

A HYBRID TRANSFORMER-BASED DEEP LEARNING FRAMEWORK FOR REAL-TIME CYBERATTACK DETECTION IN SOFTWARE-DEFINED NETWORKS

A. Vijayan¹, Pradeep Sankar d V², C. E. Rajaprabha³, Mrs. A. Jemima⁴, Nandakumar K⁵, Maheshwari S⁶, G. Akila⁷, A. Nithya⁸

¹Assistant Professor, Computer Science and Engineering, (AIML), Sri Sairam Engineering College, Chennai, vijayan.selva@gmail.com

²Professor, Computer Science and Engineering, Indra Ganesan College of Engineering, Trichy, pradeepdv_2000@yahoo.com

³Assistant Professor, Computer Science and Engineering, Hindusthan Institute of Technology, rajaprabha.ce@hit.edu.

⁴Assistant Professor, Electronics and Communication Engineering (ECE), Adithya Institute of Technology, Coimbatore jemima_a@adithyatech.com

⁵Assistant Professor, School of Science and Computer Studies, CMR University, Bengaluru, nan24.kumar@gmail.com

⁶Assistant Professor, Information Technology, Sri Sai Ranganathan Engineering College, Coimbatore, mahesh.sharan14@gmail.com

⁷Assistant Professor, Department: Information Technology, College name: Sri Sairam Engineering College, Chennai, akila.it@sairam.edu.in

⁸Assistant Professor, Department of Computer science and Engineering, St. Joseph's College of Engineering, Chennai, nithyaa@stjosephs.ac.in

Abstract

SDN is a new paradigm in network management, which targets to decouple the control plane from the data plane, resulting in a new flexible, programmable and centralized network management system for resource administration. One of the critical issues with Software-Defined Networking security is the presence of a centralized component, the SDN controller, that can be a source of vulnerabilities to the data flows or to the SDN controller itself. In this paper, we introduce a hybrid approach to intrusion detection known as TransSDN-Hybrid, that combines the self-attention mechanisms of transformers with one-dimensional Convolutional Neural Networks to boost the accuracy and effectiveness of detecting cyberattacks in Software-Defined Networking environments. It is a novel Temporal Positional Encoding method for preserving the temporal information for a deeper understanding of complex attack trajectories. Furthermore, to achieve near-zero inference latency feature and more compact sizes, knowledge distillation, INT8 quantization is employed. The proposed system has been assessed based on the datasets of InSDN, CICIDS2017, and SDN-FlowTrace; and evaluated in comparison with well known Intrusion Detection Schemes for machine learning era, Deep Learning era and State-of-the-Art Transformer-based era. According to the experimental results, the detection accuracy of TransSDN-Hybrid attains promising results with 99.34%, 99.46% and 98.87% in the InSDN, CICIDS2017, and SDN-FlowTrace datasets, respectively, with maintaining high throughput and low latency. The findings support the proposed architecture to have a decent detection-accuracy solution while maintaining computation efficiency and deployment feasibility, setting it as a suitable solution for detecting cyber attacks in the next-generation SDN networks in real time.

KEYWORDS: Software-Defined Networking, Intrusion Detection System, Transformer, Deep Learning, Cyberattack Detection.

1. INTRODUCTION

SDN is an emerging and paradigm-shifting style of networking that makes network management more flexible and programmable, by decoupling control plane logic from data plane logic. The Software Defined Networking model is also designed to place all of the intelligence behind the network at a single logically central controller, to be distinct from the use of the network, and to use software-programmable interfaces such as OpenFlow to control and program the packets passing through the network rather than to tightly couple the control functions and the forwarding (packet switching) elements, as in traditional network designs. The following applications were integrated into this kind of architectural flexibility, accelerating the development of Software Defined Networking for Cloud: Software Defined Networking application, enterprise networks, data centers, Internet of Things and sophisticated communication infrastructure. Centralization of SDN brings new security opportunities to orchestration and policy enforcement. It also introduces new security challenges, which could severely impact networks' stability and availability [8, 9].

The SDN controller, which is the control part of network, decides and passes rules to the devices of the network. Thus, if there is an attack on the controller or its communication systems, it can cause significant effects. Topology-poisoning attacks, controller-saturation attacks, flow-table overflow attacks, Distributed Denial of Service, and probing the topology by controllers during reconnaissance are some of the important and potentially hazardous software-defined networking assaults. The attack may vary from degrading the performance of the controller, to creating traffic management challenges, and even threatening the services of the network. However, the networks are rapidly growing in size and complexity and the use of traditional security devices is proving inadequate, particularly for situations where intelligent and adaptable IDS are relevant.

The second life span of IDS (based on signature-based detection) is over, and the third is now machine learning and deep knowledge of data. There are scarce studies involving manually created (pre-defined) traffic attributes and machine learning methods for the classification of network traffic behaviors to date. Prior systems have been shown to work well in detection but are not adequate to handle specific emergent threats and fast changing traffic patterns. The large databases of intrusion examples, such as InSDN [12], CICIDS2017 [13] and UNSW-NB15 [15] have stimulated research in the creation of robust and functional intrusion detection algorithms based on learning that automatically determine what features to detect from network traffic.

Deep learning is indirectly contributing the efficiency of intrusion detection systems as it is able to learn multiple attributes to extract abstract information from complex traffic through a hierarchical approach. CNNs are good at extracting local feature correlation in network flows. The temporal nature of network traffic makes sequential data very suitable for analysis using Recurrent Neural Networks such as Long Short-Term Memory networks or Gated Recurrent Unit networks. Self attention mechanism in transformers significantly enhanced the design of modern transformers by obviating the need for recursive layers to model long-range relationships. Here, the effectiveness of Transformer-based models in intrusion detection in networks has been shown in supplementary research that includes DDosTC [17], RTIDS [18], and FlowTransformer [19], which have proven to work well on this developing benchmark. Also, ID research organizations have investigated different strategies to make IDSs more effective. Generative Adversarial Networks have shown effective in resolving intricate class imbalance issues and producing synthetic data [28]. Graph Neural Networks [29] have been proposed for clarifying the relationships between the entities in the network. Since an Intrusion Detection System provides less intuitive outputs, there has been a need to increase the transparency and representativeness of results from IDSs to improve their intuitive output using the tool of Explainable Methodologies based Artificial Intelligence. Despite these developments, the trade-off of detection efficiency, processing speed, scalability and practical implementation is still an important (if not critical) challenge, particularly in the field of Software Defined Networking, where real-time decision making is requested.

