning techniques such as Convolutional Neural Networks (CNNs), LSTMs, and Recurrent Neural Networks (RNNs) can be trained to classify malware families using automatic feature extraction methods. The authors recommend employing multi-modal feature extraction and analysis techniques for effective malware detection.

Yu et al. [35] introduce a multi-modal deep learning model designed to enhance the classification of anomalies, even in highly imbalanced datasets. Their model combines 1D CNN and GRU architectures for feature extraction, facilitating multi-class classification on the CICIDS2017 and NSL KDD datasets. The incorporation of multi-modal learning significantly improves the performance of the classifier by leveraging low-dimensional latent spaces. Kiflay et al. [36] propose a novel multi-modal network intrusion detection system (NIDS) that detects anomalies using both flow-based and payload-based methods, utilizing only a limited number of features. The model is evaluated on the UNSW-NB15 dataset, employing SHAP values to provide explainability and demonstrating high accuracy in detecting various cyberattacks. Palakurti et al. [37] address the major challenges associated with anomaly detection and the identification of unusual patterns in networks. They advocate for the classification of multi-datasets to enhance the accuracy of machine learning and deep learning models. By integrating multiple datasets, it is possible to create balanced datasets that significantly improve classifier accuracy while incorporating contextual information, thus minimizing false positives.

2.2 Summary of gaps in existing literature

The analysis of previous research on network attack classification reveals a significant gap in utilizing multi-modal data and multi-dataset learning for effective intrusion detection. While some studies have focused on multi-dataset training to improve model generalizability, few have integrated multimodal learning, which is essential for capturing diverse network behaviors. As a result, the absence of multi-modal integration limits the extraction of complementary features, potentially decreasing detection accuracy for complex attacks. Models trained on single datasets often struggle with generalization to new network traffic patterns, which hinders their adaptability in real-world distributed scenarios. The proposed work combines multi-modal learning with multi-dataset training to address these issues, enhance attack classification accuracy, and improve generalizability across various network environments. While surveying, we conclude that our work is the first to use data fusion techniques to combine multi-dataset latent spaces to make sequence models generalize well on various kinds of network attacks.

3 Proposed methodology

The proposed model consists of two main components: a data preprocessing module and a Multi-Modal Autoencoder (MMAE) module. The process begins with the preprocessing of multiple datasets, which is essential for preparing the data for practical analysis and model training. The data preprocessing module, as in Figure 2, systematically extracts features from the NetFlow monitor tool, a critical component for capturing network traffic data. The extracted data is then subjected to several transformations, including normalization and scaling, to ensure consistency and enhance the quality of the input data for training the Long Short-Term Memory (LSTM) model.

In addition, the proposed work focuses on the innovative fusion of features from diverse datasets using early fusion techniques. This approach is particularly beneficial for classification tasks, as it enables a more comprehensive representation of the data by integrating features from multiple sources before feeding them into the classification model. Network capture preprocessing is executed carefully to derive relevant features using the NetFlow protocol. Packet captures contain a wealth of information; therefore, their numerous features must be effectively preprocessed, normalized, and scaled to ensure they are suitable for subsequent classification tasks. Once the individual features are processed, they are combined using a feature concatenation technique. This technique generates a unified and robust

dataset that enhances the information available for analysis.

The resulting preprocessed dataset is subsequently utilized to create multi-modal representations through the MMAE autoencoder module. This module encapsulates the extracted features, allowing for better learning and representation of the complex relationships inherent in the data. Sections 3.1 to 3.2 provide a comprehensive outline and detailed explanation of the data preprocessing steps and other related tasks. Section 3.4 explores the design and architecture of the proposed LSTM model, which is tailored explicitly for classification tasks. Finally, Section 4 discusses the evaluation results of the training performed on the multi-datasets with the proposed LSTM model, highlighting its effectiveness and potential applications.

3.1 Data pre-processing

The dataset distribution for Benign and Malicious network captures are shown in Figure 3, and attack vector distributions in each dataset are shown in Figure 4. To design a multi-dataset classifier, the model needs a pre-processed dataset, which is generated using a pre-processing module that normalizes and scales the various features to make the dataset ready for training. For each dataset, we apply various transformations normalization, and scaling of values and generate multi-modal representations using the auto-encoder model.

The data pre-processing module generates preprocessed inputs to the deep learning model. Each dataset is sampled to contain flow information of attacks and pre-processed for training the model. To train the LSTM model on these datasets, the pre-processing module divides 48 features of each dataset into two categories to represent the multi-modality information. Table 2 shows the features categorized as Entities and Quantities, which represent various dataset and destination features. Each feature needs to be normalized, scaled, and encoded with various techniques to make the dataset class imbalance-free. Entities and Quantities of network flows are preprocessed separately, and later feature fusion is performed to generate preprocessed data. Algorithm 1. show the process of extracting Entities and Quantities features from each dataset.

The dataset comprises mixed types of values: integer, floating point, and categorical. Numerical features are pre-processed for missing values and normalized using the Z-score normalization technique. Later, numerical features are scaled with a robust scaling method to enhance the model's interpretability. Categorical values are grouped to be preprocessed using an encoding technique such as a one-hot or label encoding scheme. A preprocessed feature set is used to obtain the feature importance score using Random Forest, which is trained on the normalized features with the Attack column as the target variable. The feature importance is generated for each dataset before generating latent space using the MMAE autoencoder.

3.2 Multi-modal auto encoder (MMAE) for feature representation

Autoencoders are models that can learn complex patterns from network traffic and can be effectively used for generating synthetic data, feature representations, and other tasks. These models can efficiently fuse different modalities of information into a lower-dimensional space, allowing the models to learn from a common feature space. Figure 5. Shows the proposed MMAE model that can combine multiple datasets to obtain a unified view of the multi-source dataset. The MMAE model, when combined with the LSTM model, can result in a robust multi-dataset classifier for anomaly detection. The main advantage of using MMAE is the dimensionality reduction caused by the hidden layers of the autoencoder, which can help the model converge faster. The main advantage of MMAE is that it reduces the 45 features extracted from NetFlow for each dataset to a much lower-dimensional latent space, which contains 32 features. The resulting latent space can, in turn, capture the temporal dependencies present in the original feature space.

