Evaluating Deep Sequential Models for Flow-Based Network Intrusion Detection Systems

1st Jeong Hoon Choi
Viterbi School of Engineering
University of Southern California
Los Angeles, United States
csian7386@gmail.com

2nd Yuxuan Liu
Viterbi School of Engineering
University of Southern California
Los Angeles, United States
lyx2023@gmail.com

***Abstract*—Traditional Network Intrusion Detection Systems (NIDS) treat each network flow as an isolated sample, ignoring the temporal patterns that characterize many sophisticated attacks. This work systematically evaluates three deep learning architectures for sequential flow-based intrusion detection: a baseline Fully Connected Network (BaselineFCN), a hybrid CNN+LSTM model, and a novel BERT-based architecture (NetFlowBERT). We conduct experiments on two standardized NetFlow datasets: NF-UNSW-NB15-v2 and NF-CSE-CIC-IDS2018-v2. Our results demonstrate that sequential modeling significantly improves detection performance, with NetFlowBERT achieving 83.44% accuracy and 0.6389 macro F1-score on CIC-IDS2018, representing a 65% improvement over the non-sequential baseline. The BERT-based approach excels particularly in detecting DoS attacks, achieving an average F1-score of 0.8925. This work provides empirical evidence that capturing temporal dependencies in network flows substantially enhances intrusion detection capabilities, especially for attacks that unfold over multiple flows.**

***Index Terms*—Network Intrusion Detection, Sequential Models, BERT, LSTM, NetFlow, Deep Learning**

I. Introduction

Many models utilizing Machine Learning and Deep Learning have been researched and used to detect anomalies in network packets. Models that analyze the network packets themselves at the application layer, as well as various models that target summarized flow data, have been primarily used, and these have already demonstrated excellent performance with accuracy exceeding +95%. However, these models face challenges in the classification problem of accurately predicting the type of attack.

First, benign traffic accounts for an overwhelmingly large proportion of the total network traffic. Various attacks, in comparison, represent a very small number. Second, there are limitations in classifying an attack based on a single network flow. Current attacks attempt complex attacks composed of various flows to evade AV and EDR detection, making classification impossible with only a few flows. Accordingly, various models have recently been developed that treat the flow as a time series data and use appropriately sized windows as input, aiming to improve the prediction performance for labels that were difficult to predict with existing models that used a single flow as input.

Our research was inspired by Manocchio [1], and we compared and analyzed the performance of three models: BaselineFCN (which uses a single flow as input), CNN+LSTM (which uses a flow sequence as input with CNN and LSTM layers), and NetFlowBERT (which uses a Transformer Encoder Only model, BERT, with a flow sequence as input). Ultimately, we trained the NetFlowBERT model and confirmed that it classified classes that were difficult to predict with existing models with higher accuracy.

II. Related Work

A. NetFlow Datasets

Various datasets exist for analyzing network traffic, and they have different features depending on what information from the packets is used for summarization. For example, UNSW-NB15 and CIC-IDS2018, used in our project, have 49 and 80+ feature dimensions, respectively. Sarhan et al. [2] proposes creating NetFlow data from PCAP files using a standardized format for features based on the 5-tuples with nProbe. Our research uses version 2 data, NF-UNSW-NB15-v2 and NF-CSE-CIC-IDS-2018-v2, which have 43 standardized common NetFlow features.

Table I shows the comparison between original datasets and NetFlow v2 versions.

TABLE I
Dataset Comparison - Original vs NF-v2

Dataset	Format	Level	Feat.	Inst.	Miss.	Pay.
UNSW-NB15	PCAP	Packet	49	2.5M	Yes	Yes
CIC-IDS2018	PCAP/CSV	Packet	80+	16M	Yes	Yes
NF-UNSW-v2	NetFlow	Flow	43	2.39M	No	No
NF-CIC-v2	NetFlow	Flow	43	18.89M	No	No

B. Flow Transformers

The Flow Transformers architecture proposed by Manocchio et al. [1] is broadly divided into Input Encoding, Transformer, and Classification Heads components. By modularizing these components, the model allows for the application of Transformer models to NetFlow data, facilitates easy model replacement, and enables comparative analysis of model performance.