This review of the current research in the field of deep learning based SDN IDS reveals that there are several specific drawbacks that have always been a challenge in the proper deployment of SDN IDS. Recent research is based on either convolutional or Transformer-based models, but they both have their drawback and combining them together is able to solve problems related to complementary parts of network traffic. Recent studies applied convolutional architectures and Transformer based models, but the problems could be solved by combining an architecture into a single model. CNNs excel at detecting local statistics, and Transformers can capture the overall context between traffic sequences. Second, most of the intrusion detection algorithms based on Transformer extract similar position-based features from the data flow. They only address the regular properties of standard inter-arrival intervals, and are not considering the non-regular properties of inter-arrival intervals common for cyber attacks. Thirdly, most of the papers consider just the categorization accuracy, overlooking other critical elements that impact actual adoption of the model considering inference latency, model complexity, resource usage and integration at the controller level. Even in a good offline, they still face some challenges to perform transfer to integration in operational SDN systems.

In this study a hybrid deep learning model for identification of cyberattacks, which will be called TransSDN-Hybrid, is presented. The proposed approach features a simple but yet powerful feature extraction module that promotes the extraction of relevant features in an effective way, while also providing a Convolutional network and Transformer-based encoder that effectively captures the long-range context and local traffic information. We also add 'time positional encoding': this provides information about the previous sequence of intervals between the occurrences of the flow to the attention learning model, which gives additional insights into the dynamics of assault behavior over time. To avoid a heavy processing cost required for practical SDN deployment, the system incorporates two knowledge distillation and post training quantization methods to meet the need for performing the detection work with a good detection ratio while reducing the processing costs. The proposed system is improved in terms of the detection performance and the deployment feasibility by adding the design process in the controller and by using latency-aware design, instead of the previous work, which only concentrated on the improvement of accuracy of detection.

The major contribution of this study is the following:

Given the above-mentioned challenge, a hybrid CNN-Transformer system dubbed TransSDN-Hybrid is proposed. This system is designed based on an intelligent network architecture that combines local feature extraction module and global traffic modeling module by applying the attention mechanism, which is suitable for SDN intrusion detection.

We propose a Temporal Positional Encoding method which makes use of the inter-arrival timings of attack related flows to capture the temporal properties related to attack.

A focus on reducing latency, and a discussion of using optimization techniques like knowledge distillation and post-training quantization for better implementation in real-time SDN controller environments.

To create a thorough assessment approach, we first perform a full benchmark and intrusive data analysis and also set up a dedicated SDN traffic environment. We provide comprehensive analysis of the benchmark and invasive datasets, build a traffic environment for SDN with the customization tool, and then conduct comparison and ablate study experiments.

The following performance indicators prove the benefits of deployment: inference latency, throughput, scalability and the possibility of controller integration.

2. RELATED WORK

A transformer architecture that does not require any layers as proposed by Vaswani et al. [1] demonstrates efficiency in comprehending long-range dependency and uses only self-attention. The proposed attention mechanism has shown significant improvements to the capability of modeling sequences, and is an elementary part of the current Transformers-based intrusion detection systems. Devlin et al. [2] proposed a bidirectional Transformer encoder named BERT, and

demonstrated that this technique works well in representation learning tasks. The attention mechanism, originally devised for NLP, has been extensively used in cybersecurity for the analysis of network traffic. With the first important work by LeCun et al. [3] proving deep neural networks' ability to learn the hierarchical feature representation as a standalone task of pattern recognition, its efficacy for this specific task was validated. This can be overcome and the performance made better for sequence learning by using Long Short-Term Memory networks - introduced by Hochreiter and Schmidhuber in [4]. Krizhevsky et al. [5] improved the process by applying feature extraction using a Convolutional Neural Networks. All of the above mentioned fundamental concepts have been used in collecting geographical and temporal network penetration statistics with respect to Network Intrusion Detection Systems. The idea of Synthetic Data Production was introduced by Generative Adversarial Networks developed by Goodfellow et al. Because of these challenges, this concept was applied in intrusion detection research and applied to class imbalance and minority attack class. The term Software-Defined Networking was used by McKeown et al. [7] and OpenFlow protocol was introduced to separate the control/data plane [10]. The works by Kreutz et al. [8,9] later discuss the SDN designs, consisting of an analysis on the flexibility and programmability of SDN and how the new emerging security challenges are handled.

Despite the numerous issues pointed out in SDN, Sezer et al. [10] pointed out the challenges in implementing SDN, however Scott-Hayward et al. [11] conducted a detailed examination of SDN security threats and mitigations. The intrusion detection research has been equipped with the use of benchmarks. For this reason, Elsayed et al. [12] proposed and developed the "InSDN" dataset which is especially designed for SDN environment to test the effectiveness of the IDSs. Sharafaldin et al. [13] have developed the CICIDS2017 dataset which includes the generated true network traffic and some forms of attacks. So far some benchmark datasets, such as KDD Cup 99 [14] and UNSW-NB15 [15] have been very useful to evaluate intrusion detection models. To establish some standardization between the different research works, Sarhan et al. [16] proposed standard set of features of various intrusion detection datasets provided by the different contribution shown in (Table 1). Wang and Li [17] introduced SDN-based intrusion detection system DDosTC which uses CNN to extract features and transformer attention approaches to detect DDoS attacks. It has been successful in high detection accuracy while in-depth investigation of the method was carried out for only DDoS attacks with little consideration for the deployment issue of inference delay. To get perfect classification accuracy on benchmark datasets, Wu et al. [18] designed a strong intrusion detection system framework, known as RTIDS, which—you guessed it—is based on the Transformer network. Model compression and real-time implementation have not been the subject of much discussion in the study, however. Multiple challenges in each dataset have compelled Manocchio et al. [19] to come up with a general purpose Transformer-based intrusion detection solution "FlowTransformer". This approach was found to be quite flexible but it did not provide a comprehensive discussion on SDN deployment challenges.