The proposed MMAE model uses the following components: input layer, encoding layer, fusion layer, bottleneck

Algorithm 1 Pseudocode for MMAE with LSTM model for Multi-dataset Netflow Traffic Anomaly Detection

1: **Input:** {NF-UNSW-NB15, NF-ToN-IoT, NF-BoT-IoT, NF-CSE-CIC-IDS2018, NF-UQ-NIDS}
2: **Output:** Latent representations of each dataset.
3: Divide the dataset into Entities and Quantities with features listed in Table 2.
4: **for** each feature in Entities **do**
5: Handle missing values and negative values for each column:
   – Impute missing values using appropriate methods (e.g., mean, median, mode).
   – Replace or transform negative values based on the context (e.g., set to zero).
6: Convert categorical values:
   – Apply One-hot encoding to categorical features (e.g., PROTOCOL, L7_PROTO).
   – Map attack labels using label encoding for the Attack column.
7: Normalize the feature using Z-Score normalization:
   – Calculate the mean and standard deviation for each feature.
   – Normalize each feature using the formula: $Z = \frac{X - \text{mean}}{\text{std}}$.
8: Scale floating values using Robust scaling:
   – Calculate the median and interquartile range (IQR) for each floating feature.
   – Scale values using the formula: Scaled $= \frac{X - \text{median}}{\text{IQR}}$.
9: Combine Entities and Quantities by retaining the order of Flow metadata features.
10: **end for**
11: **for** each pre-processed flow feature set for Netflow data **do**
12: Apply Encoder of MMAE to generate a latent representation of the Netflow data.
13: **end for**
14: For each latent representation of Netflow metadata, generate a combined representation using latent Concat, Average, and Weighted sum technique.
15: Train the LSTM classifier for both Per-dataset and Multi-dataset latent representations with classification reports.

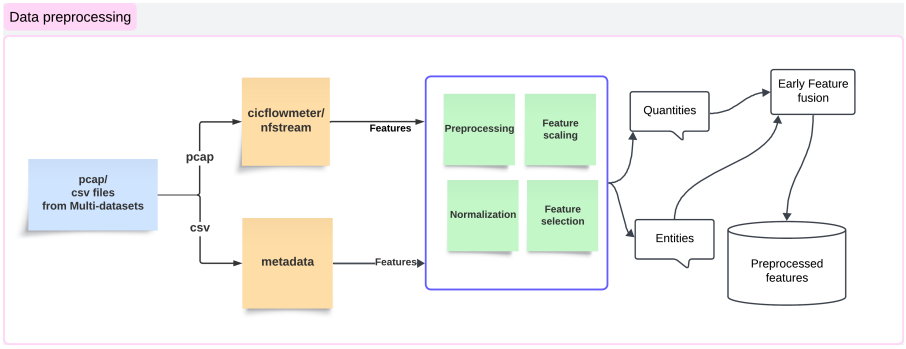


Figure 2: Pre-processing multi-dataset features and early fusion

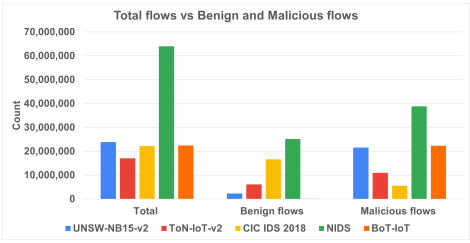


Figure 3: Benign and malicious flows in datasets

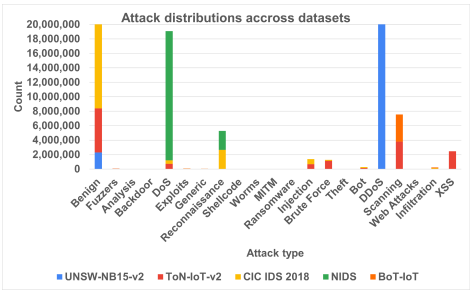


Figure 4: Attack distributions in each netflow dataset

layer, decoder layer, loss function, and lastly, regularization layers. Mathematically described with input and output dimensions as follows: **Input Layer** Let the input be represented as:

$$X \in \mathbb{R}^{45}$$

Layer 1 The first hidden layer is computed as:

$$H_1 = \sigma(W_1 X + b_1) \quad \text{where } W_1 \in \mathbb{R}^{64\times 45},\ b_1 \in \mathbb{R}^{64}$$

Layer 2 The second hidden layer is computed as:

$$H_2 = \sigma(W_2 H_1 + b_2) \quad \text{where } W_2 \in \mathbb{R}^{48\times 64},\ b_2 \in \mathbb{R}^{48}$$

Features are encoded using One-Hot encoding and label encoding is applied to the inputs depending on the type of data i.e. categorical data values. So in the proposed design of MMAE, the encoding layer is not used. Encoding of values results in embeddings that need to be fused to lower latent space.

The fusion layer in MMAE fuses the various embeddings to lower latent space using concatenation operation. The relative order of features from the orginal dataset is retained to extract spatial and temporal relations among feature space. The output of hidden layer 1 and layer 2 generates intermediate latent representations that merge to the final representation of size 32 features.

Latent Representation: The final latent representation is given by:

$$Z \in \mathbb{R}^{32} = W_3 H_2 + b_3 \quad \text{where } W_3 \in \mathbb{R}^{32\times 48},\ b_3 \in \mathbb{R}^{32}$$

Table 2: Entities and quantities values

Entities and Quantities	Values
L4_SRC_PORT	Source Layer 4 Port
IN_BYTES	Incoming Bytes
OUT_BYTES	Outgoing Bytes
TCP_FLAGS	TCP Flag Indicators
CLIENT_TCP_FLAGS	Client TCP Flags
FLOW_DURATION_MILLISECONDS	Flow Duration in ms
DURATION_IN	Incoming Flow Duration
MIN_TTL	Minimum Time-To-Live
MAX_TTL	Maximum Time-To-Live
SRC_TO_DST_SECOND_BYTES	Bytes Sent from Source to Destination per Second
RETRANSMITTED_IN_BYTES	Retransmitted Incoming Bytes
SRC_TO_DST_AVG_THROUGHPUT	Avg. Throughput from Source to Destination
NUM_PKTS_UP_TO_128_BYTES	Packets up to 128 Bytes
TCP_WIN_MAX_IN	Max TCP Window Size (Incoming)
ICMP_TYPE	ICMP Message Type
DNS_QUERY_ID	DNS Query Identifier
FTP_COMMAND_RET_CODE	FTP Command Return Code
Label	Classification Label
L4_DST_PORT	Destination Layer 4 Port
IN_PKTS	Incoming Packets
OUT_PKTS	Outgoing Packets
SERVER_TCP_FLAGS	Server TCP Flags
DURATION_OUT	Outgoing Flow Duration
LONGEST_FLOW_PKT	Longest Packet in Flow
SHORTEST_FLOW_PKT	Shortest Packet in Flow
MAX_IP_PKT_LEN	Maximum IP Packet Length
DST_TO_SRC_SECOND_BYTES	Bytes Sent from Destination to Source per Second
RETRANSMITTED_OUT_BYTES	Retransmitted Outgoing Bytes
DST_TO_SRC_AVG_THROUGHPUT	Avg. Throughput from Destination to Source
NUM_PKTS_128_TO_256_BYTES	Packets 128–256 Bytes
NUM_PKTS_256_TO_512_BYTES	Packets 256–512 Bytes
NUM_PKTS_512_TO_1024_BYTES	Packets 512–1024 Bytes
NUM_PKTS_1024_TO_1514_BYTES	Packets 1024–1514 Bytes
TCP_WIN_MAX_OUT	Max TCP Window Size (Outgoing)
ICMP_IPV4_TYPE	ICMPv4 Message Type
DNS_QUERY_TYPE	DNS Query Type
DNS_TTL_ANSWER	DNS TTL of Answer

The output of MMAE consists of 32 features that represent a compressed version of the original 45 input features; these are learned representations or embeddings derived from the preprocessed 45 features. These inputs enable the model to converge faster for downstream tasks, such as classifying network traffic. Since the objective of the proposed model is to generate a generic representation of the original dataset, the decoder layer is removed after generating the output of the encoding layer.