This paper provides a Python library for comparing performance, which aims to quickly compare model performance by training the model using a subset of the data. However, due to a lack of optimization in the process of generating the sequence flow and training the model, it suffered from the problem of not being able to utilize more than 5% of the GPU. We applied the same Categorical Embedding Input Encoding module used in the paper, and then developed and used our own Transformer and Classification Heads.

C. CNN-LSTM Hybrid Models

To analyze the performance of sequence flow, we referenced and compared it to the CNN+LSTM model developed by Sun and Liu [3], which combines spatial feature extraction from CNNs with temporal modeling from LSTMs.

III. Methodology

A. Datasets

We utilize two standardized NetFlow v2 datasets shown in Table II.

TABLE II
Dataset Statistics

Dataset	Size	Instances	Features	Classes
NF-UNSW-NB15-v2	421M	2,390,275	43	10
NF-CIC-IDS2018-v2	3.0G	18,893,708	43	8

The NF-UNSW-NB15-v2 dataset contains 10 classes in total, consisting of 1 benign class and 9 attack categories: Generic, Exploits, Fuzzers, DoS, Reconnaissance, Analysis, Backdoor, Shellcode, and Worms.

The NF-CSE-CIC-IDS2018-v2 dataset contains 8 classes, consisting of 1 benign class and 7 attack categories: Brute Force, Bot, DoS, DDoS, Infiltration, and Web Attacks.

Both datasets exhibit severe class imbalance, with benign traffic accounting for 96% and 88% of flows respectively.

1) Challenging Attack Types: It is known that achieving high accuracy in detecting Analysis, DoS, and Reconnaissance attacks in the NF-UNSW-NB15 dataset is difficult. Analysis attacks involve packet manipulation tests and server response tests to analyze IDS/IPS systems installed on the target OS. Because these attacks utilize numerous packets that appear benign, classification is extremely challenging. DoS attacks, including DDoS attacks, involve sending a massive number of packets to a server in a short period to overwhelm and disable it. Due to the rule of network flows format, where each connection to a destination server is treated as a separate flow, classifying high-volume DDoS attacks as a single flow is impossible.

In NF-CSE-CIC-IDS2018, Brute Force and DDoS attacks are known to be difficult to classify. While Brute Force attacks on specific ports like FTP (TCP/21), SSH (TCP/22) are relatively easy to predict, Brute Force attacks on web services are known to be challenging. However, web services have less consistent failure patterns, and classifying them using single-flow data to distinguish them from normal login failures (Benign traffic) is extremely difficult. DDoS attacks like HOIC DDoS, which mimic benign network traffic, are difficult to classify using single-flow data.

B. Data Preprocessing

In the NetFlow version 2 dataset, categorical features such as Port and Protocol were transformed into vectors of up to 32 dimensions using categorical encoding. After transformation, the feature sizes of NF-UNSW-NB15-v2 and NF-CSE-CIC-IDS2018 were 269 and 291, respectively.

The data was divided into training, validation, and test datasets in a 70:15:15 ratio, regardless of whether it was non-sequential or sequential data, using stratified sampling. As a result, the same test dataset could be used for all three models.

Using NumPy's `sliding_window_view` function, the sequential flow data could be used without any memory overhead, allowing both non-sequential and sequential data to be trained without bottlenecks. The entire NF-UNSW-NB15-v2 dataset (3.8GB) was used, while only the first 3M flows (5.1GB) of the NF-CSE-CIC-IDS2018 dataset were used for training and evaluation.

C. Model Architectures

To ensure a fair comparison of model performance and to prevent excessive model complexity, the models were designed with parameters that were as similar as possible.

