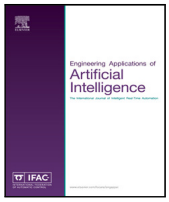
Engineering Applications of Artificial Intelligence 126 (2023) 107169

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



SeNet-I: An approach for detecting network intrusions through serialized network traffic images

Yasir Ali Farrukh [a], Syed Wali [a], Irfan Khan [a,*], Nathaniel D. Bastian [b,*]

[a] Clean and Resilient Energy System Lab (CARES), Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA
[b] Army Cyber Institute, Department of Electrical Engineering and Computer Science, United States Military Academy, West Point, NY, USA

ARTICLE INFO

Keywords:
Cybersecurity
Network intrusion detection
Machine learning
Serialized image classification
Convolutional neural network

ABSTRACT

The exponential growth of the internet and inter-connectivity has resulted in an extensive increase in network size and the corresponding data, which has led to numerous novel attacks that pose significant challenges to network security. However, conventional network security approaches predominantly rely on the metadata of network traffic, utilized in numeric form, which is becoming ineffective against new attacks that hide within the content of the traffic. Therefore, it raises the need for security systems to adapt to the changing dynamics of network attacks. To address this issue, we propose a new approach called SeNet-I that leverages computer vision capabilities to combine low-level features and develop a more abstract and high-level representation of network traffic without requiring feature engineering. The proposed approach utilizes the raw network traffic information and transforms it into serialized three-channel images, which are employed as input to a proposed deep concatenated convolutional neural network model. Additionally, SeNet-I can easily incorporate packet level information, which is often challenging for conventional approaches due to its high dimensionality. To demonstrate the effectiveness of the proposed approach, we tested SeNet-I on both packet-based and flow-based network traffic, comparing it with current state-of-the-art methods and different image-based approaches. With F1 scores of 96% and 83% achieved in the multi-class classification of flow-based and packet-based network intrusion detection, our proposed approach outperformed other existing methods in the literature. Lastly, we discussed the advantages and limitations of the proposed method.

1. Introduction

The advancement of connectivity through the Internet has brought about a profound change in the way people communicate and access information. This widespread use of the Internet has created a more connected world where people can communicate and collaborate with others in real-time from anywhere in the world (Munirathinam, 2020). The result is a global economy where businesses can reach new customers and partners all around the world. In addition, the rise of the Internet of Things (IoT) has also enabled the connection of everyday devices and objects, leading to the creation of smart homes, cities, and industries that are more efficient and sustainable.

However, the benefits of connectivity also come with downsides, and one of the major concerns is privacy and security (Khraisat and Alazab, 2021). The vast amount of personal and sensitive information that is shared and stored online has made it easier for cyber criminals to access and exploit it, leading to an increase in data breaches and cyber-attacks. These attacks can result in the loss of personal information and financial damage, intellectual property theft, and business disruption (Iasiello, 2021). The cost of cyber-attacks is increasing as businesses and governments spend billions of dollars each year to protect against these threats. Additionally, cyber criminals are becoming increasingly sophisticated, using advanced technologies such as artificial intelligence and machine learning to automate their attacks and make them more effective (Tao et al., 2021). Therefore, while the advancement of connectivity has created a more connected world that is constantly evolving and shaping the way we live, work, and communicate, it also has brought about security challenges that require attention and investment.

As a result, it is critical to employ modern methods for security systems as well. A Network Intrusion Detection System (NIDS) is often considered as a viable option for protecting against network-based attacks (Hassan et al., 2022). NIDS can be broadly categorized into two categories based on the detection method: Signature-based and Anomaly-based Intrusion Detection System (IDS) (Khraisat et al., 2019).

* Corresponding authors.
E-mail addresses: yasir.ali@tamu.edu (Y.A. Farrukh), syedwali@tamu.edu (S. Wali), irfankhan@tamu.edu (I. Khan), nathaniel.bastian@westpoint.edu (N.D. Bastian).

https://doi.org/10.1016/j.engappai.2023.107169
Received 21 April 2023; Received in revised form 9 July 2023; Accepted 14 September 2023
Available online 27 September 2023

Signature-based Intrusion Detection Systems (SIDS) protect against malicious network activity by inspecting network packets and comparing them to a known database of attack packet attributes. One of the limitations of SIDS is that they are ineffective in detecting zero-day attacks (Tas and Yahyaoui, 2022). Anomaly-based Intrusion Detection System (AIDS) analyzes the network traffic and compares it to anticipated network activity to determine whether an alert should be sent. Though they are not immediately tolerant of new behaviors, AIDS offers superior protection against zero-day attacks (Hassan et al., 2022).

Although AIDS uses different methods for mapping the normal behavior of the network, lately, much emphasis has been placed on the use of machine learning (ML) approaches. ML algorithms produce impressive results in fields where it is difficult to provide a set of rules for their procedures (Chalé and Bastian, 2022). Because ML techniques work differently, they use the data to understand patterns between occurrences (Dimitrios Tsokos Supervisor and Georgios, 2021). As a result, ML techniques become adaptive and dynamic in a variety of settings without the need for human intervention.

Previous techniques for Network Intrusion Detection Systems (NIDS) based on Machine Learning (ML) approaches have predominantly relied on utilizing network traffic metadata in the numeric form, also known as flow-based NIDS. However, there are some methods that incorporate network traffic payload data in either numeric or textual form using Natural Language Processing (NLP) techniques (Jallad, 2022), which are referred to as packet-based NIDS. While conventional approaches based on metadata have been used in the past, they are now becoming less effective as they rely solely on the header information of packets and do not consider the payload data. Moreover, these approaches are not suitable for high-dimensional data and struggle to maintain classification performance when incorporating network traffic payload information. Similarly, traditional ML approaches have failed to meet the expectations of users due to the continuous growth of digital technology and the increasing diversity of cyber-attacks (Cao et al., 2022).

Therefore, it is essential to adopt innovative techniques that can capture the relevant information from network traffic and provide improved detection accuracy for NIDS. The proposed packet-based NIDS approach is one such technique that has shown promising results in capturing the relevant information from network traffic and improving the overall effectiveness of intrusion detection systems. By leveraging the latest advancements in computer vision techniques, we can enhance the accuracy and efficiency of NIDS, making them more effective in detecting a wide range of cyber-attacks.

In recent years, deep convolutional neural networks (CNN), a machine learning approach based on image data have been widely utilized in the domain of computer vision as it has achieved striking results. The capability of combining low-level features to develop a more abstract and high-level representation of the input is one of the reasons for its vast adaptation. As CNN can better extract spatial features from images automatically (Han et al., 2020), it can address one of the key challenges faced by textual and numeric-based machine learning approaches that is limited data representation. These approaches cannot handle large amounts of information or complex relationships between features, which is likely the case when you treat network traffic's payload data as your feature vector. Additionally, CNN are designed to automatically extract features from the raw elements of images (Sharma et al., 2019), reducing the amount of work needed for manual feature engineering, an essential aspect in textual and numeric-based approaches. Therefore, utilizing CNN and transforming network data into images can be an effective approach for NIDS, especially when dealing with the network's payload data which has huge dimensionality.

In this work, we propose an approach *SeNet-I* for NIDS that is based on the concept of transforming network traffic into images. The fundamental principle of SeNet-I is to leverage the capabilities of computer vision within the realm of network security. Although some work in the literature uses the transformation of network traffic to images for CNN implementation (Cao et al., 2022), to the best of our knowledge, most of the work is limited to grayscale images (single channel images) or utilizes network metadata (flow data) for image transformation. The novelty of our proposed approach is that we introduced a unique way for transforming network traffic, especially payload data of network traffic, to a serialized image, i.e., encapsulating a series of network traffic samples into a single three-channeled image. The transformed image is subsequently utilized as an input for our proposed deep concatenated CNN model based on input and output concatenation. The intuition behind the proposed approach is that by incorporating serialized images, we can better extract spatial and temporal information from the network traffic without any feature engineering, which is lacking in conventional approaches based on the numeric and textual data form. The main contributions of this work are as follows:

(1) We introduce a novel approach *SeNet-I* for NIDS, which involves the transformation of network traffic into serialized three-channel images, followed by the use of a proposed deep concatenated CNN model.
(2) We propose a deep concatenated CNN model based on the concept of input and output concatenation specifically for our proposed serialized three-channel images.
(3) We conduct a thorough literature review of NIDS that is focused on the input format, which includes text, numeric, and image data.
(4) We illustrate the transformation of flow-based and packet-based network traffic data into grayscale, RGB, serialized grayscale, and serialized RGB images, along with the computation of their performance when used as an input for a ML-based NIDS.
(5) We conduct a comprehensive comparison between various forms of transformed images and baseline models with our proposed approach to evaluate its effectiveness.
(6) We provide a quantitative comparative analysis for our proposed method with other state-of-the-art approaches based on different type of input data (text, image, and numerical based network traffic data).

The remainder of this work is structured as follows: Section 2 provides a brief background knowledge on different format of network traffic data and how they are captured, along with available NIDS datasets. Section 3 incorporated the related NIDS work-based on input formats of the proposed approaches, i.e., numerical, text, and image. Details about the adopted methodology, including the data processing, transformation procedure, and model architecture, are described in Section 4. Section 5 presents the results and comparative analysis. Furthermore, discussion and future work are discussed in Section 6. Lastly, Section 7 brings the paper to a conclusion.

2. Background

Typically, network traffic is captured either in flow-based or packet-based format. Packet-level network traffic is generally captured by mirroring ports on network devices, which encapsulates the complete information of payload. On the other hand, flow-based data are in more aggregated form and only contains metadata from network connections (Halisdemir et al., 2022).

2.1. Flow-based format

The flow-based format is a way of representing network traffic data in which network traffic is grouped into flows. A flow is defined as a set of packets that have a common source, destination, and protocol (Krupski et al., 2021). In other words, a flow represents the traffic between a specific source and destination that uses a particular protocol. Flow-based network data is a more compact format that primarily contains

Table 1
A list of widely utilized intrusion detection datasets in the literature.

Dataset	Accessibility	Release year	Format	Volume	Traffic	Balanced	Labeled flow	Labeled packets	Modern attacks
NSL-KDD	Public	1998	Other	150 k points	Emulated	No	Yes	–	No
DARPA	Public	1998	Packets, Logs	4.9 M points	Emulated	No	Yes	Yes	No
KYOTO 2006+	Public	2006	Other	93 M points	Real	No	Yes	–	No
UNIBS	On request	2009	Flow	79 k flows	Real	No	Yes	–	No
Botnet	Public	2010	Packet	14 GB packets	Emulated	No	Yes	Yes	No
ISCX 2012	Public	2012	Packet, Flow	2 M flows	Emulated	No	Yes	No	No
CIC DoS	Public	2012	Packet	4.6 GB packets	Emulated	No	Yes	Yes	No
CTU-13	Public	2013	Packet, Flow	81 M flows	Real	No	Yes	No	No
UNSW-NB15	Public	2015	Packet, Flow	2 M points	Emulated	No	Yes	No	Yes
CIC-IDS2017	Public	2017	Packet, Flow	4.6 GB packets	Emulated	No	Yes	No	Yes
CIC-IDS2018	Public	2018	Packet, Flow, Logs	450 GB logs	Emulated	No	Yes	No	Yes

metadata about the network connection. Flow-based data combine all packets that share some attributes during a time interval into a single flow and typically excludes payload information. The widely utilized standard for matching characteristics in flow-based data is the default five-tuple definition, which includes the source IP address, source port, destination IP address, destination port, and transport protocol (Zhao et al., 2021). Flows can be in unidirectional and bidirectional formats. The flow-based format focuses on the high-level features of the network traffic, such as traffic volume, flow duration, and protocol type. Therefore, it allows for a quick and easy analysis of traffic patterns, anomalies, and potential security threats.