3.3 LSTM model architecture

In the proposed work, an LSTM model is trained on latent representations obtained from the MMAE autoencoder. LSTM models are sequence models that can process sequence-like information such as network traffic patterns. Due to the memory cells of LSTM models, we train the model with sequences of embeddings generated by MMAE for classification tasks. LSTM model comprising various gates and a hidden layer is explained as follows:

1. **Forget Gate:**

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

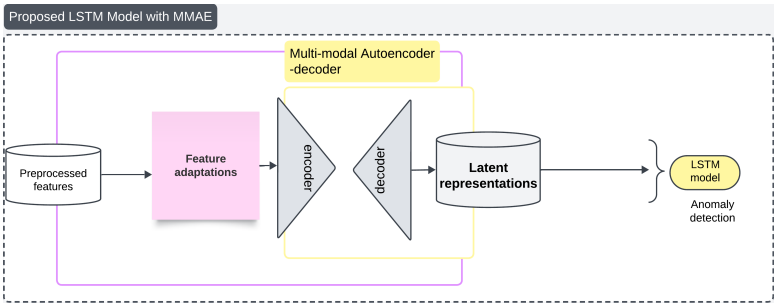


Figure 5: Proposed MMAE with LSTM model

2. **Input Gate:**

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

3. **Cell State Update:**

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{4}$$

4. **Output Gate:**

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{6}$$

Model Architecture 1. For the output of the second LSTM layer:

$$h_T = \text{LSTM2}(\text{LSTM1}(X)) \tag{7}$$

2. For the final output:

$$y = \text{Softmax}(W_d \cdot h_T + b_d) \tag{8}$$

Equations 1. to 6. explains the working of individual cells in the LSTM model, where as Equation 7 and 8. explains the two Hidden layers and output layer of the LSTM to form a concatinated output y.

4 Results and discussion

This section highlights the performance of the proposed model under two setups: per-dataset classification among five Netflow datasets and Multi-dataset classification performance, including classification metrics for each. Lastly, the evaluation of latent fusion techniques is performed using a statistical p-test among Concat, Average, and Weighted Sum techniques to determine which latent fusion is most feasible for enhancing anomaly detection.

4.1 Per-dataset classification performance

Network intrusion detection for heterogeneous networks is challenging due to the dynamics and complexity of cyber attacks. Models trained on a single dataset cannot generalize effectively across various attack scenarios, and thus, their real-world effectiveness remains limited. To overcome this, our proposed MMAE with LSTM classifier model utilizes latent representations obtained from various netflow datasets, which are combined to form a heterogeneous, averaged dataset without relying on raw data. The following discussions show how practical the LSTM model is in training on latent representations obtained by MMAE.

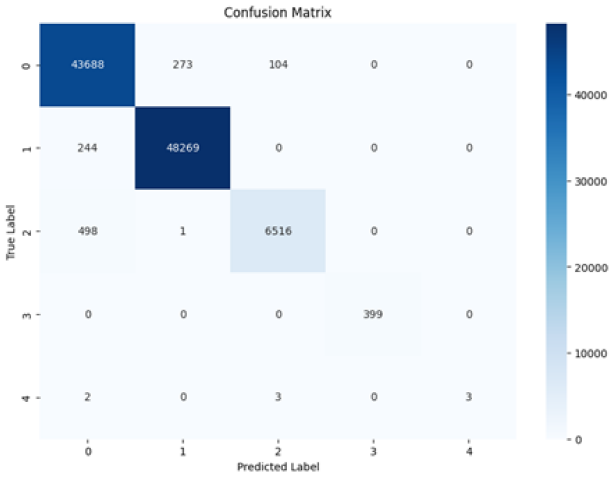
NF-BoT-IoT Dataset:

Figure 6 (a) and (b) shows the advantages of latent-based classification using the LSTM model when trained on the NF-BoT-IoT datasets comprising various IoT attacks. The classification report explains the model's performance on multiple classes, achieving

Classification report for NF-BoT-IoT Dataset

	Precision	Recall	F1-score	Support
Class 0	0.99	0.99	0.99	44065.0
Class 1	0.99	0.99	0.99	48513.0
Class 2	0.98	0.94	0.96	7015.0
Class 3	1.00	1.00	1.00	399.0
Class 4	1.00	0.13	0.22	8.0
accuracy			0.99	100000.0
macro avg	0.99	0.85	0.87	100000.0
weighted avg	0.99	0.99	0.99	100000.0

(a)



(b)

Figure 6: Classification report (a) and latent space training (b) for NF-BoT-IoT dataset

high accuracy for all classes except class 4, which is attributed to the dataset's limited number of samples. The confusion matrix of the classifier illustrates how it accurately classifies classes 0 through 2. In contrast, classes 3 and 4 have less accuracy due to class imbalance in the extracted datasets.

NF-ToN-IoT Dataset

Figure 7 (a) and (b) show the classification report and confusion matrix of the NF-ToN-IoT dataset latent representation training of the LSTM model. The model demonstrates high accuracy for classifying classes 1 and 3, whereas for other courses, the low accuracy is attributed to the class imbalance problem. The model struggles with some classes, particularly those with lower frequencies. Misclassifications are more prevalent for these classes. Overall, the LSTM model, when trained on latent representations, can classify classes with high accuracy when well-balanced datasets are available.

NF-UNSW-NB15 Dataset

Classification report for NF-ToN-IoT dataset

	precision	recall	f1-score	support
0	0.80	0.67	0.73	6
1	1.00	1.00	1.00	24546
2	0.36	0.87	0.51	77
3	1.00	1.00	1.00	55148
4	0.89	0.67	0.76	51
5	1.00	0.01	0.02	97
6	0.00	0.00	0.00	41
7	0.62	0.41	0.49	32
8	0.00	0.00	0.00	2
accuracy			1.00	80000
macro avg	0.63	0.51	0.50	80000
weighted avg	1.00	1.00	1.00	80000

(a)

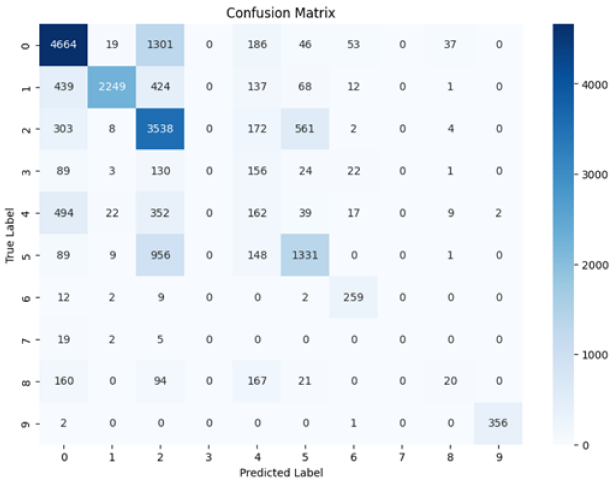
(b)