1) BaselineFCN: A 5-layer Fully Connected Network, BaselineFCN, was designed to represent the performance of a single flow. It consists of five dense layers with hidden sizes [64, 128, 256, 512, 256], ReLU activation, and dropout rate of 0.3. The model has 324,106 parameters.

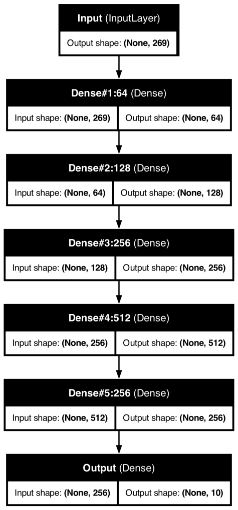

Fig. 1. BaselineFCN Architecture

2) CNN+LSTM: The CNN+LSTM model processes sequences of 8 flows. It combines two 1D convolutional layers (filters: [64, 128], kernel size: 3) with two LSTM layers (units: [128, 64]). Dropout rate is 0.3. The model has 341,642 parameters.

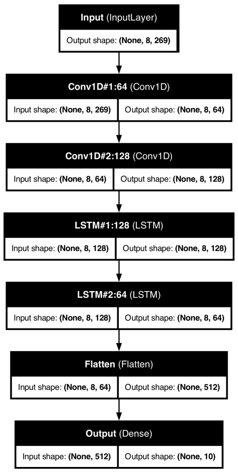
Input (InputLayer)
Output shape: (None, 8, 269)
Conv1D#1:64 (Conv1D)
Input shape: (None, 8, 269)
Output shape: (None, 8, 64)
Conv1D#2:128 (Conv1D)
Input shape: (None, 8, 64)
Output shape: (None, 8, 128)
LSTM#1:128 (LSTM)
Input shape: (None, 8, 128)
Output shape: (None, 8, 128)
LSTM#2:64 (LSTM)
Input shape: (None, 8, 128)
Output shape: (None, 8, 64)
Flatten (Flatten)
Input shape: (None, 8, 64)
Output shape: (None, 512)
Output (Dense)
Input shape: (None, 512)
Output shape: (None, 10)

Fig. 2. CNN+LSTM Architecture

3) NetFlowBERT: We designed the NetFlowBERT model using 4 stacked BERT layers with 2 self-attention heads each. The model has embedding size 64 and internal feed-forward network size 256. Dropout rate is 0.1. The model has 284,170 parameters.

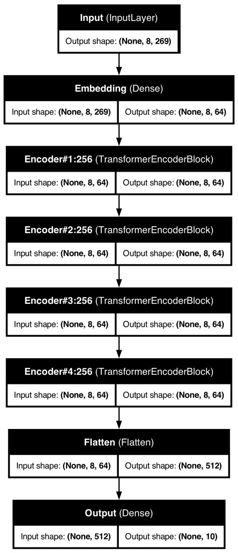
Input (InputLayer)
Output shape: (None, 8, 269)
Embedding (Dense)
Input shape: (None, 8, 269)
Output shape: (None, 8, 64)
Encoder#1:256 (TransformerEncoderBlock)
Input shape: (None, 8, 64)
Output shape: (None, 8, 64)
Encoder#2:256 (TransformerEncoderBlock)
Input shape: (None, 8, 64)
Output shape: (None, 8, 64)
Encoder#3:256 (TransformerEncoderBlock)
Input shape: (None, 8, 64)
Output shape: (None, 8, 64)
Encoder#4:256 (TransformerEncoderBlock)
Input shape: (None, 8, 64)
Output shape: (None, 8, 64)
Flatten (Flatten)
Input shape: (None, 8, 64)
Output shape: (None, 512)
Output (Dense)
Input shape: (None, 512)
Output shape: (None, 10)

Fig. 3. NetFlowBERT Architecture

Table III summarizes the three models.