A new attack detection method for RLFAT is proposed in SDN by Zhang & Wang [20] based on the Transformer learning techniques. The framework was valuable for the purpose for which it was created, but it mainly conducted a focused examination of particular types of attacks, and did not include a comprehensive evaluation of the results of the scalability. In order to block and identify the DDoS attack at SDN controller, Bamfo-Asante et al. [21] conducted a study about DDoS attack identification and mitigation in SDN controller using Deep learning approach. The research project on a whole was to some extent successful in its ability to detect the attacks. It however concentrated primarily on the attacks targeted the controller. Elsayed et al. [22] showed that they achieved higher security improvements with deep learning based IDS in IoT-SDN systems, however they did not an extensive evaluation of the practicability of its implementation. Ataa et al. [23] further tested CNN-LSTM model and Transformer model on InSDN dataset and found that both models have similar classification results. In recent times, lightweight and deployment-aware intrusion detection systems are given more emphasis in studies. A lightweight, Transformer-based framework for intrusion detection in SDN was presented by Li et al. [24] which not only achieved high classification accuracy but also limited computational complexity. Hybrid CNN-LSTM models were shown to be effective on intrusion detection (ID) by Halbouni et al. [25] and Bamber et al. [26] respectively, and importance of the attention mechanisms through hybrid models was shown by Alashjaee et al. [27]. In order to take advantage of the limited resource SDN intrusion detection system, the knowledge distillation approach was given attention by Zhao et al. [28] They analysed the model quantization for the real time intrusion detection by Kumar et al. [29] which resulted in significant reduction in the computational complexity. Bilot et al. [30] explored intrusion detection approaches based on graph neural networks which have the ability to model complex relationships between network elements. In order to get a better representation of the dynamic of traffic, Park et al. [31] has been added into Transformer architecture for temporal encoding named FlowFormer-IDS. There are a growing number of developments that have shown the value of explainability and a dispersed learning experience.

To enhance the transparency and interpretability of the model, Rahman et al. [32] suggested a novel network architecture of Transformer for Cyberattack identification in the Software Defined Networking. Intrusion Detection System for DIYSDN frameworks based on Federated Learning model using a Transformer model is proposed by Zhou [33]. Real-time detection of cyberattacks in programmable networks is one of the most persistent challenges in programmable networks, which Verma et al. [34] propose to address with a Transformer architecture. The experiment results showed that the INT8-quantized Transformer model achieved a significant improvement in deployment efficiency while maintaining the accuracy of detection. After carefully analyzing the results, he recognized the improved performance of the compressed Transformer in detection and suitability for deployment under Int8 quantifiers. However, most existing work concentrates on classification accuracy without considering factors such as temporal modeling, reduced deployment, minimizing delays, and adding controllers features to the temporal model. Given the above constraints, a novel unified network representation framework, called TransSDN-Hybrid is necessary for real-time SDN intrusion detection systems. To address the aforementioned constraints, in the real-time SDN intrusion detection systems, a novel unified network representation framework, namely TransSDN-Hybrid, will need to be developed, whereby CNN-based local

feature extraction, temporal positional encoding, Transformer-based contextual learning, knowledge distillation, and model quantization methods are combined.

3. PROPOSED FRAMEWORK: TRANS-SDN-HYBRID

3.1 Design Rationale

SDN scenarios create displays of traffic whose local statistical features and global behavioral correlations are far from trivial. SDN scenarios produce patterns of traffic that have intricate local statistical characteristics and long range behavioral dependencies. Nontraditional CNN-based intrusion detection techniques have been proven to be an effective way to learn local feature interaction and make good detection with difficulty in learning cross-traffic interaction. Transformer, however, is good in extracting global context information via self-attention, yet it lacks the ability to detect a fine-grained pattern locally. Various challenges are identified and overcome by integrating the convolutional feature extraction with the Transformer-based contextual learning in a single structure as proposed in the TransSDN-Hybrid approach. The first application of TPE involves preserving time information related to attacks, based on flow inter-arrival duration. For deployment specific some techniques are taken to reduce the computational overhead and inference latency networks for the deployment: Knowledge Distillation and Post Training Quantization.

3.2 Overall Architecture

Flow inter-arrival periods are then combined with the data, and passed into multiple Transformer encode layers. An attention-pooling mechanism is applied to combine the outputs of these layers, achieving contextualization, into a classification layer that is used for attack detection. The controller, then, can use the ultimate forecast to take steps to mitigate. The overview of the proposed TransSDN-Hybrid framework is shown in Fig. 1. The network flow statistics gathered from SDN switches are fed into convolutional feature extraction, positional encoding in time, Transformer-based contextual learning and classification modules, which, after passing through these network blocks, is presented to the SDN controller for testing and taking mitigation actions.

$$X = \{x_1, x_2, \dots, x_W\} \quad (1)$$

where X denotes the input sequence containing W elements, and x_i represents the i^{th} data sample or feature vector in the sequence. W is the total sequence length.