Moreover, flow-based network traffic can also be utilized to monitor network performance and optimize network resources by identifying bottlenecks and tracking resource utilization. However, it lacks complete information on network traffic as it neglects the payload data of the network. Many attacks, such as worms, ransomware, and Trojans, rely on payload delivery and might evade NIDS based on flow-based network traffic.

2.2. Packet-based format

Packet-based format is the representation of network traffic at the packet level. Each packet is analyzed individually, providing detailed information about its content, header, and payload. Packet-based network traffic is typically captured in packet capture (PCAP) format. This binary-based format supports nanosecond precision timestamps, although it may differ from implementation to implementation. Packet-based network traffic provides a more granular view of network activity, making it suited for low-level analysis activities like identifying certain attacks and vulnerabilities. On the other hand, the vast volume of data created by packet-based network traffic can make it difficult to manage and analyze, and the computing effort of processing each packet individually can make it unsuitable for high-level analytic tasks. Moreover, each packet's metadata depends on the transport protocol and network. Therefore, it is not easy to maintain a common feature space for a machine learning model. This is one reason that most of the deployed machine learning models are based on the flow-based format of network traffic. Additionally, no publicly available datasets of properly labeled packet-based network traffic exist. However, many well-known datasets have packet-based network traffic available in addition to flow-based format. But the problem arises at the labeling end as these raw packets are not labeled like flow-based network traffic.

2.3. Available datasets

Several publicly available datasets for NIDS exist in the literature, but most are outdated due to the changing nature of malicious and benign network traffic. Moreover, most NIDS datasets are based only on network traffic metadata (flow-based format). Although some datasets also provide raw packets, the problem is that these packets are not labeled. The unavailability of labeled packet-based network traffic is a significant issue in the packet-based NIDS. As a result, packet-based approaches are not that much popular. Since our work is more specifically focused on packet-based NIDS, we have utilized our developed tool (Farrukh et al., 2022) to label the raw packets provided by available datasets. The detail of processing the data is provided in Section 3. On the other hand, a list of widely utilized datasets based on the data format and other various features is illustrated in Table 1. The features evaluated for each dataset include accessibility (whether the dataset is publicly available or not), release year (the year in which the dataset was published), format (the different forms in which the dataset is available), volume (the size of the dataset), traffic (whether the traffic is emulated or real), balance (whether the data samples for each class are balanced), labeled flow (whether the flow-based data is labeled), labeled packet (whether the packet-based data is labeled), modern attacks (whether the dataset represents dynamics modern attacks or not).

3. Related works

A lot of research has been conducted in the field of NIDS using ML techniques. However, most of the methods proposed in the literature rely on packet header data to derive features for training the model, commonly referred to as flow-based features. These methods generally employ numerical representations for training ML models. Nonetheless, a few techniques use raw packet data or packet's payload data to integrate ML models based on textual or image data.

In this section, we provide a comprehensive review of the literature, focusing solely on the various techniques employed in NIDS. It is important to note that our review is limited to NIDS approaches and does not encompass the broader field of machine learning. By narrowing our focus, we aim to provide readers with a thorough understanding of the existing research conducted specifically in the NIDS domain, particularly in relation to packet-based and flow-based data. To facilitate this, we categorize the approaches based on the type of input utilized, highlighting the unique challenges and advancements within each category. The reviewed approaches are classified based on the type of input format used in the proposed methods, including image data, textual data, and numerical data. Fig. 1 demonstrates the categorization of different input forms utilized in NIDS, as well as the corresponding format of network traffic data used for each input.

3.1. Numeric-based approaches

Numerical attributes of network traffic, including their statistical parameters or flow-level information, have been widely used to develop ML-based IDS. Several studies have employed ensemble learning techniques to develop IDS based on such numerical attributes. These ensemble learners have demonstrated substantial enhancement over individual learning models because the primary objective of the ensemble method is to enhance weak classifiers that may include Bayesian algorithms, tree learners, neural networks, or other forms of weak learning algorithms. Recently, Tama and Lim (2021) provided an comprehensive overview of various ensemble learning models that are utilized in the

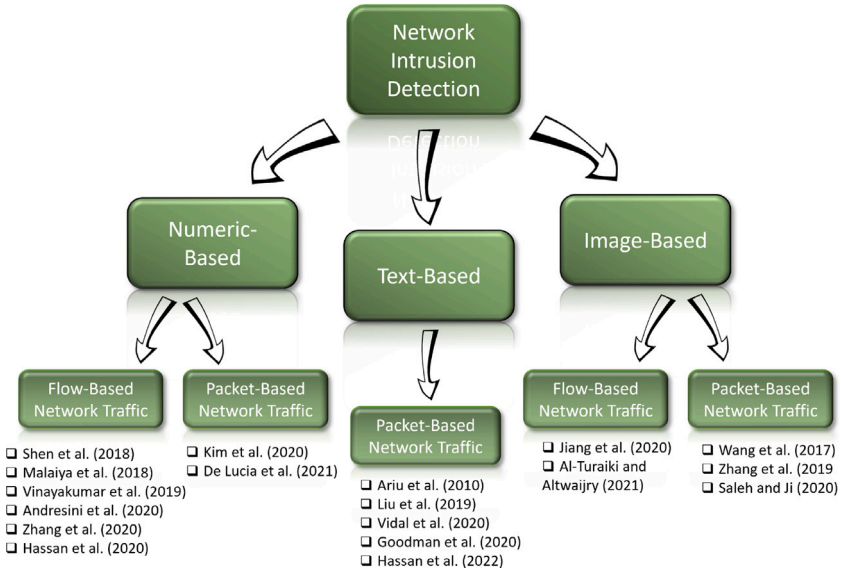


Fig. 1. Categorization of NIDS based on input type, along with the associated format of network traffic. The relevant research articles associated with each type of input are also listed.

NIDS domain. Shen et al. (2018) proposed an IDS utilizing an ensemble approach, where (Extreme Learning Machine) ELM was selected as a base classifier. To enhance the proposed method, the ensemble pruning phase was optimized using a BAT optimization algorithm. Results of the experiments demonstrated that the ensemble of multiple ELMs outperformed individual ELM in terms of performance. On the other hand, Vinayakumar et al. (2019) presented a hybrid scalable deep neural network (DNN) framework, called Scale-Hybrid-IDS-AlertNet, which is capable of identifying intrusions at both host and network levels. However, the proposed complex model encounters a low detection rate problem for minority classes. Similarly, Malaiya et al. (2018) conducted an empirical study on various IDS models, which included fully connected networks, Variational Autoencoder (VAE), and Sequence-to-Sequence (Seq2Seq) architectures. These models were assessed on different datasets, such as NSL-KDD, Kyoto Honeypot, UNSW-NB15, CIC-IDS2017, and MAWILab traces. Results indicated that Seq2Seq exhibited the best performance, although it had a higher training overhead than other models.

In terms of 1D-CNN network, Andresini et al. (2020) introduced a multistage model that incorporates AE technique and includes a 1D convolution layer and two stacked fully connected layers. The proposed methodology was evaluated on the flow-based network traffic of KDD Cup'99, UNSW-NB15, and CIC-IDS2017 datasets, and was found to outperform other deep learning models in terms of performance. Similarly, Zhang et al. (2020) proposed a novel intrusion detection model, SGM-CNN, that leverages SMOTE oversampling and GMM-based clustering under-sampling with CNN to detect attacks in network traffic. The model utilizes a 1D CNN architecture based on flow-based network traffic whereas, Kim et al. (2020) proposes anomaly detection scheme AI-IDS which employs CNN, LSTM, and spatial feature learning for packet-based network traffic. The evaluations were carried out using packet level features, which is a notable advantage of this approach. Following the architecture of 1D-CNN, De Lucia et al. (2021) proposed an approach utilizing the header and payload data. The authors utilized the raw packet bytes in conjunction with a specified feature space. However, only two protocols out of 130 were addressed in this work for UNSW-NB15. Lastly, Hassan et al. (2020) proposed a hybrid network model that combines a CNN and weight dropped LSTM (WDLSTM) to detect intrusions. The CNN is used to extract features from flow-based IDS data, while the WDLSTM preserves long-term dependencies between derived features to avoid over-fitting from recurrent connections.

3.2. Text-based approaches

Prior studies have utilized various techniques employing Natural Language Processing (NLP) for detecting anomalies in network traffic. However, these approaches typically operate on packet-based network traffic rather than flow-based traffic. The most frequently used methods include Bag-of-Words (BoW), N-grams, and Embeddings. Among these techniques, N-grams based approaches have been extensively studied, which measure the frequencies of occurrences for 256 possible byte values. These frequencies of bytes are utilized to construct a normal payload profile and are compared against incoming payloads to detect network attacks. Ariu and Giacinto (2010) presented HMMPayl that utilizes the n-gram technique on the payloads of normal traffic. This method employs a sliding window of n-bytes to extract sequences from the payload. A random subset of these sequences is used to train five Hidden Markov Model (HMM). The probability estimates obtained from each HMM model are combined, and a threshold value is used to classify a payload as either anomalous or normal. Similarly, Vidal et al. (2020) proposed EsPADA that also utilizes the n-gram technique with $n = 3$ to extract features from payloads. The extracted n-grams are stored in Counting Bloom Filters (CBF). To classify incoming packet payloads, each packet is compared to both normal and adversarial models, and a global similarity score (GSC) is computed. The packets are then classified based on their GSC scores.

In the context of deep learning techniques, Liu et al. (2019) introduced two deep learning models called PL-CNN and PL-RNN. For PL-RNN, the authors used the LSTM network, which contains a single hidden layer with 128 hidden states. The first n characters of the payloads are used for model training, where n is set to the average length of payloads in a given dataset. Another approach called Packet2Vec is proposed by Goodman et al. (2020) which employs a shallow neural network, specifically Word2Vec, to generate packet vectors. The authors utilize n-grams to create a dictionary, where $n = 2$ and the vocabulary size reaches 65,536 (2^{16}). To simplify the process, Packet2Vec limits the vocabulary size to the top $|v|$ most frequent n-grams, where $|v|$ is set to 50,000. The length of each packet vector is set to 128, and the packet vectors are ultimately utilized as features and classified using Random Forest (RF). Extending the concept of Word2Vec, Hassan et al. (2022) proposed PayloadEmbeddings that uses vector representations of bytes in network packet's payloads. The authors generated payload embeddings from the vector representations of bytes. These embeddings are then fed to a k-Nearest Neighbor (kNN) classifier to detect a network packet as anomalous or non-anomalous.

3.3. Image-based approaches

While the proposed techniques that utilize both numerical and textual data have shown potential, the rapid evolution of attack behaviors has rendered traditional methods less effective. In response, feature extraction approaches have been proposed as possible solutions to overcome these limitations. One such procedure involves transforming features into images, which has demonstrated promising results in various computer vision domains. Consequently, transforming network traffic into images appears to be a promising approach. Indeed, several studies have implemented this approach and used it to train CNN models for intrusion detection tasks.

Wang et al. (2017) proposed HAST-II architecture that employs deep CNN to acquire low-level spatial features and LSTM to capture high-level temporal features from packet-based network traffic. In this architecture, each packet is converted into a two-dimensional image by One-hot encoding (OHE), where the first n bytes of a network packet is transformed into an $m \cdot n$ two-dimensional image if the OHE vector is m dimensional. The generated grayscale image is then utilize to classify the network traffic. Similarly, Jiang et al. (2020) introduced a novel algorithm for network intrusion detection by combining hybrid sampling with deep hierarchical networks to enhance detection accuracy. The authors employed a CNN and BiLSTM architecture to detect flow-based network traffic, achieving a performance of 79% using their approach. Specifically, network traffic was transformed into an 11×11 matrix, and the model automatically extracted features through repeated multi-level learning, utilizing the benefits of deep learning. A deep CNN approach called MCNN-DFS was employed by Al-Turaiki and Altwaijry (2021) to transform flow-based network traffic into grayscale images of size 14×14. They used a two-step pre-processing strategy that involved converting categorical data into one-hot encoding and applying deep feature synthesis. The authors achieved an F-1 score of 81% for multi-class classification of the UNSW-NB15 dataset.