Figure 7: Classification report (a) and latent space training (b) for NF-ToN-IoT dataset

Classification report for NF-UNSW-NB15 Dataset

	precision	recall	f1-score	support
0	0.74	0.74	0.74	6306
1	0.97	0.68	0.80	3330
2	0.52	0.77	0.62	4588
3	0.00	0.00	0.00	425
4	0.14	0.15	0.15	1097
5	0.64	0.53	0.58	2534
6	0.71	0.91	0.80	284
7	0.00	0.00	0.00	26
8	0.27	0.04	0.07	462
9	0.99	0.99	0.99	359
accuracy			0.65	19411
macro avg	0.50	0.48	0.47	19411
weighted avg	0.66	0.65	0.64	19411

(a)



(b)

Figure 8: Classification report (a) and latent space training (b) for NF-UNSW-NB15 dataset

The effect of latent representations on classification can also result in less accuracy when we have highly imbalanced datasets, such as NF-UNSW-NB15, shown in Figure 8 (a) and (b) shows the classification report and confusion matrix of latent representation training of the classifier model. The model demonstrates an accuracy of 0.65, primarily due to the limited number of samples in certain classes. These results highlight the necessity for a well-balanced dataset to train the model effectively. For classes 0, 1, 6, and 9, the model exhibited high precision, recall, and F1 scores, indicating strong performance in these categories.

NF-CSE-CIC-IDS2018 Dataset

The significant advantage of training the LSTM model on latent representations can be analyzed on NF-CIC-IDS2018 as shown in Figure 9 (a) and (b). NF-CIC-IDS2018 dataset comprises 14 classes of attack vectors. The LSTM model shows high accuracy for classes 0 to 3 and 8 and 13. Overall, the model, when trained on these generic representations, achieves high accuracy in classifying the attacks. The model struggles with a few classes, particularly those with lower frequencies.

NF-UQ-NIDS Dataset

To illustrate the impact of training with latent representations across multiple datasets, Figure 10 presents a comprehensive classification report and the training performance of the Long Short-Term Memory (LSTM) model specifically applied to the NF-UQ-NIDS dataset. This dataset is notable for encompassing a diverse range of attack vectors, which are crucial for developing a robust classifier capable of accurately handling various types of network traffic scenarios. The classification report reveals a significant number of correct predictions, particularly evident along the diagonal of the confusion matrix for classes 0, 1, 2, and 3. These indicate that the model successfully identifies these classes with a high degree of accuracy. However, as observed in previous studies, the model's performance diminishes when dealing with datasets characterized by class imbalances. Such an imbalance can lead to skewed results, where the model may struggle to accurately predict minority classes, ultimately affecting the classifier's overall efficacy. These findings highlight the importance of addressing class imbalance to improve model performance across all courses in future iterations.

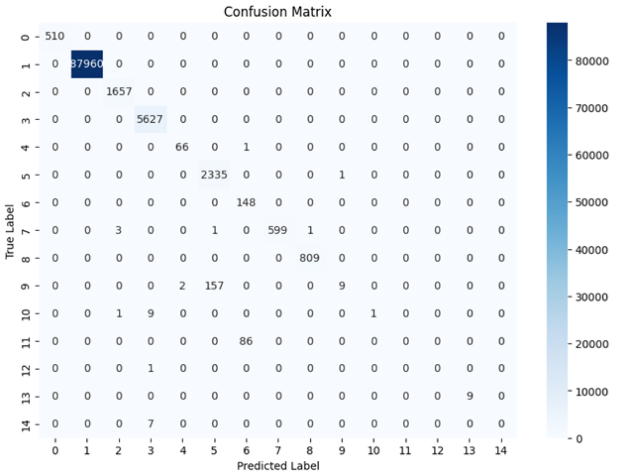
4.2 Multi-datasets classification performance

While analyzing individual latent spaces provides valuable insights into specific attack vectors, it may fail to generalize across diverse attack scenarios. The proposed approach is extended to overcome the generalization in classifying various attack vectors. By integrating multiple latent spaces into a unified representation, we aim to enhance the model's ability to learn a more comprehensive and discriminative feature space. This combined latent representation enhances classification performance by capturing shared and complementary patterns across latent datasets, resulting in a more robust and generalized detection framework. Latent spaces can be combined using various strategies, such as latent concatenation, latent average, and latent weighted sum techniques. These

Classification report for NF-CIC-IDS2018 dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	510
1	1.00	1.00	1.00	87960
2	1.00	1.00	1.00	1657
3	1.00	1.00	1.00	5627
4	0.97	0.99	0.98	67
5	0.94	1.00	0.97	2336
6	0.63	1.00	0.77	148
7	1.00	0.99	1.00	604
8	1.00	1.00	1.00	809
9	0.90	0.05	0.10	168
10	1.00	0.09	0.17	11
11	0.00	0.00	0.00	86
12	0.00	0.00	0.00	1
13	1.00	1.00	1.00	9
14	0.00	0.00	0.00	7
accuracy			1.00	100000
macro avg	0.76	0.67	0.67	100000
weighted avg	1.00	1.00	1.00	100000

(a)



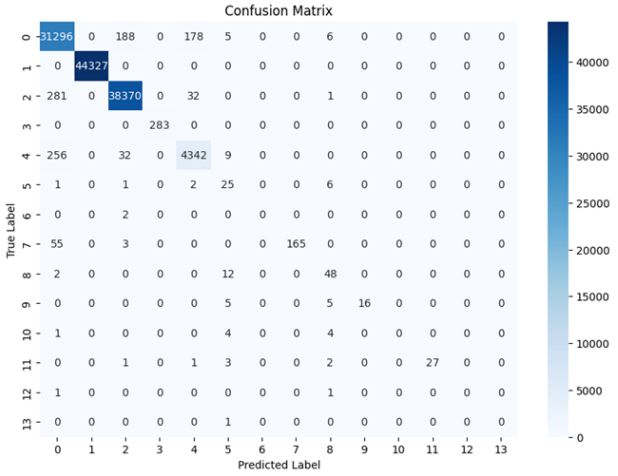
(b)

Figure 9: Classification report (a) and latent space training (b) for NF-CIC-IDS2018 dataset

Classification report for NF-UQ-NIDS dataset

	precision	recall	f1-score	support
0	0.98	0.99	0.99	31673
1	1.00	1.00	1.00	44327
2	0.99	0.99	0.99	38684
3	1.00	1.00	1.00	283
4	0.97	0.93	0.95	4639
5	0.43	0.09	0.14	35
6	0.00	0.00	0.00	2
7	0.88	0.99	0.93	223
8	0.56	0.87	0.68	62
9	0.94	0.62	0.74	26
10	0.00	0.00	0.00	9
11	1.00	0.79	0.89	34
12	0.00	0.00	0.00	2
13	0.00	0.00	0.00	1
accuracy			0.99	120000
macro avg	0.62	0.59	0.59	120000
weighted avg	0.99	0.99	0.99	120000

(a)



(b)

Figure 10: Classification report (a) and latent space training (b) for NF-UQ-NIDS dataset

techniques can combine latent values from multiple latent sources.

