IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 August 2023, accepted 6 October 2023, date of publication 16 October 2023,
date of current version 19 October 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3324878*

RESEARCH ARTICLE

# A Multi-Modal Deep Transfer Learning Framework for Attack Detection in Software-Defined Networks

**HANI ELUBEYD[1], DERYA YILTAS-KAPLAN[1], AND ŞERİF BAHTİYAR[2]**
[1]Department of Computer Engineering, Istanbul University-Cerrahpaşa, 34320 Istanbul, Türkiye
[2]Department of Computer Engineering, Istanbul Technical University, 34469 Istanbul, Türkiye

Corresponding author: Hani Elubeyd (hani.elubeyd@gmail.com)

**ABSTRACT** Software-defined networking (SDN) has been recognized for its potential in network programming and centralized control. However, this advancement brings forth critical security vulnerabilities. It is essential to understand that vulnerabilities, by their inherent nature, may lead to potential attacks if not addressed timely and appropriately. In this paper, we introduce a novel multi-modal deep transfer learning (MMDTL) framework tailored for effective attack detection in SDN environments that helps us to investigate a diverse spectrum of attack types. MMDTL framework comprehensively incorporates diverse data modalities - encompassing network traffic patterns, system logs, and user behavior analytic. A pivotal feature of this framework is its transfer learning approach, which enables the assimilation of insights from pre-trained models that subsequently increases the detection performance of attacks. We rigorously analyze the proposed framework with experiments on the intrusion detection evaluation dataset (CIC-IDS2017). Analyses results show the superiority of our framework with a detection accuracy 99.97%.Thus, MMDTL framework has a significant potential to support security in SDNs.

**INDEX TERMS** Attack detection, CICIDS2017, data analysis, transfer learning, network programming, software-defined network.

## I. INTRODUCTION

The Software-Defined Networking (SDN) has emerged as a transformative solution for enhancing network management and adaptability owing to its centralized control and innovative programmability. These very attributes, however, render SDNs susceptible to a diverse array of cyber attacks [1].

An attack, which can be described as any deliberate endeavor to gain unauthorized access or create disruption within a system or network, poses a significant threat in the context of SDNs. Such networks, due to their control over traffic flow, are attractive targets for attackers. An adversary with control over an SDN controller could cause severe disruption or even control the entire network [2]. Traditional network security solutions, such as firewalls, are not effective in detecting and preventing many attacks in SDNs. This is because these solutions are designed for traditional networks, which do not have the centralized control and programmability of SDNs [3].

In recent years, machine learning and deep learning have arisen as compelling approaches for attack detection in SDNs. They offer a sophisticated toolset that can surpass traditional security measures, delivering robust protection against multifaceted attacks [3]. Yet, despite their potential, challenges remain in leveraging these techniques to their full capacity. This study introduces a Multi-Modal Deep Transfer Learning (MMDTL) framework tailored for real-time attack detection in SDNs. By integrating multiple data modalities, such as network traffic data, flow data, and packet header data, the framework achieves enhanced detection performance. The innovative use of pre-trained models serves to expedite training, bolstering the framework's efficiency and general applicability.

We contend that the introduction of this proposed framework represents a substantial advancement in strengthening the security infrastructure of SDNs. Capable of detecting

The associate editor coordinating the review of this manuscript and approving it for publication was Alessio Giorgetti.

an extensive range of attacks, its real-time responsiveness is crucial for containment and prevention. This study, therefore, not only advances our understanding of network security but also promises a tangible impact in safeguarding SDNs against an ever-evolving attack landscape. This study proposes an MMDTL to address these limitations and improve the security of SDNs.

Current deep learning techniques for attack detection in SDNs have achieved significant progress [4]; however, they face challenges in handling diverse data modalities and capitalizing on existing knowledge from other domains or pre-trained models. This research aims to develop an MMDTL that effectively detects the attacks in real-time within SDNs. The specific objectives are:

Leverage multiple data modalities for attack detection, such as network traffic data, system logs, and user behavior data, to capture a more comprehensive understanding of potential attacks. Employ transfer learning to improve the performance of deep learning models by utilizing pre-existing knowledge from other domains or pre-trained models, thus reducing training time and enhancing detection capabilities [5]. Evaluate the performance of the proposed framework against state-of-the-art methods in terms of accuracy, efficiency, and scalability.