On the other hand, Zhang et al. (2019) presented a complex multi-layer IDS model utilizing CNN and gcForest. They introduced a novel P-Zigzag algorithm to convert raw packet-based network traffic into two-dimensional grayscale images. The proposed model consisted of two layers - a coarse-grained layer using an improved CNN model, GoogLeNetNP, for initial detection. Then a fine-grained layer using gcForest (caXGBoost) for further classification of abnormal classes into N-1 subclasses. The authors evaluated the proposed model on a combined dataset of UNSW-NB15 and CIC-IDS2017 and demonstrated that it outperforms single algorithms in terms of accuracy, detection rate, and false alarm rate. Similarly, A novel approach for transforming the payload of packets into images was presented by Saleh and Ji (2020). The authors created images using one of five possible mappings of a flow vector (1D) into a 2D matrix. The proposed mappings were linear, diagonal, waterfall, center spiral, and edge spiral. One of the objective of their experiment was to identify the optimal mapping, and the results showed that the classic linear mapping achieved the best performance. The grayscale images generated in this work had dimensions of either 17×17 or 25×25, which were then fed into a VGG-16 CNN.

4. Methodology

In this section, we will provide a brief overview of the experimental setting, selected datasets and their preprocessing. Then, we will explain our proposed methodology, which involves transforming network traffic into an image using different image types. While our approach primarily focuses on transforming packet-based network traffic (packet's payload), we also provide details on implementing our approach on flow-based data. Finally, we will present the details of our proposed deep concatenated CNN model and provide justification for why we chose this model.

4.1. Experimental setting

Our experimental setting encompasses three main evaluations to assess the performance and effectiveness of our proposed approach. In the first evaluation, we conducted experiments to evaluate the performance of packet-based and flow-based network traffic data when transformed into four different types of images: grayscale, RGB, serialized grayscale, and serialized RGB images. This allowed us to compare the performance of various image representations in capturing the relevant features of the network traffic.

In the next evaluation, we focused on comparing the performance of our proposed approach, which is based on serialized RGB images, with several baseline models using both packet-based and flow-based network traffic data. This comparative analysis enabled us to gauge the effectiveness and superiority of our approach in accurately classifying and detecting network intrusions.

Furthermore, we performed a comprehensive comparative analysis of our approach with other existing approaches in the literature that utilized numeric, textual, and image input forms for network traffic analysis. This comparison provided insights into the strengths and weaknesses of different methodologies and allowed us to position our approach in the broader context of network traffic analysis techniques.

To ensure fair comparison and evaluation, all experiments were performed in a similar testing environment. The experimentation and testing procedures were conducted on a system with an Intel(R) Core(TM) i7-11800H CPU running at 2.30 GHz, equipped with 64 GB of RAM and an 8 GB NVIDIA GeForce RTX 3070 graphics card. This standardized setup ensured consistency and reliability across the experimental evaluations, facilitating meaningful comparisons and accurate performance assessments.

4.2. Dataset prepossessing

Table 1 provides information on five publicly available datasets that contain both flow-based and packet-based data of network traffic. However, two of these datasets, namely UNSW-NB15 (Moustafa and Slay, 2015) and CIC-IDS201 (Sharafaldin et al., 2018), are considered to be the most modern and relevant for NIDS, as they reflect the current dynamics of network traffic. These datasets are widely used by researchers and practitioners in the field, and they contain a diverse representation of network traffic data, including various attack classes. Therefore, in this work, we have chosen to utilize only UNSW-NB15 and CIC-IDS2017 datasets to evaluate our proposed approach and method.

The individual data preprocessing steps for each data format are discussed in the subsequent subheading. Although we provide a brief explanation of data preprocessing in this section, we adopted the same data preprocessing as in our previous work (Farrukh et al., 2022), which should be referred to for complete details. For our experimentation, we split the data into a 70:30 training and testing ratio for both data formats.

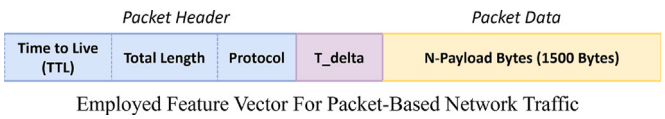


Fig. 2. An Illustration of feature vector utilized for training NIDS based on packet-based network traffic. Only three features: time to live, packet's total length and protocol has been utilized from packet's header. T-delta is the time difference between packets.

4.2.1. Flow-based dataset

Preprocessing the flow-based network traffic data of both datasets required several steps. Firstly, we removed missing, duplicated, and null values from the datasets, as well as any duplicated columns. We then removed certain features, including *FlowID, Source IP, Destination IP, Start time, Last time, and Time Stamp*, as these features could

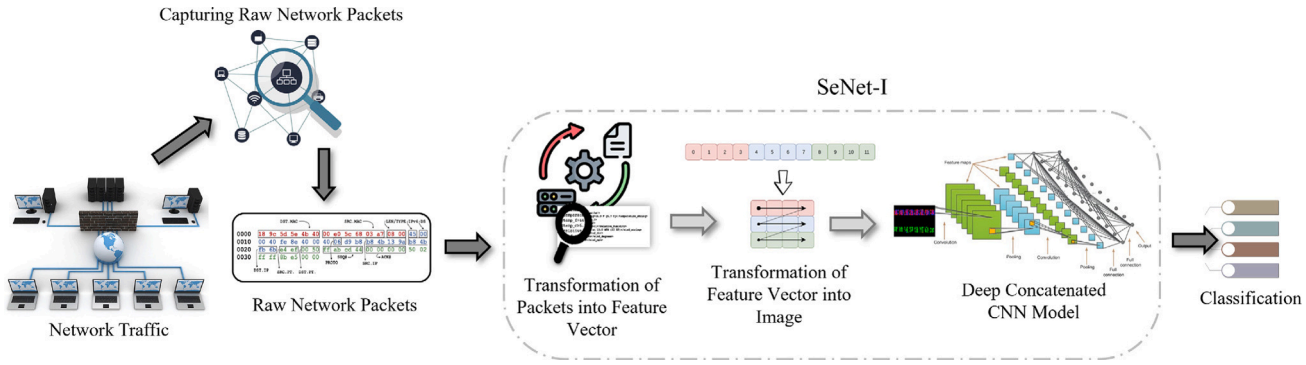


Fig. 3. Workflow representation of proposed approach SeNet-I. The raw packets of network traffic are transformed into a feature vector of 1504 features, which is then transformed into an image and serves as input to the deep concatenated CNN model.

potentially introduce bias into the ML model's understanding of the relationship between the features and attack instances. In addition, we addressed the class imbalance problem by under-sampling the benign data instances before scaling the data using Standard-Scaler. The final feature space for UNSW-NB15 and CIC-IDS2017 datasets was found to be 42 and 79 features, respectively.

For more specific details and numbers regarding our preprocessing steps, please refer to our previous paper (Farrukh et al., 2022).

4.2.2. Packet-based dataset

Packet-based datasets often lack labels, which can pose a challenge when trying to utilize packet-based information. To overcome this issue, we utilized our previous tool (Payload-Byte) (Farukh et al., 2022) that extracts and labels packet capture files of network traffic using metadata from network intrusion detection datasets. This approach standardizes the process of dataset curation and provides a baseline for future researchers to compare and reproduce other packet-based approaches. By doing so, it eliminates engineering and linguistic errors that can arise from using existing datasets.

Payload-Byte utilizes the five tuple features *Source IP, Destination IP, Source Port, Destination Port, and Protocol* to match packets with labeled flow-based data instances. The features that Payload-Byte outputs after labeling are payload content along with three additional features (*Time to Live, Total length and Protocol*) from the packet header. Since payload size varies with each packet, Payload-Byte uses a maximum payload length of 1500 bytes, as done by De Lucia et al. (2021) and Bierbrauer et al. (2023). The payload extracted by the tool is one huge feature, which is then divided into 1500 features based on bytes. The hexadecimal representation of each byte is transformed into an integer with a range of 0 to 255, resulting in one feature. In cases where the number of payload bytes is lesser than 1500, zero padding is used to preserve the standard structure of the feature vector. This ensures that the dimensionality of our feature space is fixed and predetermined. This fixed feature space ensures the feasibility of transforming a one-dimensional array into a image consistently across different samples and scenarios. A pictorial representation of the feature vector outputted by Payload-Byte is illustrated in Fig. 2.

However, labeled packets involve several duplicated values and instances where packets have no payload. Since the number of labeled packets is extensive, we removed duplicates and instances having no payload. Afterward, the data is under-sampled to reduce its size by decreasing benign instances, as done in the preprocessing of the flow-based dataset. The final feature space for both datasets (UNSW-NB15 and CIC-IDS2017) is 1504. For complete details of preprocessing and the workings of Payload-Byte, refer to our work Farrukh et al. (2022).

4.3. Network traffic to images

In this work, our primary objective was to examine packet-based network traffic by transforming raw packet data into images. This was a priority due to the major challenge posed by packet-based network traffic, which is the vast volume of data and the extensive feature space. As the payload of an individual packet varies, it is difficult to handle such huge data in a fixed feature space. Therefore, utilizing an image-based model for such a problem seems to be the feasible solution. However, the basis of image-based models is built upon the spatial correlation between the pixels of an image, whereas there is no major correlation between features of network traffic. But, in terms of network traffic payload data, each neighboring byte has some relation with each other. Therefore, our intuition behind our proposed approach was to exploit this relation along with the capabilities of image-based models. The fundamentals of our proposed approach is built on the success of current image-based models that utilize raw binary bytes for malware detection (Rong et al., 2020; Wang et al., 2022) and authorship attribution (Alrabaee et al., 2019). An additional plus point of image-based models is that they are designed to automatically extract features from image data, making it possible to handle high-dimensional data without feature engineering. Lastly, they can also inherently learn patterns and relationships in the data that may not be visible but could result in improved performance.

To evaluate our hypothesis that converting network traffic into images would enhance performance, we experimented with four different image forms for both packet-based and flow-based network traffic: grayscale (single-channel image), RGB (three-channel image), serialized grayscale image (aggregated single-channel image), and serialized RGB images (aggregated three-channel image). After comparing the performance of each image form, we discovered that serialized RGB images yielded the best results. Therefore, our proposed approach employed serialized RGB images. The workflow for our proposed approach and the NIDS based on SeNet-I is depicted in Fig. 3. Further details on the transformation process can be found in the subsequent headings.