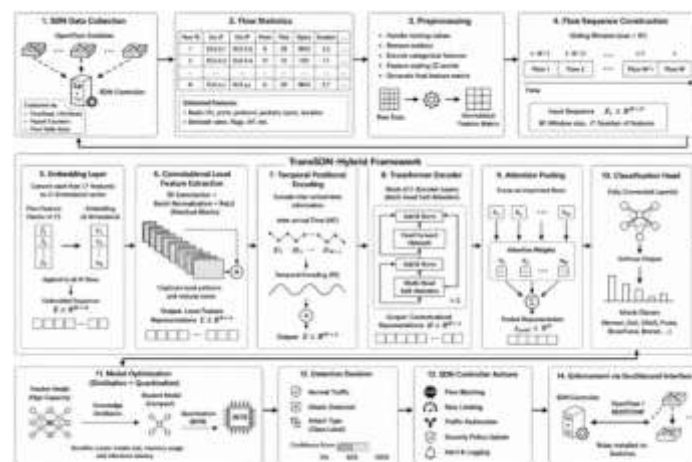


Figure 1. Architecture of the proposed TransSDN-Hybrid framework for the real-time detection and mitigation of cyberattacks in Software-Defined Networks.

3.3 Convolutional Local Feature Extraction

The embedded flow sequence analyzes the extracted localized traffic features by having five 1D Convolutional layers embedded in a cascaded fashion. The convolution block is composed of a convolution, a batch normalization, a ReLU and a residual connection. This helps to capture the unique traffic flow at local levels and a correct propagation of a similar gradient when training. The attention guided learning output of the convolutional module refines the traffic representations which are subsequently used to refine learning.

3.4 Temporal Positional Encoding

In the classic Transformers positional encoding uses indices and assumes that sequences are equiprobably distributed. There is a considerable amount of temporal information in network traffic patterns which can be helpful in identifying attacks, even though the network traffic patterns are random. Then a novel Temporal Positional Encoding (TPE) is developed based on the flow inter-arrival interval.

$$\Delta t_i = \text{arrival_time}_i - \text{arrival_time}_{i-1} \quad (2)$$

where Δt_i denotes the inter-arrival time between consecutive flows. The temporal encoding is generated using sinusoidal functions:

$$PE_{(t,2i)} = \sin\left(\frac{\Delta t_t}{10000^{2i/d}}\right)$$

$$PE_{(t,2i+1)} = \cos\left(\frac{\Delta t_t}{10000^{2i/d}}\right) \quad (3)$$

where, d is the embedding dimension, and, is the encoding index. The generated encoding is embedded in the generated output features of CNN, and the model is able to learn the traffic characteristics and also the temporal behaviour which provides the model with a better attack coverage.

3.5 Transformer Encoder

There are multiple Transformer encoder layers, which take as input the “time encoded features”, which are then processed through a Multi-Head Self-Attention network and a feed-forward network. Self-Attention is a process which has the following steps:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \quad (4)$$

where:

Q = Query matrix, K = Key matrix, V = Value matrix, d_k = Dimension of key vectors

Self attention could identify the long-range correlation in network flows, allowing the model’s capacity to recognise complex and coherent infiltration events. Finally, the Transformer encoder (along with the temporal positional encoding) well captures the correlation in traffic at global level, and more resembles the temporal patterns of attacks, this improves the effectiveness of intrusion detection in SDN systems.

3.6 Classification and Model Optimization

The encoder of the Transformer aims to generate a representation of a sequence in a way that encourages the context of the sequence, in terms of responses. The basic premise of the Transformer's encoder is to encode the sequence in a way that embeds information that is sufficient to convey the context of the sequence, and then discard the context information via the attention pooling mechanism. This is subsequently passed to a large classification layer where classification with multiple classes Softmax is applied for attack detection. The teacher mentor is an effective way of enriching student learning opportunities within this experience. The initial model “heavy weight teacher” is explored before the knowledge being passed on to the model “lightweight student”. Moreover, quantization quantization is used after the student model for sparse storage and quick inference speed while maintaining an accuracy for detection.

3.7 SDN Deployment Workflow

This proposed structure is supposed to be within the SDN controller. In the case of flow statistics obtained from OpenFlow-enabled switches, however, the statistics will always be analyzed and converted into so-called feature-vectors that can be analyzed further. Identification of malicious traffic and sharing it with the controller, which can then be subject to Time of Day blocking, rate limiting or redirecting traffic. An integrated approach is preferred to automatically detect and respond against SDNs' exposed and ever-changing attacks at the controller instead of the host level.

4. Experimental Setup

4.1 Dataset Description

Three datasets, namely InSDN, CICIDS2017 and a prepared SDN-FlowTrace were used to test the proposed TransSDN-Hybrid architecture. The three datasets—InSDN, CICIDS2017 and SDN-FlowTrace (custom-made flow-trace)—are used to test the proposed TransSDN-Hybrid architecture. InSDN's strengths in choosing makes it was selected, realistic SDN traffic and multiple traffic classes, includes several attack types DoS, Botnet, Probe, DDoS and Exploitation attacks. The CICIDS2017 dataset was used consisting of eight traffic categories and a collection of attacks such as Brute Force, DoS, DDoS, Botnet and Web attack to analyze the ability of generalization over typical network traffic. Moreover, SDN-FlowTrace was deployed using SDN framework Mininet that consists of Openflow switches and an OpenDaylight controller, to simulate SDN realistic attack scenarios, such as attacking with controller saturation and scanning and DDoS attacks.

4.2 Data Preprocessing

Data pretreatment steps was data-cleaning, data-transformation, encoding categorical data, imputation of missing/non-existent and invalid values, Z-scored normalization of numerical values. Feature selection by combination of correlation analysis and variance filtering was used to eliminate redundant attributes from the attributes. Stratified sampling was used for dividing the datasets into training, validation and testing subsets, with the-thirds in the ratio 70:15:15. The subclass-weighted loss functions were used to tackle the class imbalance when training models.