TABLE III
MODEL ARCHITECTURE COMPARISON

Aspect	**Baseline FCN**	**CNN+ LSTM**	**NetFlow BERT**
Input	Single flow	8-flow seq.	8-flow seq.
Temporal	None	LSTM	Bi-attn.
Arch.	5-layer FCN	2 Conv1D + 2 LSTM	4-layer BERT
Dropout	0.3	0.3	0.1
Batch	64	256	256
Params	324K	342K	284K

D. Training Configuration

To minimize problems caused by data imbalance during the learning process, we trained the model using balanced class weights and implemented and used a custom Sparse Categorical Focal Loss function. Furthermore, in the evaluation of the learning process, we focused on Macro F1 and Micro F1 scores rather than overall accuracy.

We trained all models for 300 epochs with early stopping (patience: 5) and used Adam optimizer with default learning rate.

IV. EXPERIMENTAL RESULTS

A. Hardware and Software

We used the Nvidia RTX 4060, a GPU commonly used in home computing, for model training and evaluation. Training times were under 12 hours per model. We used Python 3.13.7 and TensorFlow 2.20.0, running on Arch Linux. Code is available at: https://github.com/csian98/Anomaly-Detection

B. Performance on NF-UNSW-NB15-v2

We performed training and evaluation using models with the parameters described above. BaselineFCN consists of dense layers and takes a single flow as input. CNN+LSTM and NetFlowBERT take 8 flow sequences as input.

In the NF-UNSW-NB15 dataset, the overall accuracy of all models showed similar levels, exceeding 98%. However, significant differences were observed in the Macro F1 score, with BaselineFCN, CNN+LSTM, and NetFlowBERT showing progressively better performance (Table IV).

TABLE IV
OVERALL PERFORMANCE ON NF-UNSW-NB15-V2

Model	**Macro F1**	**Accuracy**
BaselineFCN	0.5072	98.16%
CNN+LSTM	0.5888	98.41%
NetFlowBERT	0.6233	98.39%

When examining performance by attack class, significant performance differences were observed between BaselineFCN, CNN+LSTM, and NetFlowBERT (Table V). Specifically, for the Analysis attack, BaselineFCN completely failed to learn the pattern, while CNN+LSTM and NetFlowBERT showed improved performance in prediction.

TABLE V
PER-CLASS PERFORMANCE ON NF-UNSW-NB15-V2

Model	**Analysis**		**DoS**		**Recon**	
	Rec	**F1**	**Rec**	**F1**	**Rec**	**F1**
BaselineFCN	0.00	0.01	0.21	0.16	0.79	0.82
CNN+LSTM	0.54	0.30	0.39	0.30	0.82	0.82
NetFlowBERT	0.59	0.62	0.50	0.27	0.88	0.87

C. Performance on NF-CSE-CIC-IDS2018-v2

The NF-CSE-CIC-IDS2018 dataset yielded somewhat different results than expected. While the Macro F1 score was generally similar across models, we observed an improvement in the order of CNN+LSTM, BaselineFCN, and NetFlowBERT. Accuracy, however, showed a significant improvement in the order of BaselineFCN, CNN+LSTM, and NetFlowBERT (Table VI).

TABLE VI
OVERALL PERFORMANCE ON NF-CSE-CIC-IDS2018-V2

Model	Macro F1	Accuracy
BaselineFCN	0.6190	50.58%
CNN+LSTM	0.5770	64.45%
NetFlowBERT	0.6389	83.44%

This large difference in accuracy is likely due to the poor performance of the BaselineFCN and CNN+LSTM models in predicting Benign traffic. On the other hand, CNN+LSTM showed relatively high performance for Brute Force-HTTP and DDoS HOIC attacks (Table VII).