4.3.1. Grayscale images

The grayscale or single-channel image is a two-dimensional representation where each point corresponds to a pixel in the image. In an image with dimensions 24 × 24, there are 576 unique values that are arranged in the two-dimensional array. These unique values correspond to individual pixels in the image. In case of network traffic, the extracted features are represented as a one-dimensional array, which can be transformed into a two-dimensional array to create a grayscale image. There are several ways in which one dimensional data can be converted into two-dimensional data for images. However, in this work we have based our basis on Golubev et al. (2022) in which the authors have utilized linear transformation as it is the most widely adopted approach in NIDS for laying out the pixels within the image. In this

Algorithm 1 SeNet-I

$PacketCount$ = Counter for captured Packets
n_o = Number of packets to be serialized
(h, w) = Selected height & width for SeNet-I Images
$Serialized_{Packets}$ = Deque(maxlen=n_o)
$CurrentPacket \leftarrow$ Most recent captured packet
while *Operational* **do**
  if $PacketCount < n_o$ **then**
    $Serialized_{Packets} \leftarrow Serialized_{Packets}.\ append\ (CurrentPacket)$
    Continue
  else
    $Serialized_{Packets} \leftarrow Serialized_{Packets}.\ append\ (CurrentPacket)$
    $Green_{channel} \leftarrow$ Linear transformation of $Serialized_{Packets}$ into 2D array of (h,w) dimensions
    $Blue_{channel} \leftarrow$ Flip 90° $Green_{channel}$
    $Red_{channel} \leftarrow$ Flip 90° $Blue_{channel}$
    $Serialized_{Images} \leftarrow Red_{channel}$, $Blue_{channel}$, $Green_{channel}$
    Input to DC-CNN: $Serialized_{Images}$ ▷ DC-CNN: Deep Concatenated CNN
    Output: Classification
  end if
end while

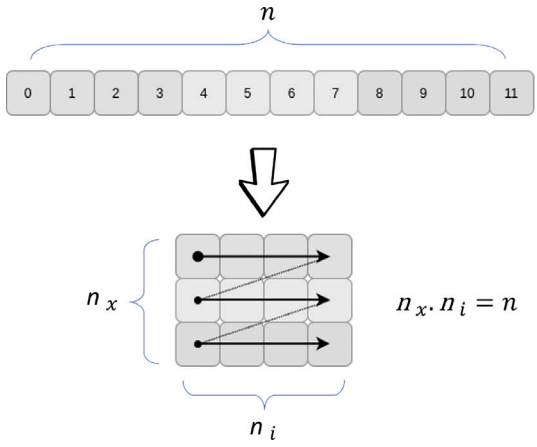


Fig. 4. Transformation of one-dimensional array into two-dimensional array where n is the total length of data. n_x represents number of segments and n_i shows the length of each segment.

linear transformation, the pixel matrix is filled row by row. However, comparative analysis for all other possible ways for transformation will be carried out in our future work.

To transform traffic features into a two-dimensional array, we divide the data into n_x segments with a length of n_i, such that $n_x.n_i = n$, as illustrated in Fig. 4. We then utilize linear transformation to transform the data into the selected image dimensions. In addition to the various methods available for laying out the pixels within an image, there are also several options for choosing the image dimension during the transformation process. However, in this work, the transformation of the data into images is performed using the convention of selecting image dimensions that are in powers of two. Various other image dimensions will be explored in our future work to asses the impact of selection of image dimension on the performance.

For packet-based network traffic with 1504 features, we transformed the data to 2048 by padding zeros, distributing them evenly between the start and end. We then normalized the data to ensure all features were on the same scale and performed linear transformation into a 2-dimensional array and image. We chose 2048 by multiplying the two closest numbers that are powers of two, and selected n_x = 32 and n_i = 64, resulting in a transformed image dimension of 64 × 32. This dimension is the same for both the UNSW-NB15 and CIC-IDS2017 datasets since their packet-based network traffic features

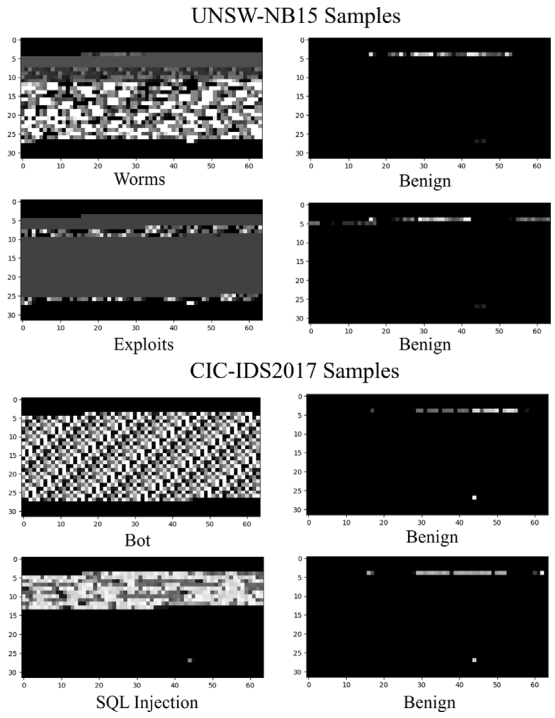


Fig. 5. Illustration of the converted images for the UNSW-NB15 and CIC-IDS2017 datasets. Random samples are selected for illustration.

are similar. For flow-based network traffic, the final feature space after pre-processing was found to be 42 and 79 for the UNSW-NB15 and CIC-IDS2017 datasets, respectively. Using the same methodology, we selected dimensions of 8 × 8 and 8 × 16 for the two datasets. Fig. 5 provides examples of transformed images for both datasets.

4.3.2. RGB images

RGB images, also known as three-channel images, represent colors and shades by utilizing three channels: red, green, and blue. The combination of these color channels allows for the creation of a vast array of different colors and hues. Each pixel in the image is represented by three separate intensity values for the red, green, and blue components, which are stored as three separate 2D arrays. An array of numeric data can be transformed into RGB image following the same procedure as of grayscale image. However, the only difference is that we need to have

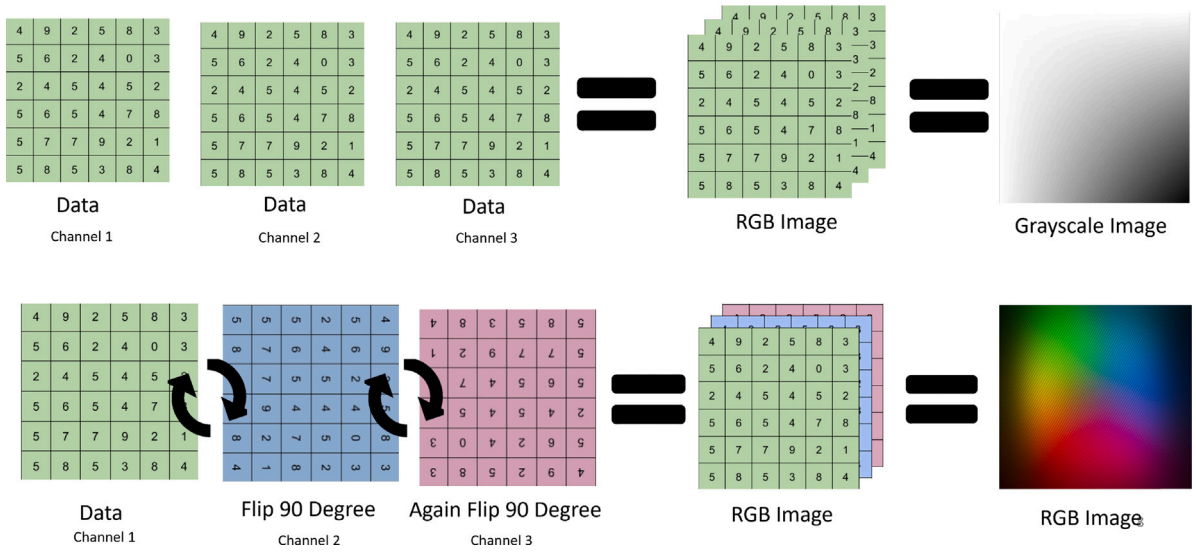


Fig. 6. An illustration for the conversion of network traffic data into three-channel images using the array reversal or flipping technique. The top row of the image demonstrates that, when identical data is supplied to all three channels, the output is equivalent to a grayscale image.

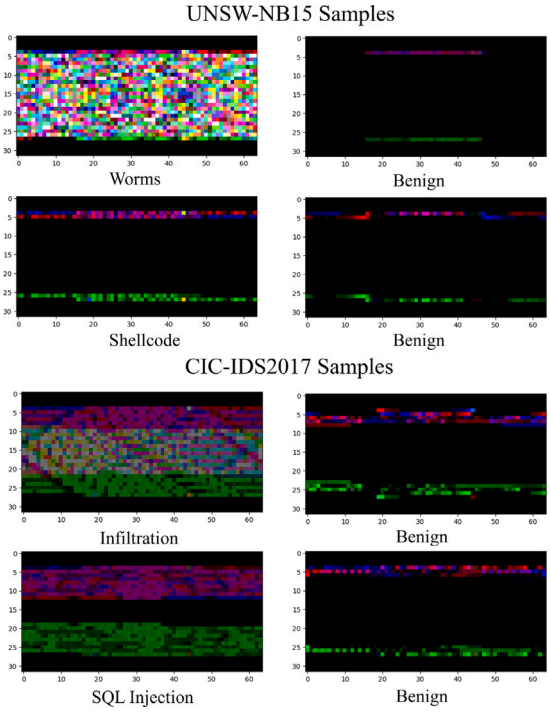


Fig. 7. Illustration of the converted RGB images for the UNSW-NB15 and CIC-IDS2017 datasets. Random samples are selected for illustration.

three different 2D arrays for each channel. Providing the same array to all three color channels results in an image similar to a grayscale image, lacking in any added benefits. To overcome this issue, we implemented a method of reversing the array, also known as flipping the array. We transformed the data into a 2D array, as we did for grayscale, and then flipped the same data by 90 degrees for the second channel and again by 90 degrees for the third channel. The adopted approach is illustrated in Fig. 6, where the top row of the picture shows that providing similar data to all three channels results in an output that is similar to a grayscale image. On the other hand, applying the flipping method results in an RGB image. The hypothesis behind this approach is that combining two additional placements of the feature space and then combining them creates relationships between features that are not visible to the human eye and can enhance the visual representation of data. However, exploration of different ways through which we can incorporate other channels data will be explored in our future work. The dimensions for RGB images are the same as those for grayscale images, with only the addition of the three color channels. Examples of transformed images for both dataset is illustrated in Fig. 7.

4.3.3. Serialized images

Serialized images refer to a type of image representation where multiple samples are combined or aggregated into a single image. This approach might be effective in capturing both spatial and temporal aspects of the image data. Instead of converting a single instance of a sample into an image, the idea behind serialized images is to convert a stream of samples into an image, thereby allowing the image to capture the temporal information contained within the data and potentially uncover anomalous behavior that might not have been visible otherwise. The same methodology was utilized for generating serialized images from network traffic, but instead of transforming a single data sample into a 2D array, we first combined n_o number of samples into a single long array. This long array is then converted into a 2D array using the same process as before. For our proposed approach we selected n_o to be 5 which is selected randomly. However, a detailed study exploring the sensitivity of the proposed approach to the number of samples aggregated will be carried out in future.