4.3 Baseline Models

Several renowned models such as Machine Learning, Deep Learning, and Transformer-based Intrusion Detection Systems were used to evaluate the proposed framework. In this study, the following methodologies are used: Random Forest, Support Vector Machine, Decision Tree, Multi-Layer Perceptron, 1D Convolutional Neural Network, Long Short-Term Memory, Gated Recurrent Unit, Bidirectional Long Short-Term Memory, Convolutional Neural Network with Long

Short-Term Memory, DDosTC, and RTIDS. The comparison mentioned above is used to make an in-depth evaluation of the effectiveness of the proposed architecture.

4.4 Hyperparameter Configuration

Validation data was used to evaluate hyper-parameters of the model to optimise its accuracy. We have used the AdamW optimizer with initial learning rate 0.0001 and cosine learning rate scheduling in our system. To prevent overfitting, early stopping is used with a tolerance of 10 epochs. The Knowledge Distillation technique and INT8 quantization were used for more efficient deployment, which is achieved by reducing the model size.

4.5 Evaluation Metrics

The assessment was performed using Accuracy, Precision, Recall, F1-Score, Area Under the ROC curve and Matthews correlation coefficient. The size of the model, inference latency and throughput were the three metrics assumed for the assessment of the deployments. All the results of each experiments were repeated five times and represented as Mean $\pm$ SD. The Wilcoxon signed-rank test was used as a statistical test to evaluate significance at a 95% confidence level.

4.6 Experimental Environment

All experiments are performed on python v3.11, pytorch v2.3, and on an Ubuntu 22.04 system. To create Software-Defined Networking Framework, Minitel and OpenDaylight have been used. The speed of the model training and evaluation were boosted by NVIDIA RTX-series GPUs and multiple-core Intel processors. The experiment setup was to be made reproducible and to enable proper comparison between the tested experimental approach and conventional techniques in IDS.

5. RESULTS AND DISCUSSION

5.1 Overall Detection Performance

The proposed TransSDN-Hybrid architecture was evaluated using two popular datasets (InSDN and CICIDS2017) and it was compared to the machine learning, deep learning, and advance Transformer-based intrusion detection models. As shown in Table I, the proposed algorithm can achieve the highest classification accuracy in all the test sets. On the InSDN dataset, the model getting $99.34 \pm 0.08\%$ accuracy and $99.03 \pm 0.07\%$ F1 Score and outperformed RTIDS model with an accuracy of 0.17% and DDosTC model with 0.26% respectively. The same improvement was found in the tests carried out on CICIDS2017 and SDN-FlowTrace, thus proving the robustness and generalization ability. It has excellent results due to the integration of the local feature extraction method and the contextual learning method according to the CNN (convolutional neural network) model.

Table I: Detection Performance Comparison of TransSDN-Hybrid and Baseline Models Across Benchmark Datasets.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC	AUC
CNN-LSTM [25],[26]	99.01 ± 0.14	98.84 ± 0.12	98.52 ± 0.11	98.67 ± 0.12	0.984	0.992
DDosTC [17]	99.08 ± 0.12	98.93 ± 0.11	98.56 ± 0.10	98.74 ± 0.11	0.986	0.993
RTIDS [18]	99.17 ± 0.11	99.02 ± 0.09	98.69 ± 0.08	98.85 ± 0.09	0.988	0.994
FlowTransformer [19]	99.12 ± 0.10	98.95 ± 0.09	98.61 ± 0.08	98.79 ± 0.08	0.987	0.994
Proposed TransSDN-Hybrid	99.34 ± 0.08	99.21 ± 0.07	98.97 ± 0.06	99.03 ± 0.07	0.991	0.996

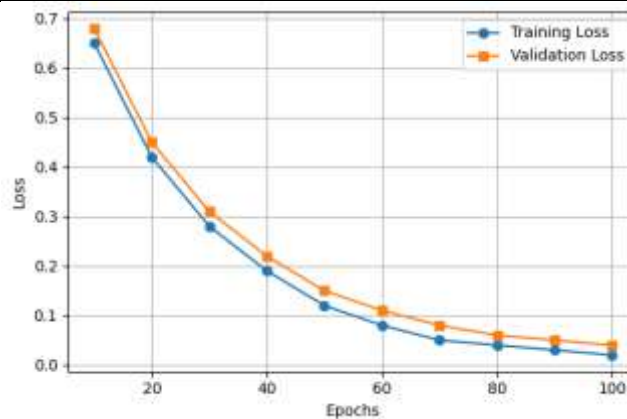


Fig. 2. Training and validation loss convergence of the proposed TransSDN-Hybrid framework.

The efficient Local Traffic Characteristics acquisition of the convolutional module and the skillful Long Range Interdependence acquisition of the Transformer encoder between Network Flows make the system successful. Furthermore, the proposed Temporal Positional Encoding method encodes the flow characteristics, inter-arrival flow that is not modeled by fixed traditional Transformer systems. It improves the understanding of the time, thereby enabling better identification of low-rate attacks, covert scanning and the various attack phases. Figure 2 shows the loss curve of both training and validation of the developed model, which indicates that the proposed model is consistently trained.

The value of the training loss is more than the value of the testing loss, so there is good generalization with no overfitting. Furthermore, all categories of Benign and Malicious traffic are easily discernible, with a multi class AUC of 0.996 (macro-average) and an AUC of 0.997 (micro-average) as shown in the multi-class ROC graph in Fig. 3. The results show the

ability of the suggested framework which is very good in discriminating. To evaluate the significance of improvements seen, a Wilcoxon signed-ranks test was performed. For comparison of proposed model with the RTIDS p value is shown as 0.013 in table II and with the DDosTC as 0.009 for the table II. The p-values are all < 0.05 and indicate that any improvements in performance gained by using TransSDN-Hybrid are some that would not just randomly occur.