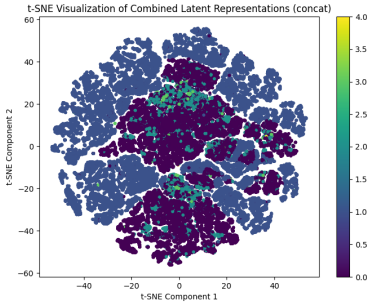
Provided with these techniques lies an advantage to the model training, as it can help the model to generalize well in classifying unseen attacks. The latent merge also has significant drawbacks, including misalignment of latent features and problems with feature heterogeneity. Figure 11 shows the effect of combining various latent spaces generated by the MMAE autoencoder to obtain a shared combined latent space. The following are the latent fusion techniques used in the proposed methodology:

- **Latent concat:** Computes the resultant latent by concatenating the latent vectors from different sources along the feature dimension.
- **Latent average:** Computes the element-wise average of the latent vectors.
- **Latent weighted sum:** Computes a weighted sum of the latent vectors, allowing some representations to have more influence.

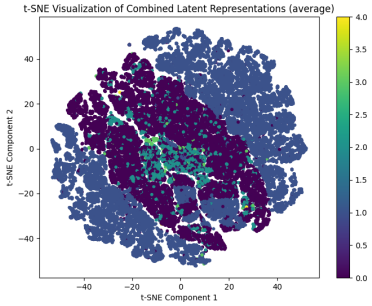
Table 3: Comparison of latent space fusion methods based on clustering quality and RF classifier accuracy.

Method	Silhouette Score	Intra Distance	Inter Distance	RF Accuracy
Concatenation	0.0674	8.5576	9.5855	0.9896
Averaging	0.0639	1.7048	1.9082	0.9626
Weighted Sum	0.2581	2.5383	3.5140	0.9846

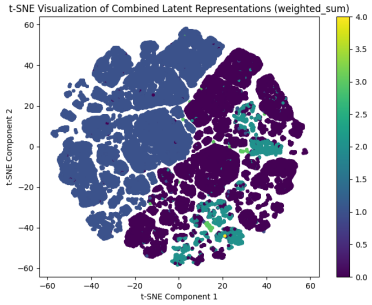
To demonstrate the effectiveness of the latent fusion techniques, the proposed methodology utilizes various metrics commonly employed in clustering techniques. Table 3 shows the clustering metrics used to decide the quality of the latent space combined. Using the Silhouette score, inter- and intra-cluster distance metrics, latent space formations can be justified. Even training a simple tree-based classifier, such as a Random Forest classifier, can help determine the quality of the latent space formed. A combined latent space is formed from these fusion methods for training the LSTM classifier. The results in Figure 12 show the confusion matrix of the classifier model during the training and testing phases on the combined latent space. The matrix shows significant improvement in model classification metrics, enhancing model robustness in classifying class distributions with larger samples in the latent space. By analyzing the LSTM classification reports on the combined latent space, we can conclude that the model is well-trained and effectively captures the patterns in the training data. The perfect classification of classes 3 and 4 might suggest that these classes are distinct or that the model has overfitted to these classes. The slight misclassifications in classes 0, 1, and 2 might indicate some overlap or similarity between these classes. Additionally, the testing phase results suggest that the model has successfully learned patterns from the training data and generalizes well to unseen data. The performance of the testing



(a)



(b)



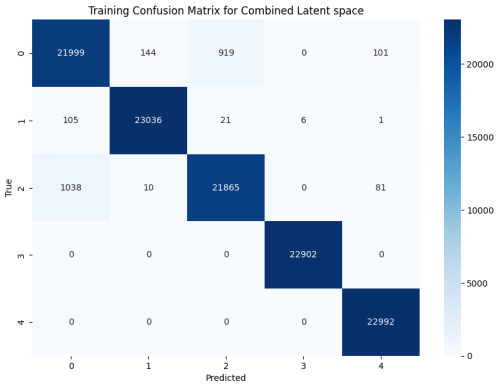
(c)

Figure 11: t-SNE visualization of different latent space fusion strategies: (a) Concat space, (b) Average space, and (c) Weighted sum space

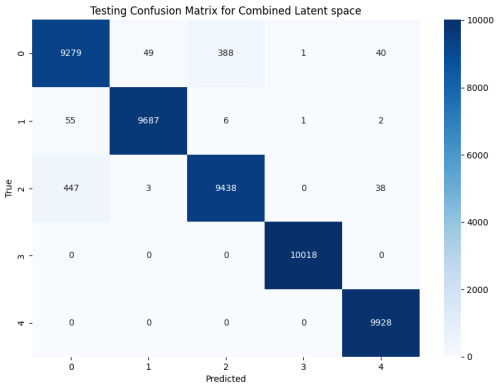
data is consistent with the training data, suggesting the model is robust. The only limitation of the above fusion techniques is that they require datasets of the same size and feature dimensionality. Due to these limitations, the LSTM model can classify well only classes 0 to 4 in Figure 12.

4.3 Fusion and evaluation strategy

In this section, the fusion strategies used in the proposed methodology are evaluated by training an LSTM model on diverse latent representations from multiple netflow datasets. At the feature level, latent space fusion is achieved through techniques such as concatenation, averaging, or weighted sum, each offering distinct trade-offs between information retention and dimensionality control. The temporal fusion is achieved through an LSTM network model, allowing the model to capture both spatial and temporal patterns to detect sequential anomalies. To evaluate the effectiveness of our approach, comprehensive metrics-accuracy, precision, recall, and F1-score are obtained from 5-fold validations, which were conducted on the combined latent space. Figure 14(a) to (c) illustrate the robustness of our proposed model on each of the combined latent spaces.



(a)



(b)

Figure 12: LSTM Classification report: (a) Training phase and (b) Testing phase

Table 4: Statistical comparison of classification accuracy using paired t-tests between latent space fusion strategies.

Comparison	t-statistic	p-value	Significance
Concat vs Average	21.2687	0.000029	Statistically significant
Concat vs Weighted sum	0.5200	0.630522	Less significant

Evaluation of the LSTM classifier on these combined latent spaces is tested using 5 5-fold cross-validation operation. All three fusion methods performed consistently well on the majority of the classes (0, 1, 2), which were fused from Netflow datasets with high true positive rates and minimal misclassifications. These results show that fused latent features are effective in capturing dominant traffic behaviors. However, limitations exist due to the minority classes (3 and 4), where the concatenation and weighted sum approaches perform poorly for class 3 and fail to classify class 4, yielding zero true positives across all folds. In contrast, the averaging method offers a slight but consistent improvement in detecting class 3, achieving marginally higher true positives, while still failing on class 4. Therefore, while all three techniques are suitable for large-scale classification tasks, averaging is slightly more advantageous in scenarios with imbalanced datasets or limited samples, particularly when rare anomalies must be detected alongside dominant classes. Figure 13 shows the comparison of accuracies obtained from the LSTM classifier for three approaches used for latent merge.