When generating serialized images by combining multiple samples, the question of determining the label for the resulting image arises. In our approach, which involves continuous aggregation of new data packets and feeding the generated images into a classifier, the label for each image is based on the latest packet that is combined into it. This real-time process utilizes a First-in, First-out (FIFO) methodology, where new packets replace the oldest ones during image formation. Fig. 8 provides an example of serialized image generation using a parameter $n_o = 3$. Each color represents samples collected at different times. Initially, the samples are combined into a long array, which is then transformed into a 2D array to form the image. The label of the generated image is determined based on the label of the latest sample included. As new samples arrive, the oldest sample is discarded, and the new sample is added to the long array, resulting in a refreshed image. It is important to note that the effectiveness of the FIFO mechanism is not influenced by the number of samples, making it a reliable approach regardless of the sample size. Furthermore, the processing time for generating continuous images, as new network data arrives, remains independent of the sample size. This is achieved by fixed extraction of protocol-independent features from the network data, as mentioned in

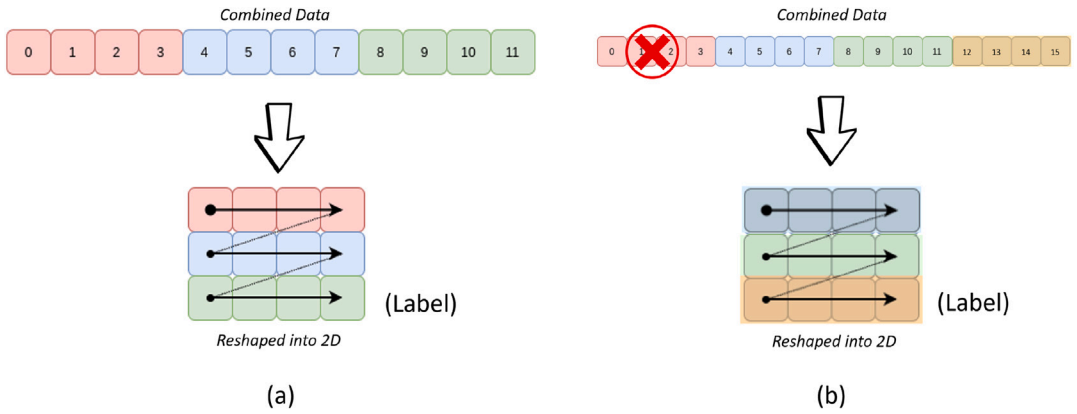


Fig. 8. An illustration of the process of generating serialized images in real-time. Each time a new sample arrives, the oldest sample in the current image is discarded and the new sample is combined with the other, previous samples. The label for the newly generated image is then determined based on the most recent sample.

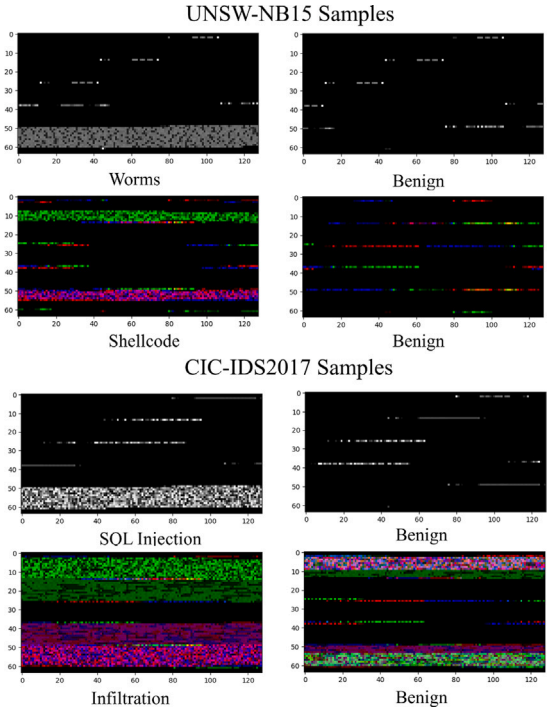


Fig. 9. Illustration of the converted serialized grayscale and RGB images for the UNSW-NB15 and CIC-IDS2017 datasets. The images of attack samples are just one time step ahead of benign images.

Section 4.1. Consequently, the processing time is solely dedicated to extracting predefined features from the latest network data sample and integrating them into the previously generated serialized image. The mechanism continuously updates the aggregated image with real-time data instances, ensuring that the generated images consistently reflect the most up-to-date information about network traffic. The pseudocode for the proposed SeNet-I approach is illustrated Algorithm 1. In terms of complexity, Algorithm 1 has the same complexity as the underlying deep 2D CNN model itself along with the input image dimension (h, w). Moreover, Examples of transformed images for both datasets are shown in Fig. 9.

The dimensions selected for serialized images, both for grayscale and RGB images, adhere to the same principles as previously outlined. As a result, for packet-based network traffic data, the dimension of the image is set to 128×64. In the case of flow-based network traffic, the dimensions are 16×16 for UNSW-NB15 and 32×16 for CIC-IDS2017, respectively. The dimensions chosen for the various forms of images, for both datasets based on flow and packets data, are summarized in Table 2.

Table 2
Selected dimensions for various image types for both packet and Flow-based network traffic.

Type of image	Dataset	Data format	Selected dimension
Grayscale	UNSW-NB15	Packet-based	64×32
		Flow-based	8×8
	CIC-IDS2017	Packet-based	64×32
		Flow-based	8×16
RGB	UNSW-NB15	Packet-based	$64 \times 32 \times 3$
		Flow-based	$8 \times 8 \times 3$
	CIC-IDS2017	Packet-based	$64 \times 32 \times 3$
		Flow-based	$8 \times 16 \times 3$
Serialized grayscale	UNSW-NB15	Packet-based	128×64
		Flow-based	16×16
	CIC-IDS2017	Packet-based	128×64
		Flow-based	32×16
Serialized RGB	UNSW-NB15	Packet-based	$128 \times 64 \times 3$
		Flow-based	$16 \times 16 \times 3$
	CIC-IDS2017	Packet-based	$128 \times 64 \times 3$
		Flow-based	$32 \times 16 \times 3$

4.4. Deep concatenated CNN model architecture

For our proposed approach SeNet-I, we utilized a special CNN model based on input and output concatenation concept. The proposed model combines multiple CNNs in a cascaded or concatenated manner that allows the model to learn multiple levels of abstraction and representations of the input data. The motivation for incorporating such a model can be attributed to the success and inspiration provided by the ResNet model (He et al., 2016) in the field of computer vision. ResNet introduced the concept of residual connections, which enabled the development of very deep networks by mitigating the degradation problem. However, the main goal of integrating input/output concatenation, akin to residual connections, in our proposed model was to capture high-level abstractions of the network traffic data and leverage diverse features at various levels. By incorporating concatenation, we aimed to enhance the model's ability to capture and utilize information from different stages of processing, ultimately improving its performance and robustness. Moreover, by combining multiple CNNs, the model can also learn more complex relationships between the input features, which can improve its ability to classify or perform other tasks on the input data. Additionally, the deep structure of the model allows it to capture both the local and global features of the input data, making it more effective in recognizing patterns and making predictions.

Table 3
Model architecture for deep concatenated convolutional neural network.

	Layer type
Input	InputLayer (height, width, channel)
Block#1	BatchNormalization() Activation('Relu') Conv2D(filters = 16, kernel_size = (3,3), padding = 'same') BatchNormalization() Activation('Relu') Conv2D(filters = 16, kernel_size = (3,3), padding = 'same')
Block#2	BatchNormalization() Activation('Relu') Conv2D(filters = 32, kernel_size = (3,3), padding = 'same') BatchNormalization() Activation('Relu') Conv2D(filters = 32, kernel_size = (3,3), padding = 'same')
Block#3	BatchNormalization() Activation('Relu') Conv2D(filters = 64, kernel_size = (3,3), padding = 'same') BatchNormalization() Activation('Relu') Conv2D(filters = 64, kernel_size = (3,3), padding = 'same',strides = (2,2))
Block#4	Flatten() Dense (units = 128, activation = Activation('Relu')) Dropout (rate = 0.2) Dense (units = 64, activation = Activation('Relu')) Dropout (rate = 0.2)
Output	Dense (units = Number of Attack Classes, activation = 'Softmax')

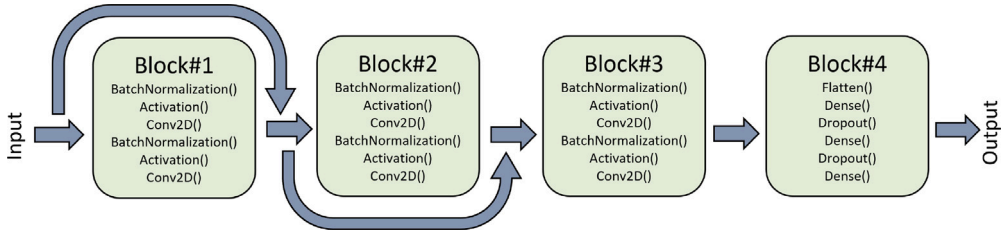


Fig. 10. An illustration of proposed deep concatenated CNN model. The model comprises of four blocks, incorporating the concept of input and output concatenation.

The adopted model architecture adheres to the baselines of the well-known 2D-CNN architectures. Following common practice, the number of filters increases progressively in our proposed model as we traverse the layers, enabling the network to capture more complex and abstract features. Table 3 provides a clear depiction of the layer-wise progression within the model. Furthermore, a 3×3 kernel size is opted for the proposed model as it is commonly preferred and balances effectively, capturing local patterns and maintaining computational efficiency. The same padding ensures that the spatial dimensions of the feature maps remain unchanged post-convolution, preserving crucial spatial information across layers. These design choices were driven by maximizing the model's capacity to learn intricate features while upholding computational efficiency and spatial information preservation.

Additionally, the model incorporates a unique structure where the output of block 1 is combined with the input data, forming the input for block 2. This process is reiterated for block 3, where the output of block 2 is concatenated with the input of block 2. However, the same process is not repeated for block 4 as block 4 is based on dense layers rather than convolutional layers. Since the dense layers operate on flattened feature representations, concatenating the images would not provide significant benefits. This choice was validated by comparing the performance of the proposed model with an alternative model that included concatenation in every block, confirming that omitting concatenation for block 4 yielded superior results. A visual representation of the proposed model architecture is presented in Fig. 10, while comprehensive details can be found in Table 3. For this study, a sparse categorical crossentropy function was employed as the loss function, and Adam was selected as the optimizer.

4.4.1. Model tuning and training

The model training process for our proposed model involved several steps and evaluations. It is noteworthy that our experimentation covered four different types of images and considered two distinct formats of network data, namely flow and packet, encompassing two different datasets. As a result, a total of 16 different models were trained. However, for the sake of clarity and conciseness, we will focus on showcasing the training and tuning of our main proposed model, specifically utilizing serialized RGB images on the CIC-IDS2017 dataset. It is important to highlight that the same procedures and steps were followed for training and tuning all other models in the evaluation.

Before training the model, we conducted a comprehensive hyperparameter tuning process to optimize performance. Our focus was primarily on tuning the Learning Rate and Batch Size parameters, as they play crucial roles in determining convergence and training efficiency. In addition, we cross-validated the model's structure, as discussed in Section 4.4. By comparing the performance of our proposed model with an alternative model that included concatenation in every block, we confirmed that omitting concatenation for the last block yielded superior results.

Fig. 11 presents the hyperparameter tuning curves, demonstrating the model's performance under various configurations of the Learning Rate and Batch Size. For the batch size, we experimented with values of 32, 64, and 128, while exploring learning rates of 0.1, 0.01, and 0.001. It is worth mentioning that the learning rate curves for 0.01 and 0.1, with batch sizes of 32 and 128, respectively, had loss values ranging from 1.5 to 2. To enhance clarity, these curves were omitted from the plot. After thorough hyperparameter tuning, we determined

that the optimal parameters were a batch size of 64 and a learning rate of 0.001. Notably, these parameters were also found to be optimal for the UNSW-NB15 dataset.

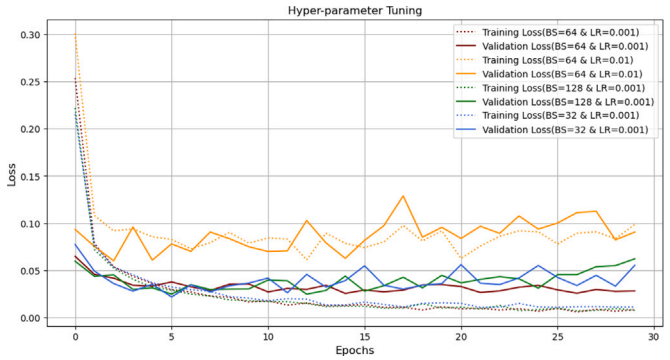


Fig. 11. Hyper-parameter tuning graph for Serialized RGB images on CIC-IDS2017 flow data. The dotted line represents the training loss with specified parameters, while the solid line represents the validation loss. In the graph, 'BS' refers to batch size and 'LR' refers to learning rate.