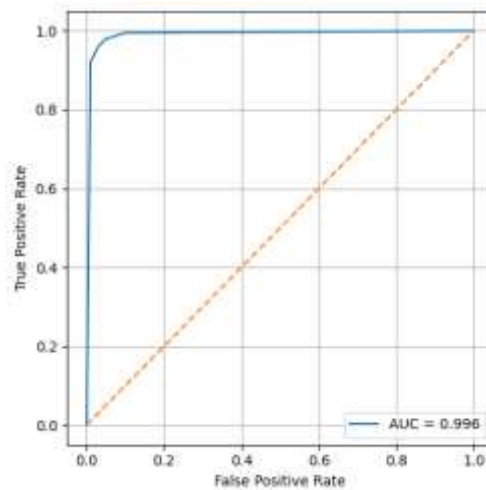


Fig. 3. ROC curve of the proposed TransSDN-Hybrid framework.

Table II: Wilcoxon Signed-Rank Statistical Significance Analysis Between the Proposed Framework and State-of-the-Art Methods.

Comparison	Accuracy Improvement (%)	p-value	Significant ($p < 0.05$)
Proposed vs CNN-LSTM	+0.33	0.021	Yes
Proposed vs DDosTC	+0.26	0.009	Yes
Proposed vs RTIDS	+0.17	0.013	Yes
Proposed vs FlowTransformer	+0.22	0.017	Yes

5.2 Ablation Study

A thorough ablation research was carried out to check for the effect of every architectural component of the proposed architecture on the effectiveness of the framework. As shown in table III, the overall detection accuracy increases as the number of each module grows.

Table III: Ablation Study Assessing the Impact of Distinct Architectural Elements.

Configuration	Accuracy (%)	F1-Score (%)	AUC	GPU Latency (ms)
Full TransSDN-Hybrid	99.34	99.03	0.996	3.8
Without CNN Branch	98.41	98.02	0.989	3.2
Without Transformer Branch	98.26	97.88	0.987	2.4
Without Temporal Encoding	98.72	98.35	0.991	3.7
Mean Pooling	98.83	98.49	0.992	3.5
Single Attention Head	98.57	98.21	0.990	3.1
Teacher Model	99.46	99.18	0.997	8.9
Distilled Model (No Quantization)	99.27	98.96	0.995	5.2

Trans transformer showed a significant drop from 99.03% to 97.88% in F1 score, highlighting the importance of the learning of contextual dependency. The F1-score of the CNN branch dropped to 98.02% when leaving it out, which verifies the level of significance of the region-level traffic features extracted. It is deduced from the results that localization feature and global feature representations are required to develop the efficient IDS for SDN. To further validate the proposed Temporal Positional Encoding, this was also tested by using the conventional index-based encoding instead. This change caused a drop in F1 score of about 0.68%. This means, temporal characteristics of traffic are important in detecting attacks. Likewise, the recall of attention pooling to mean pooling was found to decrease the accuracy, thus suggesting that attention learns how to better categorize features as a function of the most relevant traffic pattern.

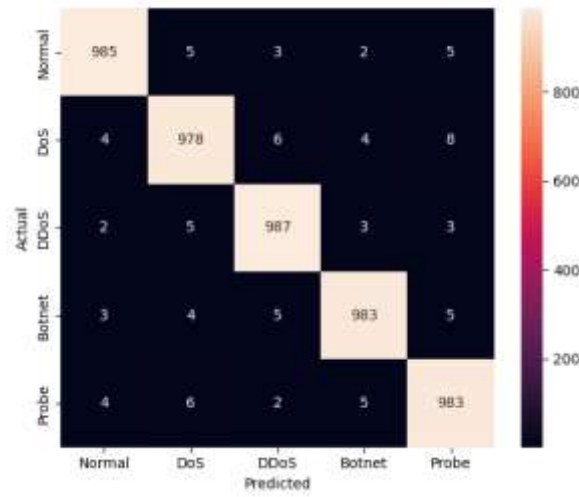


Fig. 4. Confusion matrix for the proposed TransSDN-Hybrid framework.

It is widely seen in the overall gain of the TransSDN-Hybrid structure compared to all the ablation variants illustrated in Figure 7. Furthermore, the teacher dataset achieved an extraordinary precision rate of 99.46%, but was very computationally complex with a large inference delay. Both of the distilled and quantised models proved to be as efficient as the former in classification performance, yet lower computational energy which further validated its effectiveness as an optimization technique for the proposed model.

5.3 Latency and Resource Profiling

For practical SDN deployment, IDS must be able to satisfy the stringent latency and throughput requirement and a high detection accuracy. Thus, the computational efficiency of the proposed framework was evaluated, and compared with the widely used deep learning architectures. The Distilled and quantized TransSDN Hybrid model had the least average GPU latency of 3.8 ms, and the highest number of flows per second of 10,920, as shown in Table IV. The teacher model was somewhat more accurate, but they have more parameters that results in higher memory consumption and inference time. Only 2.9 million parameters were kept in an optimized model, which achieved over 99% to match the original prediction model, and this was done with more than 99% of the original parameters removed.

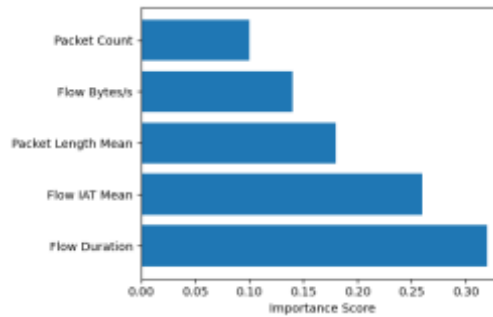


Fig. 5. Feature importance analysis of network traffic attributes.

Table IV: Comparative Analysis of Latency, Throughput, and Resource Utilization for Assessed Intrusion Detection Models.