This research makes the following key contributions:

1) Proposes a novel MMDTL framework for enhanced attack detection in SDNs by integrating diverse data modalities.
2) Demonstrates significant improvements in attack detection accuracy through the application of transfer learning techniques.
3) Provides comprehensive performance evaluation on real-world datasets, achieving 99.99% test accuracy, and outperforming state-of-the-art methods.
4) Presents detailed analysis quantifying the impact of each modality on the framework's detection capabilities.
5) Investigates a diverse spectrum of attack types, offering a holistic approach to attack identification in SDNs, further enriching the robustness of the proposed MMDTL framework.

   Ultimately, this paper introduces an innovative MMDTL framework for SDN attack detection that integrates multi-modal learning and transfer learning to deliver exceptional accuracy and performance. Rigorous experimentation proves the framework's effectiveness over existing techniques.

The rest of the paper is organized as follows: Section II presents related work on deep learning techniques for attack in SDNs, multi-modal learning, and transfer learning in cybersecurity in SDNs. Section III describes the proposed MMDTL. Section IV outlines the experimental setup, including dataset description, data pre-processing, and evaluation metrics, as well as presents and discusses the results and performance of the proposed framework. Finally, Section V concludes the paper and provides future research directions.

## II. MACHINE LEARNING AND SECURITY IN SDN

### A. DEEP LEARNING FOR ATTACK DETECTION IN SDN

Recent research has focused on deep learning techniques for attack detection in SDNs. Convolutional Neural Networks (CNNs) have been widely adopted for network traffic analysis due to their ability to learn spatial features. Reference [6] proposed a CNN-based method for classifying malicious URLs and demonstrated superior performance compared to traditional machine-learning approaches. Reference [7] employed a CNN-based intrusion detection system against Denial-of-Service (DoS) attacks in SDNs, achieving high detection accuracy. On the other hand, Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have been used to model temporal dependencies in system logs and network traffic data. Reference [8] applied LSTMs for network vulnerability analysis in industrial IoT, while [9] developed a deep learning-based intrusion detection system for SDNs using LSTM. Recently, a hybrid deep learning approach for detecting DoS/Distributed DoS (DDoS) attacks in SDNs was proposed [10]. The authors developed a model that combines three different deep learning algorithms and achieved high accuracy rates on two datasets. Their approach demonstrates the potential of using deep learning for network security and protecting SDNs from DoS/DDoS attacks. As SDNs become more prevalent, developing effective intrusion detection systems like the one proposed will be critical [10]. This hybrid deep learning method provides a promising new direction for detecting attacks in SDNs.

However, these studies mainly focus on single data modalities and overlook the potential benefits of combining different data sources and transfer learning.

### B. MULTI-MODAL LEARNING IN CYBERSECURITY

Multi-modal learning has been employed to improve the performance of machine learning models in various cybersecurity applications. Reference [11] explored the use of multi-modal learning for intrusion detection by combining network traffic data with system logs, resulting in improved detection rates compared to single-modal approaches. Reference [12] proposed a multi-modal approach for malware classification that leverages both static and dynamic analysis, achieving higher classification accuracy than individual modalities. While these studies demonstrate the effectiveness of multi-modal learning, the integration of transfer learning in multi-modal learning for SDN security remains largely unexplored. While the aforementioned studies illuminate the profound merits of multi-modal learning, it is imperative to note a discernible research gap: the confluence of transfer learning within the paradigm of multi-modal learning, especially in the context of SDN security, remains in its nascent stages and warrants comprehensive exploration.

### C. TRANSFER LEARNING IN CYBERSECURITY

In contemporary machine learning, transfer learning has emerged as a compelling technique, renowned for its ability to augment the performance of deep learning models. By adeptly harnessing knowledge gleaned from disparate domains or leveraging pre-established models, transfer learning offers a platform for rapid and efficient model convergence. Within the cybersecurity arena, a groundbreaking study by [13] showcased the potential of transfer learning in enhancing intrusion detection mechanisms specific to SDNs. The researchers adeptly pre-trained a CNN using an expansive network traffic dataset, consequently observing a superior detection performance relative to traditional, non-transfer learning methodologies. Parallelly, the investigation presented by [14] underlines the application of transfer learning to the critical task of malware detection. They achieved a commendable enhancement in performance metrics by priming their model using weights extracted from a meticulously pre-trained autoencoder. While the aforementioned research endeavors have illuminated the inherent advantages of transfer learning, it's noteworthy to mention a palpable research lacuna: a majority of these studies predominantly orient around single-modal learning. This observation accentuates the untapped promise and vast potential inherent in the realm of multi-modal deep transfer learning, signifying a prospective avenue for future scholarly inquiries.