Once the hyperparameters were finalized, we proceeded to train the model using these parameters. Early stopping was incorporated into the training process, employing a patience value of 5 to halt training when the training loss no longer decreased. The epochs for both packet-based and flow-based datasets were similar, as illustrated in Fig. 12. This figure highlights the training curves for the CIC-IDS2017 dataset, showcasing the convergence and performance of our model during the training process.

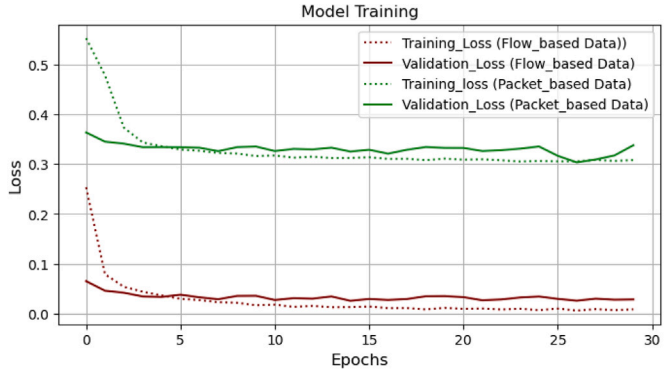


Fig. 12. Training curve for our proposed model on CIC-IDS2017 dataset. Green line represents the training and validation loss for packet based data while maroon line represents the training and validation loss for flow based data.

5. Results and analysis

In this section, a comprehensive analysis of the computed results for the proposed approach is presented. Firstly, we have compared four distinct image forms to showcase that our proposed serialized three-channel images perform far better than the conventional methods. This comparative analysis is computed under the same experimentation setting. Afterward, to demonstrate the efficacy of the proposed approach, we compare it with several widely adopted baseline models. Finally, a comparative analysis between various NIDS techniques based on textual, numeric, and image data was presented and compared with the proposed approach to illustrate how image-based approaches can be incorporated into the security domain for enhanced performance. Moreover, we validated our comparison on both packet-based and flow-based formats to demonstrate its applicability.

Table 4
Results for different transformed images of the CIC-IDS2017 dataset.

Dataset format		Grayscale images	RGB images	Serialized images	SeNet-I
Packet-based network traffic	Accuracy	71%	71%	78%	**79%**
	F1-score	72%	69%	82%	**83%**
	Precision	76%	71%	85%	**86%**
	Recall	71%	68%	82%	**82%**
Flow-based network traffic	Accuracy	99%	99%	99.6%	**99.9%**
	F1-score	86%	87%	91%	**96%**
	Precision	88%	88%	93%	**97%**
	Recall	86%	87%	90%	**95%**

Table 5
Results for different transformed images of UNSW-NB15 dataset.

Dataset format		Grayscale images	RGB images	Serialized images	SeNet-I
Packet-based network traffic	Accuracy	72%	85%	89%	**95%**
	F1-score	58%	60%	75%	**81%**
	Precision	61%	62%	76%	**83%**
	Recall	58%	59%	73%	**80%**
Flow-based network traffic	Accuracy	84%	84%	**97%**	**97%**
	F1-score	57%	58%	89%	**91%**
	Precision	69%	64%	**91%**	**91%**
	Recall	57%	57%	89%	**91%**

It is noteworthy to mention that all the results presented in this section are based on multi-class classification, with 15 separate categories for CIC-IDS2017 and ten distinct categories for UNSW-NB15. Furthermore, we utilized macro average for all performance metrics as it provides a better picture of results where there is class imbalance.

5.1. Different forms of images

In this subsection, the results of our experiments utilizing the proposed deep concatenated CNN model on the both packet and flow based formats, are presented. We utilize the same experiment setting and same models for each image form. For generation of images we utilized our pre-processed data as described earlier. The results obtained for the transformed network traffic in the UNSW-NB15 and CIC-IDS2017 datasets are depicted in Fig. 13. The figure demonstrates that the proposed approach for serialized RGB images outperforms all other types of transformed images. Furthermore, it is evident that the packet-based network traffic yields better results for the UNSW-NB15 dataset, while the opposite is true for the CIC-IDS2017 dataset. A comprehensive analysis of the results for both the UNSW-NB15 and CIC-IDS2017 datasets can be found in Tables 4 and 5, respectively.

5.2. Comparison with baseline models

To assess the effectiveness of our proposed *SeNet-I* approach, we conducted experiments comparing its performance to several baseline models using both packet-based and flow-based network traffic data formats for both the CIC-IDS2017 and UNSW-NB15 datasets. The models used for evaluation include Random Forest (RF), Logistic Regression, AdaboostClassifier, Multilayer Perceptron (MLP), a DNN consisting of three layers, and a combination of a 1D-CNN and Long Short-Term Memory (LSTM) with five layers. We have selected these models as they are widely adopted and proven effective in the literature. The performance of baseline models is shown in Fig. 14, where F1-Score is reported for each model. However, we have not tuned the hyper-parameter of these baseline models and have utilized default hyper-parameters. The results reveal that the proposed approach *SeNet-I* outperforms all other baseline models. Table 6 provides detailed results for each baseline model. Based on the results obtained, it is evident that the random forest model is performing better than other

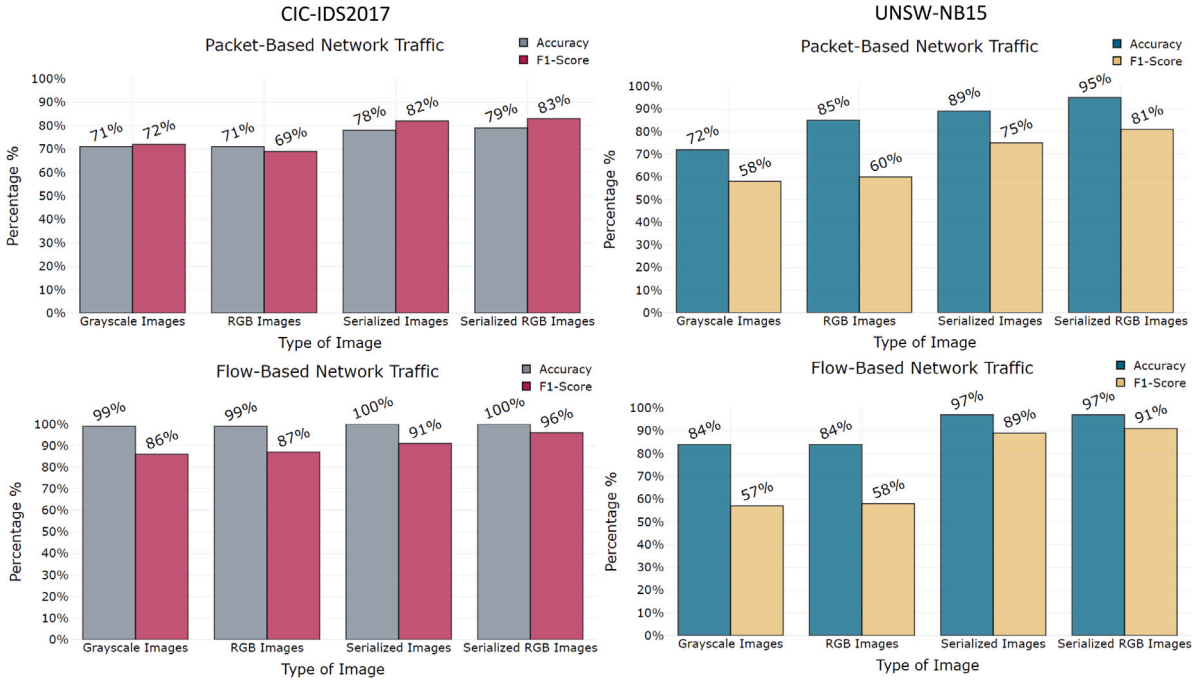


Fig. 13. The macro accuracy and F1-score for both packet-based and flow-based network traffic of CIC-IDS2017 and UNSW-NB15 datasets, after transforming the data into grayscale, RGB, serialized grayscale, and serialized RGB images.

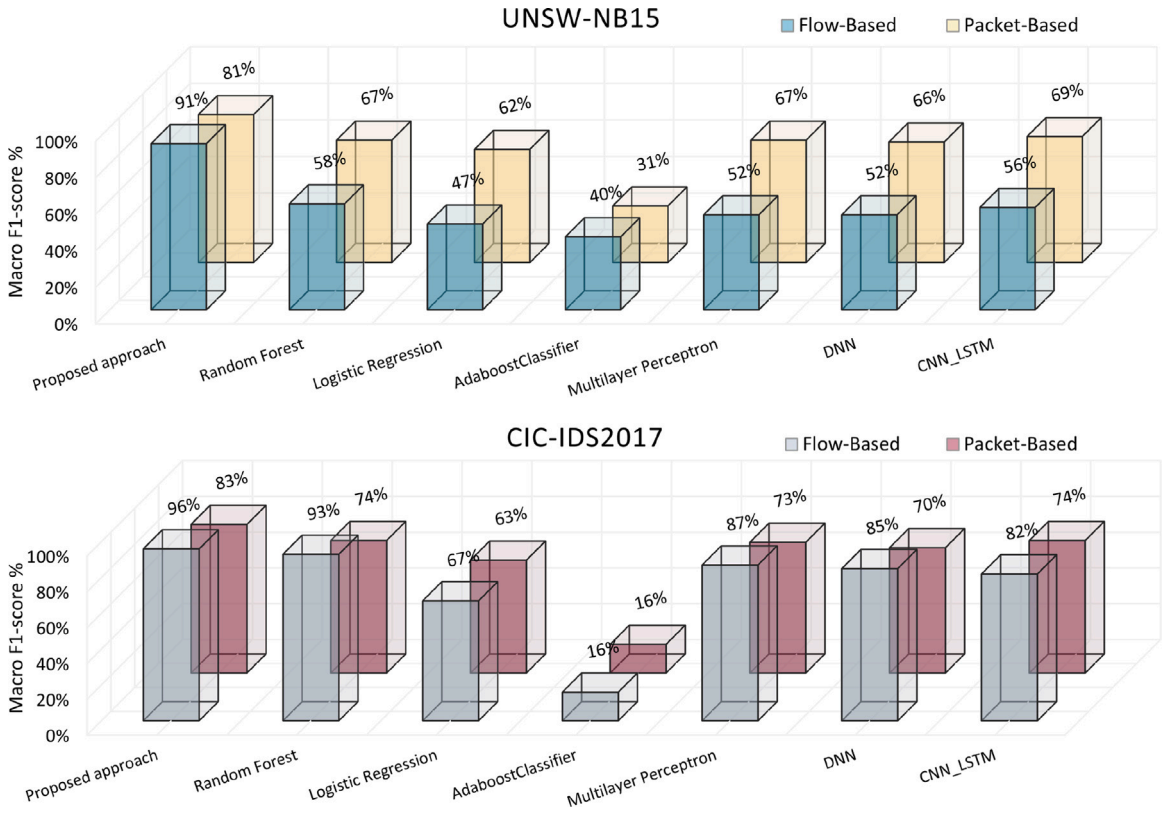


Fig. 14. Comparison between the proposed approach and baseline models using macro-average F1-score as the evaluation metric. Since all of these models are for multi-class classification, accuracy is not a suitable metric. The baseline models have been trained using network traffic data in numeric form whereas proposed approach utilizes serialized RGB images.

Table 6
Results for baseline models.