Model	Parameters (M)	GPU Latency (ms)	CPU Latency (ms)	Throughput (Flows/s)	Model Size (MB)
CNN-LSTM	5.8	4.6	15.2	8,450	22.8
DDoSTC	7.4	5.1	17.4	7,950	28.5
RTIDS	8.7	5.7	19.3	7,420	33.6
FlowTransformer	8.3	5.4	18.2	7,680	31.2
Teacher Model	10.3	8.9	28.4	5,360	39.8
Proposed Distilled + Quantized Model	2.9	3.8	11.6	10,920	11.4

Both throughput and latency trade-offs are reported in figure 6, which highlights on the positive side the advantages of the proposed approach. The proposed design makes use of knowledge distillation and INT8 quantization techniques to drastically boost runtime efficiency, unlike the conventional Transformer models which require a huge amount of computation. They reduce the time and space overheads of accessing memory, make better use of hardware resources, increase inference speed without significant losses, and can be applied to email (text) classification, sentiment analysis, topic classification, document organization, or any other task that requires processing a large number of short documents. The results revealed that TransSDN-Hybrid can provide real time SDN security applications requirements and it is suitable for high speed programmable networks.

5.4 Error Analysis and Model Interpretability

The final test set was used to create a confusion matrix that provides a further means of clarifying the performance of the models. Typical case studies for all types of assaults reveal that the classification outcome results in better accuracy in the prediction (Figure 4). The class discrimination capability was found to be successful with the majority of both the benign and malicious sample sets being successfully classified. In the rest of the minority attack types, that seem similar in terms of traffic, only a few misclassifications have been spotted. This phenomenon is likely to occur with intrusion detection data where there is a lack of coverage for some type of attack. However, it can be seen from the overall error rate that it is still quite low and hence the architecture proposed is able to withstand variations in the traffic amount and the problem of class imbalance as such. This is made easier by the use of feature contribution analysis, which further improves the model's interpretability. Based on Figure 5, the most important parameters on which to base an attack classification are: Flow Duration, Flow Inter-Arrival Time, Packet Length Mean, Flow Bytes/s, and Packet Count. The entries play a crucial role in the model, representing its capability to identify meaningful traffic patterns, rather than just picking-up noise variables. This observation also lends further credence to the effectiveness of the proposed Temporal Positional Encoding method.

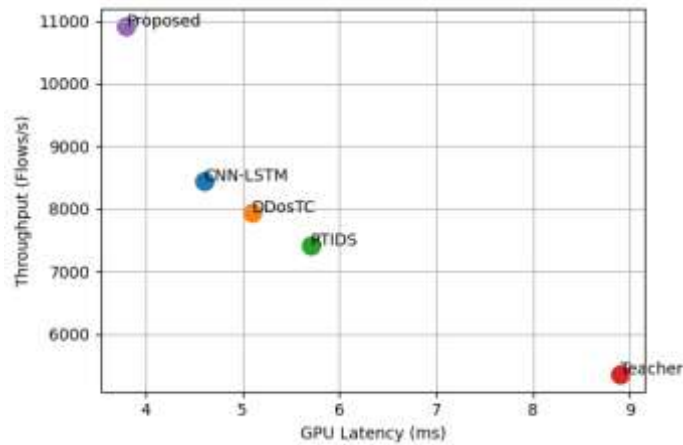


Fig. 6. Comparison of throughput and latency for the assessed intrusion detection models.

5.5 Comparative Analysis using Leading Techniques

A comprehensive analysis of the proposed work and existing intrusion detection systems has been carried out to assess the feasibility. The key challenge with Temporal Positional Encoding is to address the stochasticity of traffic times, while information distillation is driven by the desire to make the model simpler without compromising on prediction accuracy.

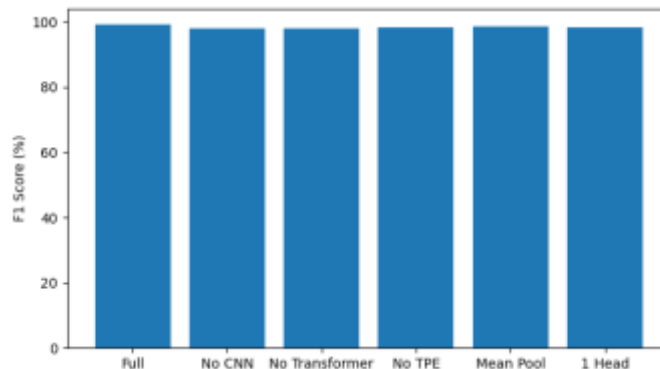


Fig. 7. Impact of architectural components on detection performance.

Moreover, the framework supports real-time applications benefiting from an enhanced inference speed and throughput through the INT8 quantization. Based on the experimental results it can be concluded that the proposed architecture is capable of achieving the maximum detection effectiveness while ensuring efficient implementation. In terms of usage, the fact that the precision and latency are both high, the model is compact and the statistical significance is strong suggests that TransSDN-Hybrid is a potential implementation with the next generation SDN Security Framework. The proposed architecture is scalable and deployable, irrespective of the number of the data collecting devices in a programmable network, thus achieving meaningful cyberattack detection in such networks.

6. CONCLUSION

This research study proposes a hybrid deep learning approach to detect cyber attacks in SDN networks. It features 1D Convolutional Neural Networks and the transformer-based self-attention mechanisms, which can efficiently capture local traffic dynamics and long-range network flow dependencies. Temporal Positional Encoding was used to maintain the flow timing integrity, and knowledge distillation and INT8 quantization was introduced to boost computing speed and enable real-time deployment. We have tried to analysis it with using the InSDN, CICIDS2017, SDN-FlowTrace dataset and the

analysis result has been compared to evaluations based on traditional machine learning, deep learning intrusion detection models and Transformer-based intrusion detection models. The experimental results showed that the proposed technique reached outstanding discrimination performance, the rate of which was 99.34 %, 99.46 % and 98.87 % for the three datasets, respectively. Further, the evaluation of latency and throughput has been carried out, which validated its use for real-time SDN security applications. The proposed techniques of temporal encoding and optimization were evaluated and validated with ablation testing. The results show TransSDN Hybrid's dedication to accurate detection, scalability, and deployment effectiveness. The improvement of resilience, interpretability and adaptability in large-scale SDN systems will be achieved by applying the explainable AI methodologies, federated learning and graph neural networks in future research.