Table 5: Comparative performance of earlier works on Multi-dataset IDS

Sl. No	Authors	Model Used	Dataset(s) Used	Performance
1	Popoola et al., [21]	LAE-BLSTM	IoT datasets	91.89% input reduction, real-time adaptability
2	Huang et al., [24]	MFFAN (1D-CNN + LSTM)	ISCXIDS2012, CICIDS2017	99% accuracy, interpretability challenges
3	Torre et al., [25]	CNN, LSTM, AE, DNN	Not specified	Feature space reduction, long-range dependency
4	Bovenzi et al., [26]	AE + DL models	IoT datasets	Robustness, generalization emphasized
5	Fox et al., [27]	1D-CNN, 2D-CNN, MLP	USTC-TFC2016	$>$ 97% accuracy, overfitting issues
6	Zhu et al., [28]	CMTSNN, BiLSTM, 1D-CNN	ToN, BoT, ISCX VPN	Superior classification metrics
7	Ghani et al., [29]	AE-based reduction	5G traffic datasets	Enhanced performance with AE
8	Manocchio et al., [30]	FlowTransformer	CICIDS-17, UNSW-NB15, KDD	50% model size reduction
9	Wang et al., [31]	Multi-modal perception model	Power terminal attack datasets	Dynamic detection and attack map reconstruction
10	Irfan et al., [34]	Unified multimodal NIDS	Custom dataset	Dataset creation for multimodal IDS
11	L. Yu et al., [35]	1D-CNN + GRU	CICIDS2017, NSL-KDD	High performance on imbalanced datasets
12	Kiflay et al., [36]	Multimodal NIDS	UNSW-NB15	High accuracy, SHAP-based explainability
13	**Proposed Work**	**MMAE + LSTM**	Netflow datasets-NSW-NB15,BoT-IoT,ToN-IoT,CICIDS2018, UQ-NIDS	Up to 98.96% accuracy, robust generalization

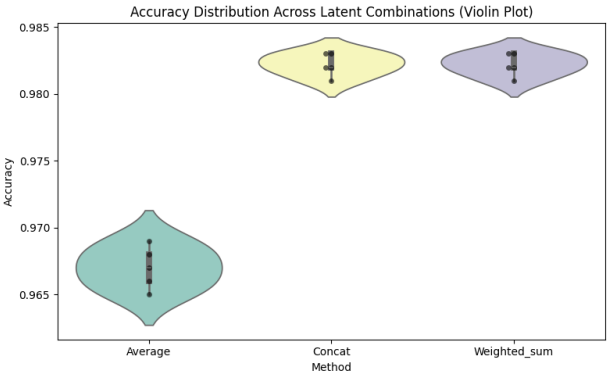


Figure 13: Violine plots for various latent space accuracies

5 Experimental setup and results

The experimental setup involves training a Multi-Modal Autoencoder (MMAE) and Long Short-Term Memory (LSTM) model on five benchmark NetFlow datasets to detect network traffic anomalies. The proposed work was tested on HPE ProLiant DL380 Gen11 with dual NVIDIA L40 GPUs, 96-core Xeon CPU, 503.6 GB RAM running in Ubuntu 24.04 LTS system. Each dataset undergone pre-processing, including normalization and encoding, followed by feature grouping into ”Entities” and ”Quantities.” The MMAE compresses 45 input features into a 32-dimensional latent space, which is then fused using concatenation, averaging, or weighted sum techniques. These fused representations are fed into an LSTM model to capture temporal patterns and classify traffic as normal or anomalous. The models are trained using the Adam optimizer with a learning rate of 0.001, a batch size of 128, and early stopping based on validation loss. Evaluation metrics include accuracy, precision, recall, F1-score, and ROC-AUC. To compare our model results with other baseline models, a comparative analysis against other datasets are considered from earlier works is listed in Table 5 and model performance is validated towards robustness and generalization.

Table 6 shows the hyperparameters used for proposed Multi-modal Autoencoder with LSTM model for training and testing phases.

Table 6: Key hyperparameters used for MMAE, LSTM model and Latent space fusion techniques

Component	Hyperparameter	Value	Description
Multi-modal Autoencoder	Latent_dim	32	Size of combined latent vector
	Hidden_dim	128	Neurons in decoder hidden layer
	Activation_function	ReLU	Non-linearity in decoder
	Loss_function	MSELoss	Measures reconstruction error
	Optimizer	Adam	Optimization algorithm
	learning_rate	0.001	Step size in parameter updates
	num_epochs	50	Number of training iterations
LSTM Classifier	LSTM units	32	Number of hidden units in LSTM layer
	Activation (output layer)	softmax	Multiclass probability estimation
	Loss_function	categorical_crossentropy	Suitable for one-hot encoded targets
	Optimizer	Adam	Optimization algorithm
	Training setup	epochs=10, batch_size=32	Number of passes and mini-batch size
	Hyperparameter tuning	Keras Tuner	Automated selection of optimal configuration
Netflow data selection	min_rows truncation	Min across all inputs	Ensures equal sample size
Combination Strategy	Fusion Methods	concat, average, weighted_sum	Technique to merge latent spaces
	Weights (weighted sum)	[0.6, 0.4]	Modality importance weighting

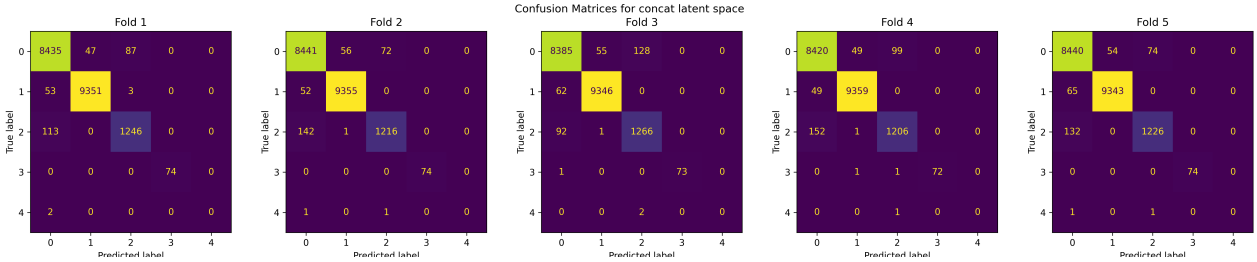
To validate the results of our model, a statistical paired t-tests were performed and results are tabulated in Table 4 between the various fusion techniques. A fusion technique which yields $p - value < 0.05$ was considered statistically significant to assess the robustness of the LSTM model.