Dataset format		Random forest	Logistic regression	Adaboost-classifier	Multilayer perceptron	DNN	CNN LSTM
(CIC-IDS2017)	Precision	**96%**	75%	15%	92%	89%	82%
Flow-based	Recall	**90%**	66%	20%	87%	85%	84%
network traffic	F1-score	**93%**	67%	16%	87%	85%	82%
(CIC-IDS2017)	Precision	**78%**	69%	15%	82%	84%	77%
Packet-based	Recall	74%	65%	19%	73%	72%	**78%**
network traffic	F1-score	**74%**	63%	16%	73%	70%	74%
(UNSW-NB15)	Precision	**62%**	48%	37%	54%	56%	61%
Flow-based	Recall	**56%**	42%	54%	51%	51%	56%
network traffic	F1-score	**58%**	42%	40%	52%	52%	56%
(UNSW-NB15)	Precision	66%	57%	34%	72%	74%	**75%**
Packet-based	Recall	67%	55%	34%	68%	66%	**69%**
network traffic	F1-score	67%	55%	31%	67%	66%	**69%**

Table 7
Comparison of packet-based approaches for CIC-IDS2017.

Methods	Performance metric			
	Accuracy	Precision	Recall	F1-score
EsPADA (Vidal et al., 2020)	64.4%	86.7%	71.6%	78.6%
PL-RNN (Liu et al., 2019)	68%	83.3%	56.3%	69.3%
Packet2Vec (Goodman et al., 2020)	66.2%	84%	72.8%	62.7%
HAST-II (Wang et al., 2017)	64.2%	**88.5%**	68.9%	74.9%
AEIDS (Pratomo et al., 2018)	77.3%	63.2%	58.4%	61.8%
Proposed approach	**79%**	86%	**82%**	**83%**

Table 8
Comparison of Packet-based approaches for UNSW-NB15.

Methods	Performance metric			
	Accuracy	Precision	Recall	F1-score
EsPADA (Vidal et al., 2020)	85.3%	75.2%	85.6%	68.1%
PL-RNN (Liu et al., 2019)	76.3%	72%	**88.2%**	74.7%
Packet2Vec (Goodman et al., 2020)	72.5%	72%	87.9%	60.9%
HAST-II (Wang et al., 2017)	76.8%	79.8%	77%	77.5%
AEIDS (Pratomo et al., 2018)	73.3%	58.9%	80.7%	69.3%
Proposed approach	**95%**	**83%**	80%	**81%**

baseline models. Nonetheless, it is worth noting that for packet-based network traffic, where the feature space dimension is considerably large, the CNN-LSTM model is almost on par with the RF model in terms of performance. For detailed result of our proposed approach, refer to Tables 4 and 5.

5.3. Comparison with state of the art

After conducting experiments on baseline models and various possible image form approaches, we found that our proposed approach outperformed all of them. However, we wanted to compare our approach with other available approaches in the literature that used numeric, textual, and image input forms. Unfortunately, we encountered some obstacles while trying to make these comparisons.

Firstly, some of the previous works used different datasets than our own, such as CIC-IDS2017 or UNSW-NB15, making it impossible to compare them directly. Secondly, some of the authors did not provide results for multi-class classification, and some only used accuracy as a performance metric, which is not ideal for multi-class classification problems. Thirdly, some of the previous works combined different classes into a single class, making it difficult to make a fair comparison.

Despite these challenges, we endeavored to find state-of-the-art approaches that performed experiments on the same dataset as ours, and reported key performance metrics such as precision, recall, accuracy, and F-1 Score. We searched through review articles for cases where authors have performed experimentation of the proposed state of the art on similar experiment setting as of ours. Our ultimate goal was to provide a fair comparison between our proposed approach and the state-of-the-art, and to include only those results that were obtained under similar experimental settings.

Tables 7 and 8 present a comprehensive comparison between our proposed approach (SeNet-I) and other state of the art approaches utilizing packet-based network traffic. We included only those approaches that make use of the same datasets and have shown performance metrics on multi-class classification. However, most of the packet-based network traffic approaches either utilize proprietary data or datasets that are not very well known. Therefore, finding an appropriate match for comparison was difficult task. To mitigate this issue, we utilized the work of Hassan et al. (2022) in which the authors have computed most of the packet-based approaches on well known datasets. The results shown in Tables 7 and 8 are based on that work. It can be depicted from the results that our proposed approach is outperforming other approaches in almost every performance metric. One important point to emphasize is that the macro average has been presented for each performance metric instead of the weighted average which is crucial if we are evaluating multi-class classification having class imbalance.

Tables 9 and 10 compare the proposed approach and state-of-the-art approaches when applied to flow-based network traffic of CIC-IDS2017 and UNSW-NB15, respectively. Our proposed approach outperforms every other method in three out of four performance metrics for CIC-IDS2017, and for UNSW-NB15, we have better performance in terms of two performance metrics. It is important to note that F1-score is considered the most relevant performance evaluation metric for multi-class classification, and our proposed approach outperforms any other method in this metric. Therefore, we have based our performance assessment on the F1 score, which provides a more holistic representation of the model's effectiveness. By considering both precision and recall, the F1 score offers a balanced measure of the model's performance. Additionally, it is possible to adjust the model to prioritize one metric over the other, depending on the specific requirements and objectives of the application. This flexibility allows for further optimization and customization of the model's performance characteristics.

6. Discussion and future work

One of the key questions that this research intended to investigate was how to incorporate images into NIDS and utilize the true power

Table 9
Comparison of Flow-based approaches for CIC-IDS2017.

Methods	Performance metric			
	Accuracy	Precision	Recall	F1-score
Meta-learning (Yulianto et al., 2019)	–	81.8%	100%	90%
MLP (Yulianto et al., 2019)	–	77%	83%	81%
Bayesian network (Abdulhammed et al., 2019)	95%	–	–	**96%**
Hybrid AE-CNN (Mohammadpour et al., 2022)	95.24%	90.76%	79.97%	85.02%
1D-CNN (Mohammadpour et al., 2022)	97.66%	88.38%	**99.2%**	93.94%
Proposed approach	**99.9%**	**97%**	95%	**96%**

Table 10
Comparison of flow-based approaches for UNSW-NB15.

Methods	Performance metric			
	Accuracy	Precision	Recall	F1-score
MCNN-DFS (Al-Turaiki and Altwaijry, 2021)	80.51%	81%	81	81
CNN-BiLSTM (Sinha and Manollas, 2020)	76.8%	81.4%	78.5%	79.95%
CNN–WDLSTM (Hassan et al., 2020)	**98.43%**	68.6%	54.8%	57.2%
VLSTM (Zhou et al., 2020)	89.5%	86.2%	**97.8%**	90.7%
Scale-Hybrid-IDS-AlertNet (Vinayakumar et al., 2019)	65.1%	59.7	65.1%	58.8%
Proposed approach	97%	**91%**	91%	**91%**

of CNN models as it has been adopted in several other domains as mentioned in Section 4.2. Since CNN models can extract features more effectively than conventional approaches due to their unique architecture and operations, it was one of the major attractions for implementing such an approach. CNN models can convolve input images with a set of learnable filters to extract features at various levels of abstraction, which may not be visible to the human eye. Additionally, the primary objective of this research was to address the issue of developing a NIDS based on raw network packets, as current approaches face challenges in determining a suitable feature space. Through the proposed approach, the complete packet information can be utilized by transforming network packets into images without truncating any information. Experimentation revealed that transforming network packets into images can capture more complex and nuanced features than tabular data. The model can learn patterns and features at different scales and orientations with images, leading to better performance, as demonstrated in this work.

Through our research, we discovered a significant limitation of using image transformation techniques to represent network traffic. We found that such approaches only consider the spatial information of the network traffic, ignoring the crucial temporal information contained in the sequential packets. To overcome this limitation, our proposed approach incorporates the serialization of images, enabling our model to learn the sequential nature of the traffic and capture critical timing details. By aggregating packets into visual representations, our model becomes adept at identifying patterns, dependencies, and anomalies present in the traffic. The integration of low-level features extracted from individual packets and the high-level abstractions learned by the deep convolutional neural network empowers our model to effectively analyze and classify network traffic based on crucial timing information. Consequently, there is no need to introduce additional models, such as Recurrent Neural Networks (RNNs), to capture temporal information. By combining spatial and temporal information within our proposed approach, we can fully leverage the capabilities of Convolutional Neural Network (CNN) models. This facilitates the extraction of varying levels of abstraction from network traffic data, leading to improved accuracy and performance in detecting intrusions.

While several works have achieved impressive results by utilizing a 1D-CNN along with RNN for packet-based network intrusion detection systems (NIDS), there is a question that arises as to why one should prefer to use a 2D-CNN instead. In this context, our proposed approach, based on a 2D-CNN, has been shown to outperform existing 1D-CNN and RNN approaches in our experiments. One reason for its superior performance may be its ability to take advantage of the spatial relationships between different packets in the network traffic payload. By representing the network traffic as an image, a 2D-CNN can learn to identify patterns and anomalies that are spatially distributed across the payload of several packets, which a 1D-CNN may struggle to capture due to its limited perspective on the one-dimensional sequence of packets. Additionally, a 2D-CNN can recognize patterns that are both spatially and temporally dependent, such as those that occur repeatedly over a period of time and may be indicative of an attack or intrusion. This ability to capture complex patterns gives the 2D-CNN an advantage over the 1D-CNN in network traffic analysis which is demonstrated through our proposed approach.

Another key advantage of the proposed approach is that it may be less vulnerable to adversarial attacks. This is due to the utilization of the concept of array reversal for generating RGB images, which randomizes the packet information. If an attacker attempts to evade the NIDS through an adversarial example (Alhajjar et al., 2021), it would be challenging to generate such a sample while maintaining the functionality of the attack. This is because any change made to the image would have a different effect due to the array reversal step incorporated in the proposed approach. Additionally, the CNN model captures the spatial relationships and local patterns within an image, allowing each pixel to have a relationship with its neighboring pixels. This neighboring relationship is extended three times as array reversal is incorporated in the proposed approach. As a result, each byte of a packet some sort of develop a spatial relationship with other payload bytes, and this relationship is further extended after convolution and pooling effects. Therefore, we posit that our proposed approach is less prone to adversarial attacks than other proposed approaches.

One crucial consideration that arises with such an approach is whether the inference speed of our proposed model will meet real-time requirements. It is important to acknowledge that the inference speed of our model, based on deep concatenated CNNs, may not match the rate at which individual packets are captured in real-time. CNNs involve computationally intensive operations, and their inference speed can be influenced by factors such as model size, depth, available computational resources, and input data dimensions. Therefore, achieving real-time performance comparable to capturing a single packet can be challenging. However, it is worth noting that the operational speed of practically implemented NIDS is still relatively low compared to network speeds (Lai et al., 2004). Nonetheless, It is also possible to increase the inference speed of our model by leveraging high-end computational power and optimizing the implementation. Utilizing powerful hardware configurations like GPUs or dedicated inference accelerators can significantly enhance the model's performance and reduce the inference time. Furthermore, algorithmic optimizations and parallel processing techniques can be applied to further improve the overall inference speed.

Lastly, the proposed approach was extensively evaluated and compared with other state-of-the-art approaches. Both flow-based and packet-based network traffic were considered during the implementation. However, one minor limitation in the proposed approach is that it only considers packets' payload data while disregarding the packets' header information. Therefore, the proposed method is vulnerable to

header-based attacks such as scanning or probing attacks. However, for future research, we are devising a feature space that includes packets' payload information and packet header information that can be utilized for NIDS without creating any spurious correlation with the extracted feature. Additionally, we are also conducting experiments to evaluate the sensitivity of different methods for laying pixels in an image, as well as the selection of image size and approaches for generating data for the other two channels of images. These experiments will be incorporated into our next work to further improve the effectiveness and efficiency of our approach. By considering various options and fine-tuning the image-based approach, we hope to achieve even better results in network traffic analysis and intrusion detection.