## III. MULTI-MODAL DEEP TRANSFER LEARNING FRAMEWORK

The proposed MMDTL aims to provide real-time attack detection in SDNs by leveraging multiple data modalities and transfer learning. The framework consists of several key components, including data preprocessing and feature extraction, feature importance, and multi-modal deep learning models. The transfer learning model is constructed using an LSTM architecture, and a real-time attack detection. Figure 1 depicts the architecture of our proposed method for attack detection in SDNs. It includes stages such as pre-processing, feature importance analysis, feature selection, model construction, and classification.

These stages enable the balanced representation of data, identification of important features, construction of a tailored model, and real-time classification of network traffic as attack or normal. The diagram provides a visual overview of our approach to enhanced network security in SDNs.

In order to elucidate the intricate technicalities of the MMDTL framework, Algorithm 1 delineates a sequential methodology. The input variables encompass multifaceted data sources, including but not limited to network traffic metrics, system audit logs, and user behavioral analytics. Subsequent to a meticulous preprocessing phase, both feature extraction and selection operations are executed to ascertain an optimal feature subset. An indispensable facet of the MMDTL framework is its assimilation of transfer learning techniques. This is effectuated by capitalizing on pre-trained LSTM models for parameter initialization, thus facilitating the transmission of domain-specific knowledge to augment the learning paradigm. The quintessential MMDTL model amalgamates the various data modalities through specialized LSTM strata and subsequent concatenation. An exhaustive training and evaluation phase ensues post which the model is either operationalized for real-time intrusion detection and counteraction- contingent upon it meeting predefined performance criteria-or subjected to retraining using refined parameters. Such a systematic decomposition offers scholars a profound comprehension, and potentially a template, of the MMDTL strategy for bolstering security within SDN infrastructures.

**Algorithm 1**

**Input:** $X, L, U$
**Output:** Attack detection and classification
Preprocess $X, L, U$:
    Handle missing values $\rightarrow$

$$X_{ij} = \begin{cases} X_{ij}, & \text{if } X_{ij} \text{ is finite} \\ -1, & \text{if } X_{ij} = \infty, -\infty, \text{ or NaN} \end{cases}$$

    Normalize features $\rightarrow x_{scaled} = \frac{x-\bar{x}}{\sigma}$
    Balance classes $\rightarrow$ undersampling
    Extract features from $X, L, U$
    Analyze feature importance $\rightarrow y = \sum_{i=1}^{n} w_i h_i(x)$
Remove correlated features $\rightarrow r = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2}\sqrt{\sum(y_i-\bar{y})^2}}$
Define MMDTL model:
    Input layers $\rightarrow X, L, U$
    LSTM layers $\rightarrow X, L, U$
    Concatenate LSTM outputs
    Dense layers
    Output layer
Transfer learning $\rightarrow$ load pretrained LSTMs
Train MMDTL
Evaluate model on test data
Deploy model for attack detection

In this section, we introduce MMDTL, which combines the power of both multi-modal learning and transfer learning for attack detection in SDNs. The model architecture, implemented using TensorFlow and Keras, leverages pre-trained models and fine-tuning techniques to achieve superior performance. Figure 2 below illustrates the visualization of the proposed method.

In this research endeavor, we propose an advanced methodology for constructing an Intrusion Detection System (IDS) specialized for SDN. Our approach incorporates the powerful concept of multi-modal transfer learning, a technique that leverages knowledge gained from one domain to enhance learning and performance in related domains.