6 Conclusion and future work

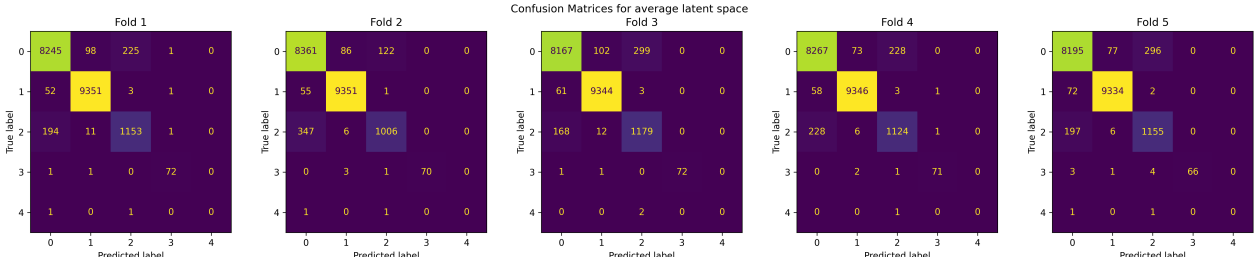
In conclusion, our study addresses the challenge of detecting anomalies in encrypted NetFlow traffic, where traditional Intrusion Detection Systems (IDS) often fall short due to limited visibility and evolving attack patterns. Also, the design of distributed or federated IDS depends on effective averaging of quality features from multi-dataset or data sources. To overcome this, a combination of Multi-Modal Autoencoder (MMAE) and Long Short-Term Memory (LSTM) networks were designed to generate a combined latent space for multi-datasets. The MMAE effectively compresses and harmonizes features from multiple NetFlow datasets into a unified latent space, which is then used by the LSTM for anomaly classification. The model was evaluated on five distinct NetFlow datasets—NF-UNSW-NB15, NF-BoT-IoT, NF-ToN-IoT, NF-CSE-CIC-IDS2018, and NF-UQ-NIDS—demonstrating high classification accuracy across individual datasets. When latent features from these datasets were fused using techniques like concatenation and weighted sum, the multi-dataset classification performance improved significantly, achieving up to 98.96% accuracy.

Statistical tests validated the robustness and significance of the fusion strategies, confirming that the proposed multi-modal approach generalizes well across diverse network environments. Future work in the direction of refining the model architecture to support incremental retraining for emerging threat patterns can be significant for real-time, scalable and privacy-preserving IDS deployment. Despite the promising results of the proposed MMAE-LSTM framework for multi-dataset NetFlow anomaly detection, following limitations should be acknowledged:

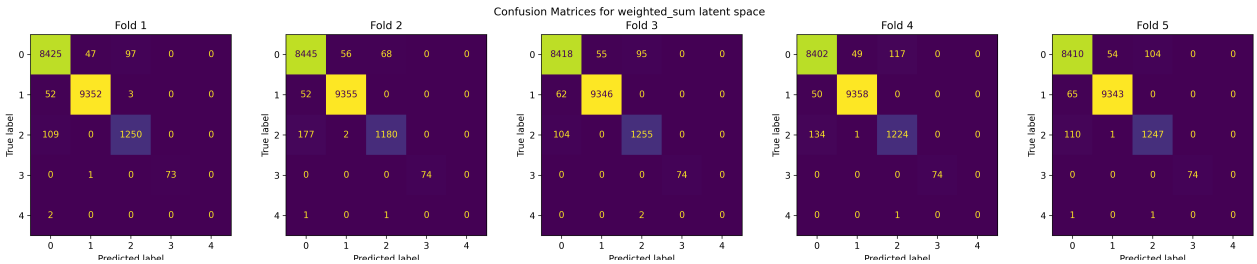
1. **Dataset Imbalance:** Some datasets used in the study exhibit significant class imbalance, which may impact the model’s ability to accurately detect rare or minority attack types.
2. **Fusion Strategy Constraints:** Although fusion methods



(a) Confusion Matrices for Concat Latent Space across 5-Fold Cross-Validation



(b) Confusion Matrices for Average Latent Space across 5-Fold Cross-Validation



(c) Confusion Matrices for Weighted Sum Latent Space across 5-Fold Cross-Validation

Figure 14: Comparison of confusion matrices across latent integration strategies over 5-fold cross-validation.

such as concatenation and weighted sum performed well, they may not effectively capture complex interdependencies between latent representations from different datasets.

3. **Computational Overhead:** Integrating multiple datasets and training deep architectures such as MMAE and LSTM increases computational load, potentially limiting applicability in resource-constrained environments.

References

[1] Garcia S., Parmisano A. and Erquiaga M. J. (2020) IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set], *Zenodo*, `https://doi.org/10.5281/zenodo.4743746`.

[2] MalwareBazaar (2025) Malware sample exchange, *MalwareBazaar Platform*, [Online]. Available: `https://bazaar.abuse.ch`.

[3] Bogatinovski, J. and Nedelkoski, S. (2021). Multi-source anomaly detection in distributed IT systems. *Lecture Notes in Computer Science*, pp. 201–213. `https://doi.org/10.1007/978-3-030-76352-7_22`.

[4] CICFlowMeter (n.d.) CICFlowMeter: A Network Traffic Flow Generator, *GitHub Repository*, [Online]. Available: `https://github.com/ahlashkari/CICFlowMeter`.

[5] Cisco Systems (n.d.) Cisco IOS NetFlow, *Cisco Systems*, [Online]. Available: `https://www.cisco.com/go/netflow`.

[6] NFStream (n.d.) NFStream: Flexible Network Data Analysis Framework, *GitHub Repository*, [Online]. Available: `https://github.com/nfstream/nfstream`.

[7] Sarhan M., Layeghy S., Moustafa N. and Portmann M. (2021) NetFlow datasets for machine learning-based network intrusion detection systems, *Lecture Notes in Computer Science*, Springer, pp. 117–135, `https://doi.org/10.1007/978-3-030-72802-1_9`.

[8] Sasi T., Lashkari A. H., Lu R., Xiong P. and Iqbal S. (2024) An efficient self attention-based 1D-CNN-LSTM network for IoT attack detection and identification using network traffic, *Journal of Information and Intelligence*, Elsevier, https://doi.org/10.1016/j.jiixd.2024.09.001.

[9] Verkerken M. et al. (2023) A novel multi-stage approach for hierarchical intrusion detection, *IEEE Transactions on Network and Service Management*, IEEE, pp. 3915–3929, https://doi.org/10.1109/TNSM.2023.3259474.

[10] Zhao X., Miao W., Yuan G., Jiang Y., Zhang S. and Li Q. (2024) Abnormal traffic detection system based on feature fusion and sparse transformer, *Mathematics*, MDPI, pp. 1643, https://doi.org/10.3390/math12111643.

[11] Lin Y.-D., Wang Z.-Y., Lin P.-C., Nguyen V.-L., Hwang R.-H. and Lai Y.-C. (2022) Multi-datasource machine learning in intrusion detection: Packet flows, system logs and host statistics, *Journal of Information Security and Applications*, Elsevier, pp. 103248, https://doi.org/10.1016/j.jisa.2022.103248.