7. Conclusion

This work presented SeNet-I, a novel approach for network intrusion detection that transforms network traffic into serialized three-channel images and uses a deep concatenated CNN model. This proposed approach is capable of capturing both spatial and temporal aspects of network traffic without the need for additional components. By utilizing these spatial and temporal features, we can develop a more abstract and high-level representation of network traffic images that may not be apparent to the naked human eye. Additionally, we provide a detailed literature review categorizing NIDS approaches based on their input forms: numeric, text, and image. We explain the process of transforming network traffic into serialized three-channel images and discuss the benefits and drawbacks of this approach.

Furthermore, to evaluate the proposed method, we use two widely utilized NIDS datasets: UNSW-NB15 and CIC-IDS2017, and perform multi-class classification without altering the number of available classes. We compare SeNet-I with other commonly used ML models and state-of-the-art approaches on packet-based and flow-based network traffic formats. Additionally, we transform packet-based and flow-based network traffic into three additional image forms, grayscale, RGB, and serialized grayscale, for a more comprehensive comparison. Our extensive analysis demonstrates that SeNet-I outperforms all baseline models and available state-of-the-art approaches regarding the F1-score, a reliable metric for multi-class classification. Finally, we discuss the advantages and disadvantages of the proposed approach.

CRediT authorship contribution statement

Yasir Ali Farrukh: Methodology, Data curation, Formal analysis, Experimentation, Writing – original draft. **Syed Wali:** Writing – original draft, Visualization, Investigation, Formal analysis. **Irfan Khan:** Writing – review & editing, Supervision, Resources. **Nathaniel D. Bastian:** Conceptualization, Methodology, Writing – review & editing, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the U.S. Military Academy (USMA) under Cooperative Agreement No. W911NF-22-2-0081 and the U.S. Army Combat Capabilities Development Command (DEVCOM) C5ISR Center under Support Agreement No. USMA21056. The views and conclusions expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Military Academy, U.S. Army, U.S. Department of Defense, or U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. The U.S. Government reserves a royalty-free, nonexclusive and irrevocable right to reproduce, publish, or otherwise use this data for Federal purposes, and to authorize others to do so in accordance with 2 CFR 200.315(b).

References

Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., Abuzneid, A., 2019. Features dimensionality reduction approaches for machine learning based network intrusion detection. Electronics 8 (3), 322.

Al-Turaiki, I., Altwaijry, N., 2021. A convolutional neural network for improved anomaly-based network intrusion detection. Big Data 9 (3), 233–252. http://dx.doi.org/10.1089/big.2020.0263, PMID: 34138657.

Alhajjar, E., Maxwell, P., Bastian, N., 2021. Adversarial machine learning in network intrusion detection systems. Expert Syst. Appl. 186, 115782.

Alrabaee, S., Karbab, E.B., Wang, L., Debbabi, M., 2019. Bineye: Towards efficient binary authorship characterization using deep learning. In: Computer Security–ESORICS 2019: 24th European Symposium on Research in Computer Security, Luxembourg, September 23–27, 2019, Proceedings, Part II 24. Springer, pp. 47–67.

Andresini, G., Appice, A., Di Mauro, N., Loglisci, C., Malerba, D., 2020. Multi-channel deep feature learning for intrusion detection. IEEE Access 8, 53346–53359.

Ariu, D., Giacinto, G., 2010. HMMPayl: an application of HMM to the analysis of the HTTP payload. In: Proceedings of the First Workshop on Applications of Pattern Analysis. PMLR, pp. 81–87.

Bierbrauer, D.A., De Lucia, M.J., Reddy, K., Maxwell, P., Bastian, N.D., 2023. Transfer learning for raw network traffic detection. Expert Syst. Appl. 211, 118641. http://dx.doi.org/10.1016/j.eswa.2022.118641.

Cao, B., Li, C., Song, Y., Qin, Y., Chen, C., 2022. Network intrusion detection model based on CNN and GRU. Appl. Sci. 12 (9), 4184.

Chalé, M., Bastian, N.D., 2022. Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems. Expert Syst. Appl. 207, 117936.

De Lucia, M.J., Maxwell, P.E., Bastian, N.D., Swami, A., Jalaian, B., Leslie, N., 2021. Machine learning raw network traffic detection. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III, Vol. 11746. SPIE, pp. 185–194.

Dimitrios Tsokos Supervisor, F., Georgios, P., 2021. Network Dataset Generation and Implementation of a Network Intrusion Detection System using Neural Networks.

Farrukh, Y., Khan, I., Wali, S., Bierbrauer, D.A., Pavlik, J., Bastian, N.D., 2022. Payload-Byte: A Tool for Extracting and Labeling Packet Capture Files of Modern Network Intrusion Detection Datasets. IEEE/ACM.

Farukh, Y., Khan, I., Wali, S., Bierbrauer, D.A., Pavlik, J., Bastian, N.D., 2022. Payload-byte. https://github.com/Yasir-ali-farrukh/Payload-Byte.

Golubev, S., Novikova, E., Fedorchenko, E., 2022. Image-based approach to intrusion detection in cyber-physical objects. Information 13 (12), 553.

Goodman, E.L., Zimmerman, C., Hudson, C., 2020. Packet2vec: Utilizing word2vec for feature extraction in packet data. arXiv preprint arXiv:2004.14477.

Halisdemir, M.E., Karacan, H., Pihelgas, M., Lepik, T., Cho, S., 2022. Data quality problem in AI-based network intrusion detection systems studies and a solution proposal. In: 2022 14th International Conference on Cyber Conflict: Keep Moving! Vol. 700. CyCon, IEEE, pp. 367–383.

Han, X., Yin, R., Lu, Z., Jiang, B., Liu, Y., Liu, S., Wang, C., Li, N., 2020. Stidm: A spatial and temporal aware intrusion detection model. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications. TrustCom, IEEE, pp. 370–377.

Hassan, M.M., Gumaei, A., Alsanad, A., Alrubaian, M., Fortino, G., 2020. A hybrid deep learning model for efficient intrusion detection in big data environment. Inform. Sci. 513, 386–396.

Hassan, M., Haque, M.E., Tozal, M.E., Raghavan, V., Agrawal, R., 2022. Intrusion detection using payload embeddings. IEEE Access 10, 4015–4030. http://dx.doi.org/10.1109/ACCESS.2021.3139835.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Iasiello, E., 2021. What is the role of cyber operations in information warfare? J. Strateg. Secur. 14 (4), 72–86.

Jallad, K.A., 2022. FastPacket: Towards pre-trained packets embedding based on FastText for next-generation NIDS. arXiv preprint arXiv:2209.14727.

Jiang, K., Wang, W., Wang, A., Wu, H., 2020. Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access 8, 32464–32476.

Khraisat, A., Alazab, A., 2021. A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity 4, 1–27.

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., 2019. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2 (1), 1–22.

Kim, A., Park, M., Lee, D.H., 2020. AI-IDS: Application of deep learning to real-time web intrusion detection. IEEE Access 8, 70245–70261.

Krupski, J., Graniszewski, W., Iwanowski, M., 2021. Data transformation schemes for cnn-based network traffic analysis: A survey. Electronics 10 (16), 2042.

Lai, H., Cai, S., Huang, H., Xie, J., Li, H., 2004. A parallel intrusion detection system for high-speed networks. In: Applied Cryptography and Network Security: Second International Conference, ACNS 2004, Yellow Mountain, China, June 8-11, 2004. Proceedings 2. Springer, pp. 439–451.

Liu, H., Lang, B., Liu, M., Yan, H., 2019. CNN and RNN based payload classification methods for attack detection. Knowl.-Based Syst. 163, 332–341.

Malaiya, R.K., Kwon, D., Kim, J., Suh, S.C., Kim, H., Kim, I., 2018. An empirical evaluation of deep learning for network anomaly detection. In: 2018 International Conference on Computing, Networking and Communications. ICNC, IEEE, pp. 893–898.

Mohammadpour, L., Ling, T.C., Liew, C.S., Aryanfar, A., 2022. A survey of CNN-based network intrusion detection. Appl. Sci. 12 (16), 8162.

Moustafa, N., Slay, J., 2015. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings. Institute of Electrical and Electronics Engineers Inc., http://dx.doi.org/10.1109/MILCIS.2015.7348942.

Munirathinam, S., 2020. Industry 4.0: Industrial internet of things (IIOT). In: Advances in Computers, Vol. 117, No. 1. Elsevier, pp. 129–164.

Pratomo, B.A., Burnap, P., Theodorakopoulos, G., 2018. Unsupervised approach for detecting low rate attacks on network traffic with autoencoder. In: 2018 International Conference on Cyber Security and Protection of Digital Services. Cyber Security, IEEE, pp. 1–8.

Rong, C., Gou, G., Cui, M., Xiong, G., Li, Z., Guo, L., 2020. TransNet: Unseen malware variants detection using deep transfer learning. In: Security and Privacy in Communication Networks: 16th EAI International Conference, SecureComm 2020, Washington, DC, USA, October 21-23, 2020, Proceedings, Part II 16. Springer, pp. 84–101.

Saleh, I., Ji, H., 2020. Network traffic images: A deep learning approach to the challenge of internet traffic classification. In: 2020 10th Annual Computing and Communication Workshop and Conference. CCWC, IEEE, pp. 0329–0334.

Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. http://dx.doi.org/10.5220/0006639801080116.

Sharma, A., Vans, E., Shigemizu, D., Boroevich, K.A., Tsunoda, T., 2019. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. Sci. Rep. 9 (1), 11399.

Shen, Y., Zheng, K., Wu, C., Zhang, M., Niu, X., Yang, Y., 2018. An ensemble method based on selection using bat algorithm for intrusion detection. Comput. J. 61 (4), 526–538.

Sinha, J., Manollas, M., 2020. Efficient deep CNN-BiLSTM model for network intrusion detection. In: Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition. pp. 223–231.

Tama, B.A., Lim, S., 2021. Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. Comp. Sci. Rev. 39, 100357.

Tao, F., Akhtar, M.S., Jiayuan, Z., 2021. The future of artificial intelligence in cybersecurity: A comprehensive survey. EAI Endorsed Trans. Creative Technol. 8 (28), e3.

Tas, O., Yahyaoui, A., 2022. Machine learning based intrusion detection system using grey wolf optimization for feature selection.

Vidal, J.M., Monge, M.A.S., Monterrubio, S.M.M., 2020. Espada: Enhanced payload analyzer for malware detection robust against adversarial threats. Future Gener. Comput. Syst. 104, 159–173.

Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A., Venkatraman, S., 2019. Deep learning approach for intelligent intrusion detection system. IEEE Access 7, 41525–41550.

Wang, F., Chai, G., Li, Q., Wang, C., 2022. An efficient deep unsupervised domain adaptation for unknown malware detection. Symmetry 14 (2), 296.

Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., Zhu, M., 2017. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access 6, 1792–1806.

Yulianto, A., Sukarno, P., Suwastika, N.A., 2019. Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset. In: Journal of Physics: Conference Series, Vol. 1192, No. 1. IOP Publishing, 012018.

Zhang, X., Chen, J., Zhou, Y., Han, L., Lin, J., 2019. A multiple-layer representation learning model for network-based attack detection. IEEE Access 7, 91992–92008.

Zhang, H., Huang, L., Wu, C.Q., Li, Z., 2020. An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. Comput. Netw. 177, 107315.

Zhao, J., Jing, X., Yan, Z., Pedrycz, W., 2021. Network traffic classification for data fusion: A survey. Inf. Fusion 72, 22–47.

Zhou, X., Hu, Y., Liang, W., Ma, J., Jin, Q., 2020. Variational LSTM enhanced anomaly detection for industrial big data. IEEE Trans. Ind. Inform. 17 (5), 3469–3477.