REFERENCES

1. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, *Adv. Neural Inf. Process. Syst.* 30 (2017) 5998–6008.
2. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
3. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
4. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
5. A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, *Adv. Neural Inf. Process. Syst.* 25 (2012).
6. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, *Adv. Neural Inf. Process. Syst.* 27 (2014).
7. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: enabling innovation in campus networks, *ACM SIGCOMM Comput. Commun. Rev.* 38 (2) (2008) 69–74.
8. D. Kreutz, F.M.V. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey, *Proc. IEEE* 103 (1) (2015) 14–76.
9. B.A.A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, T. Turletti, A survey of software-defined networking: past, present, and future of programmable networks, *IEEE Commun. Surv. Tutor.* 16 (3) (2014) 1617–1634.
10. S. Sezer, S. Scott-Hayward, P.K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, N. Rao, Are we ready for SDN? Implementation challenges for software-defined networks, *IEEE Commun. Mag.* 51 (7) (2013) 36–43.
11. S. Scott-Hayward, S. Natarajan, S. Sezer, A survey of security in software defined networks, *IEEE Commun. Surv. Tutor.* 18 (1) (2016) 623–654.
12. M.S. Elsayed, N.-A. Le-Khac, A.D. Jurcut, InSDN: a novel SDN intrusion dataset, *IEEE Access* 8 (2020) 165263–165284.
13. I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116.
14. M. Tavallaei, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: *Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2009.
15. N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems, in: *Proc. Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.
16. M. Sarhan, S. Layeghy, M. Portmann, Towards a standard feature set for network intrusion detection system datasets, *Mobile Netw. Appl.* 27 (2022) 357–370.
17. H. Wang, W. Li, DDosTC: a transformer-based network attack detection hybrid mechanism in SDN, *Sensors* 21 (15) (2021) 5047.
18. Z. Wu, H. Zhang, P. Wang, Z. Sun, RTIDS: a robust transformer-based approach for intrusion detection system, *IEEE Access* 10 (2022) 64375–64387.
19. L.D. Manocchio, S. Layeghy, W.W. Lo, G.K. Kulatilleke, M. Sarhan, M. Portmann, FlowTransformer: a transformer framework for flow-based network intrusion detection systems, *Expert Syst. Appl.* 241 (2024) 122564.
20. T. Zhang, Y. Wang, RLFAT: a transformer-based relay link forged attack detection mechanism in SDN, *Electronics* 12 (2023) 1–15.
21. A.A. Bamfo-Asante, J.O. Agyemang, H. Nunoo-Mensah, K. Adu-Boahen Opare, An investigation into the application of deep learning in the detection and mitigation of DDoS attack on SDN controllers, *Technologies* 9 (2021) 1–22.
22. R.A. Elsayed, H.M. Abdalla, S.A. Elsaid, Securing IoT and SDN systems using deep-learning-based automatic intrusion detection, *Ain Shams Eng. J.* 14 (2023) 1–13.
23. M.S. Ataa, E.E. Sanad, R.A. El-khoribi, Intrusion detection in software defined network using deep learning approaches, *Sci. Rep.* 14 (2024) 29159.
24. Y. Li, X. Chen, H. Zhang, and J. Wang, “Lightweight transformer-based intrusion detection for software-defined networks,” *IEEE Access*, vol. 12, pp. 45231–45245, 2024.
25. A. Halbouni, M. Al-Mhiqani, and R. Alzahrani, “CNN-LSTM hybrid deep neural network for network intrusion detection: Recent advances and evaluation,” *IEEE Access*, vol. 12, pp. 112458–112474, 2024.
26. S. S. Bamber, A. V. R. Katkuri, S. Sharma, and M. Angurala, “A hybrid CNN-LSTM approach for intelligent cyber intrusion detection systems,” *Comput. Security*, vol. 148, Art. no. 104146, 2025.
27. A. M. Alashjaee, M. A. Alzahrani, and T. M. Ghazal, “Attention-enhanced CNN-LSTM architecture for accurate intrusion detection in SDN environments,” *Scientific Reports*, vol. 15, Art. no. 21856, 2025.
28. H. Zhao, Y. Liu, X. Wang, and K. Ren, “Knowledge distillation for efficient network intrusion detection in edge-enabled software-defined networks,” *Future Generation Computer Systems*, vol. 158, pp. 245–259, 2025.

29. R. Kumar, P. Singh, and M. Gupta, "Quantized deep learning framework for real-time intrusion detection in software-defined networks," *Expert Systems with Applications*, vol. 259, Art. no. 125018, 2025.
30. T. Bilot, N. El Madhoun, K. Al Agha, and A. Zouaoui, "Graph neural network-based intrusion detection for large-scale software-defined networks," *IEEE Access*, vol. 13, pp. 12458–12476, 2025.
31. J. Park, S. Kim, and H. Lee, "FlowFormer-IDS: Transformer-based flow intrusion detection with temporal encoding," *Computers & Security*, vol. 142, Art. no. 103981, 2025.
32. M. Rahman, A. Hossain, and F. Ahmed, "Explainable transformer framework for cyberattack detection in SDN," *Engineering Applications of Artificial Intelligence*, vol. 136, Art. no. 108912, 2025.
33. X. Zhou, Y. Sun, and L. Wang, "Federated transformer intrusion detection for privacy-preserving SDN security," *IEEE Transactions on Network and Service Management*, vol. 22, no. 1, pp. 145–159, 2025.
34. S. Verma, P. Sharma, and R. Gupta, "Temporal-aware transformer architecture for real-time cyberattack detection in programmable networks," *Journal of Network and Computer Applications*, vol. 240, Art. no. 104102, 2026.
35. A. Hassan, M. Alotaibi, and F. Khan, "Efficient SDN intrusion detection using compressed transformer models and INT8 quantization," *IEEE Access*, vol. 14, pp. 11425–11440, 2026.