[12] Li P., Pei Y. and Li J. (2023) A comprehensive survey on design and application of autoencoder in deep learning, *Applied Soft Computing*, Elsevier, pp. 110176, https://doi.org/10.1016/j.asoc.2023.110176.

[13] Aldhaheri A., Alwahedi F., Ferrag M. A. and Battah A. (2024) Deep learning for cyber threat detection in IoT networks: A review, *Internet of Things and Cyber-Physical Systems*, Elsevier, pp. 110–128, https://doi.org/10.1016/j.iotcps.2023.09.003.

[14] Liang P. P., Zadeh A. and Morency L. P. (2024) Foundations & trends in multimodal machine learning: Principles, challenges, and open questions, *ACM Computing Surveys*, ACM, https://doi.org/10.1145/3656580.

[15] Thakkar A. and Lohiya R. (2020) A review of the advancement in intrusion detection datasets, *Procedia Computer Science*, Elsevier, pp. 636–645.

[16] Geng X., Liu H., Lee L., Schuurmans D., Levine S. and Abbeel P. (2022) Multimodal masked autoencoders learn transferable representations, *arXiv preprint*, [Online]. Available: https://arxiv.org/abs/2205.14204.

[17] Gioacchini L., Mellia M., Drago I., Houidi B. and Rossi D. (2022) Learning generic multi-modal representations from network traffic for machine learning tasks, *SSRN Electronic Journal*, [Online]. Available: https://ssrn.com/abstract=4524861.

[18] Alkenani J. and Nickray M. (2025) Enhancing network QoS via attack classification using convolutional recurrent neural networks, *Informatica (Slovenia)*, Informatica, pp. 1–10, https://doi.org/10.31449/inf.v49i2.7637.

[19] Sharafaldin I., Habibi Lashkari A. and Ghorbani A. A. (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization, *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, SCITEPRESS, Funchal, Madeira, Portugal, pp. 108–116, https://doi.org/10.5220/0006639801080116.

[20] Shaukat K., Luo S., Varadharajan V., Hameed I. A. and Xu M. (2020) A survey on machine learning techniques for cyber security in the last decade, *IEEE Access*, IEEE, pp. 222310–222354, https://doi.org/10.1109/ACCESS.2020.3041951.

[21] Popoola S. I., Adebisi B., Hammoudeh M., Gui G. and Gacanin H. (2021) Hybrid deep learning for botnet attack detection in the Internet-of-Things networks, *IEEE Internet of Things Journal*, IEEE, pp. 4944–4956, https://doi.org/10.1109/JIOT.2020.3034156.

[22] Sarhan M., Layeghy S. and Portmann M. (2022) Feature analysis for machine learning-based IoT intrusion detection, *arXiv preprint*, [Online]. Available: https://arxiv.org/abs/2108.12732, [Accessed: Oct. 29, 2024].

[23] Nguyen L. G. and Watabe K. (2022) Flow-based network intrusion detection based on BERT masked language model, *Proceedings of the CoNEXT Student Workshop 2022*, ACM, pp. 7–8, https://doi.org/10.1145/3565477.3569152.

[24] Huang W. et al. (2022) MFFAN: Multiple features fusion with attention networks for malicious traffic detection, *Proceedings of the 2022 IEEE 21st International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, pp. 391–398, https://doi.org/10.1109/TrustCom56396.2022.00061.

[25] Torre D., Mesadieu F. and Chennamaneni A. (2023) Deep learning techniques to detect cybersecurity attacks: A systematic mapping study, *Empirical Software Engineering*, Springer, https://doi.org/10.1007/s10664-023-10302-1.

[26] Bovenzi G., Aceto G., Ciuonzo D., Montieri A., Persico V. and Pescapé A. (2023) Network anomaly detection methods in IoT environments via deep learning: A fair comparison of performance and robustness, *Computers and Security*, Elsevier, https://doi.org/10.1016/j.cose.2023.103167.

[27] Fox G. T. and Boppana R. V. (2023) On early detection of anomalous network flows, *IEEE Access*, IEEE, pp. 68588–68603, https://doi.org/10.1109/ACCESS.2023.3291686.

[28] Zhu S., Xu X., Gao H. and Xiao F. (2023) CMTSNN: A deep learning model for multiclassification of abnormal and encrypted traffic of Internet of Things, *IEEE Internet of Things Journal*, IEEE, pp. 11773–11791, https://doi.org/10.1109/JIOT.2023.3244544.

[29] Ghani H., Salekzamankhani S. and Virdee B. (2024) Critical analysis of 5G networks' traffic intrusion using PCA, t-SNE, and UMAP visualization and classifying attacks, *Lecture Notes in Networks and Systems*, Springer, pp. 421–437, https://doi.org/10.1007/978-981-99-6544-1_32.

[30] Manocchio L. D., Layeghy S., Lo W. W., Kulatilleke G. K., Sarhan M. and Portmann M. (2024) FlowTransformer: A transformer framework for flow-based network intrusion detection systems, *Expert Systems with Applications*, Elsevier, https://doi.org/10.1016/j.eswa.2023.122564.

[31] Wang R., Zou X., Li Y., Li F., Liu J. and Wang R. (2024) Research on power terminal attack detection technology based on ATT&CK multi-modal perception, *Proceedings of the 3rd International Conference on Cryptography, Network Security and Communication Technology (CNSCT 2024)*, ACM, pp. 287–294, https://doi.org/10.1145/3673277.3673327.

[32] Neloy A. A. and Turgeon M. (2024) A comprehensive study of auto-encoders for anomaly detection: Efficiency and trade-offs, *Machine Learning with Applications*, Elsevier, pp. 100572, https://doi.org/10.1016/j.mlwa.2024.100572.

[33] Torabi S., Klisura D., Khoury J., Bou-Harb E., Assi C. and Debbabi M. (2024) Internet-wide analysis, characterization, and family attribution of IoT malware: A comprehensive longitudinal study, *IEEE Transactions on Dependable and*

Secure Computing, IEEE, pp. 1–14, https://doi.org/10.1109/TDSC.2024.3454573.

[34] Khan I., Bastian N., Wali S. and Farrukh Y. A. (2024) Unified multimodal network intrusion detection systems dataset, *IEEE Dataport*, IEEE, https://doi.org/10.21227/d8at-gb29.

[35] Yu L., Xu L. and Jiang X. (2024) A high-performance multimodal deep learning model for detecting minority class sample attacks, *Symmetry*, MDPI, https://doi.org/10.3390/sym16010042.

[36] Kiflay A., Tsokanos A., Fazlali M. and Kirner R. (2024) Network intrusion detection leveraging multimodal features, *SSRN Electronic Journal*, [Online]. Available: https://ssrn.com/abstract=4629013.

[37] Palakurti N. R. (2024) Challenges and future directions in anomaly detection, *Practical Applications of Data Processing, Algorithms, and Modeling*, IGI Global, pp. 269–284, https://doi.org/10.4018/979-8-3693-2909-2.ch020.