TABLE VII
PER-CLASS PERFORMANCE ON NF-CSE-CIC-IDS2018-V2

Model	Benign		BF-HTTP		HOIC	
	Rec	F1	Rec	F1	Rec	F1
BaselineFCN	0.50	0.66	0.20	0.02	0.10	0.18
CNN+LSTM	0.65	0.79	0.48	0.00	0.24	0.38
NetFlowBERT	0.87	0.93	0.02	0.00	0.10	0.18

D. Enhanced NetFlowBERT Performance

To evaluate the final performance of the NetFlowBERT model, we trained it using larger weights and consolidated the classes into more detailed labels. The model consists of 8 transformer encoder layers with 8 attention heads each, and the internal FNN has 1,024 parameters each.

As a result, we achieved an accuracy of over 99% on both datasets and significantly improved macro F1 scores of 0.78 and 0.73 from NF-UNSW-NB15 and NF-CSE-CIC-IDS2018 (Table VIII).

TABLE VIII
ENHANCED NETFLOWBERT PERFORMANCE

Dataset	Micro F1	Macro F1	Accuracy
NF-UNSW-NB15-v2	0.9923	0.7779	99.23%
NF-CIC-IDS2018-v2	0.9951	0.7316	99.51%

V. DISCUSSION

Traditional Machine Learning or Deep Learning models that used single-flow as input successfully predicted some attacks, including Benign traffic, with high accuracy, but showed lower prediction accuracy for other attacks.

Our results show that sequential modeling significantly improves detection performance. The sequential models (CNN+LSTM and NetFlowBERT) consistently outperformed the non-sequential baseline, with improvements ranging from 27% to 65% in accuracy.

The benefits of sequential modeling are particularly evident for attacks that distribute malicious behavior across multiple flows. For Analysis attacks in NF-UNSW-NB15, BaselineFCN achieved essentially zero performance (F1: 0.01), while NetFlowBERT achieved reasonable detection (F1: 0.62).

NetFlowBERT generally achieved the best overall performance, particularly on the more challenging CIC-IDS2018 dataset where it achieved 83.44% accuracy compared to 64.45% for CNN+LSTM and 50.58% for BaselineFCN.

VI. FUTURE WORK

While this showed fast speeds, allowing for relatively short training times (< 12 hours) and real-time partial inferencing, it clearly had limitations when it came to training larger models and datasets. If we use AI GPU accelerators to train larger models and make appropriate compromises in inference, it will be perfectly usable as a server-side EDR solution.

Furthermore, it is expected that using upsampling or downsampling techniques such as SMOTE will improve the F1 Macro score, and it will be possible to predict attack patterns that could not be learned previously due to the limited amount of data. Additionally, research on inferring data trained on different NetFlow format data is also considered necessary in the future.

VII. CONCLUSION

We applied three models BaselineFCN, CNN+LSTM and NetFlowBERT with similar numbers of parameters and complexity to compare and analyze the performance between non-sequential flow models and sequential flow models.

Our results demonstrate that sequential modeling significantly improves detection performance, with NetFlowBERT achieving 83.44% accuracy and 0.6389 macro F1-score on CIC-IDS2018, representing a 65% improvement over the non-sequential baseline. The enhanced NetFlowBERT model with 8 layers and 8 attention heads achieved over 99% accuracy on both datasets.

These results suggest that BERT-based sequential modeling represents a promising direction for network intrusion detection systems, particularly for detecting sophisticated attacks that evade traditional single-flow analysis.

REFERENCES

[1] L. D. Manocchio, S. Layeghy, W. W. Lo, G. K. Kulatilleke, M. Sarhan, and M. Portmann, "FlowTransformer: A transformer framework for flow-based network intrusion detection systems," *Expert Systems with Applications*, vol. 241, p. 122564, May 2024.

[2] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems," in *Big Data Technologies and Applications*, Springer International Publishing, 2021, pp. 117–135.

[3] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, and J. Chen, "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," *Security and Communication Networks*, vol. 2020, Article ID 8890306, 2020.

[4] M. S. Sakib and N. Tabassum, "Analyzing Deep Learning Model Performance for Intrusion Detection on CIC-IDS2017 Dataset," *SSRN Electronic Journal*, April 2025.