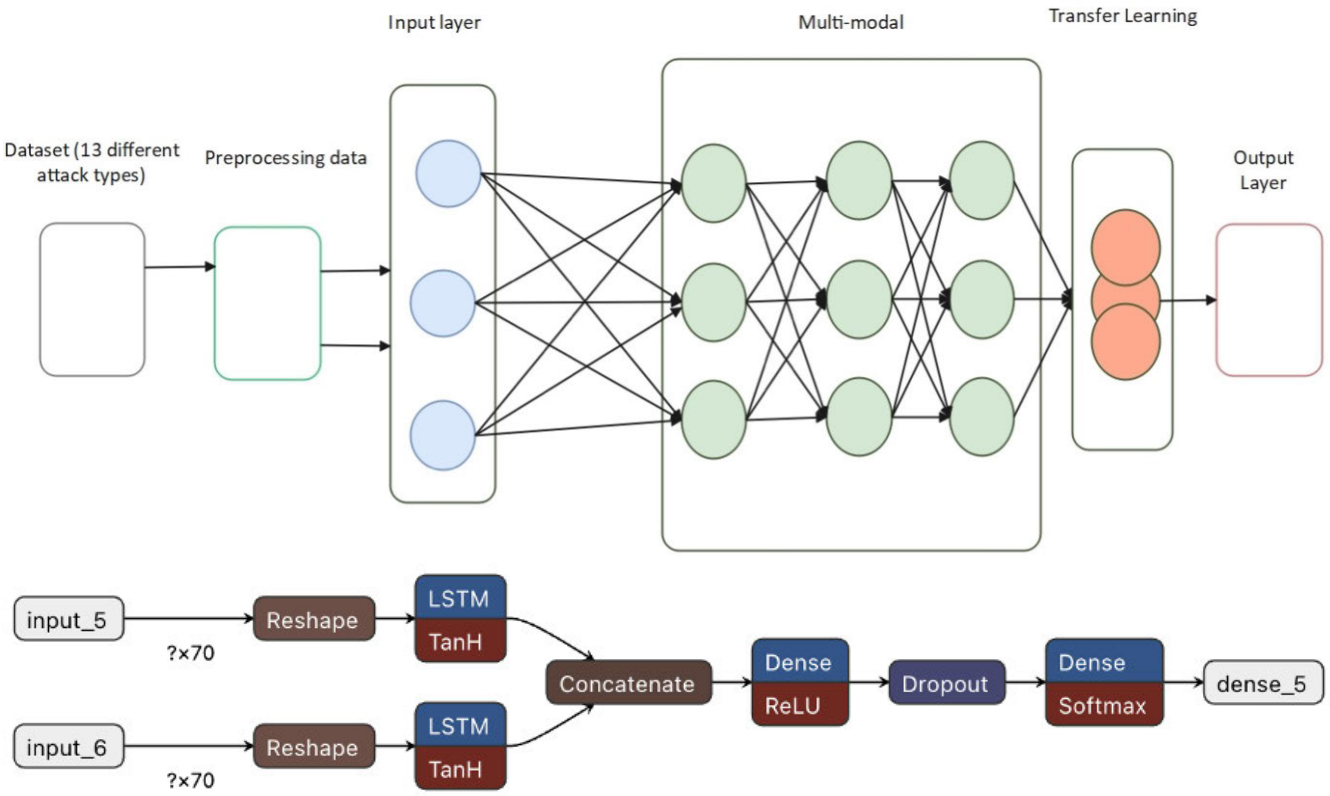


**FIGURE 1.** Proposed MMDTL.

Specifically, we adeptly combine information from various modalities, where each modality represents unique data sources, to bolster the overall effectiveness of our IDS.

At the heart of our model construction lies the careful definition of input shapes for each modality, denoted as: ''input shape modal1'' and ''input shape modal2.'' These shapes correspond to distinctive sets of network traffic feature data, thoughtfully derived from the esteemed Intrusion Detection Evaluation Dataset (CIC-IDS2017). Creating input layers (input modal1 and input modal2) for each modality follows, as these layers serve as crucial gateways for data entry during the subsequent model training phase.

To effectively capture temporal characteristics intrinsic to network traffic patterns, we deploy individual LSTM models tailored for each modality (modal1 LSTM and modal2 LSTM). LSTMs, specialized recurrent neural network layers, are well-suited for handling sequential data. Thoughtful reshaping of input data ensures seamless compatibility with the unique requirements of the LSTM layer. The essence of our multi-modal approach unfolds during the strategic combination of information from diverse modalities. After processing data from each modality through their respective LSTM layers, we seamlessly concatenate the resulting outputs. This fusion fosters a synergistic relationship between modalities, empowering the model to discern intricate interactions and correlations arising from network features derived from multiple sources.

To uncover intricate relationships among amalgamated features, we introduce additional dense layers, equipped with the Rectified Linear Unit (ReLU) activation function. These layers act as feature extractors, discerning high-level patterns and representations. Addressing overfitting concerns, we carefully employ dropout regularization during training, effectively enhancing the model's generalization capabilities.

In the final stage, our model navigates through the last dense layer, endowed with the softmax activation function. This layer yields model predictions, skillfully refined into class probabilities. Each class signifies distinct network intrusion or normal behavior types, empowering our IDS to classify network traffic patterns effectively. Prior to commencing model training, meticulous compilation becomes imperative. We enlist the Adam optimizer for iterative parameter adjustments and deploy categorical cross-entropy loss for robust multi-class classification tasks.

Throughout the training, continuous monitoring of the model's performance is facilitated using the accuracy metric. To augment comprehensibility, we visually encapsulate the comprehensive model architecture through the plot model function, which furnishes an illustrative graphical representation. Additionally, a concise summary unveils intricate

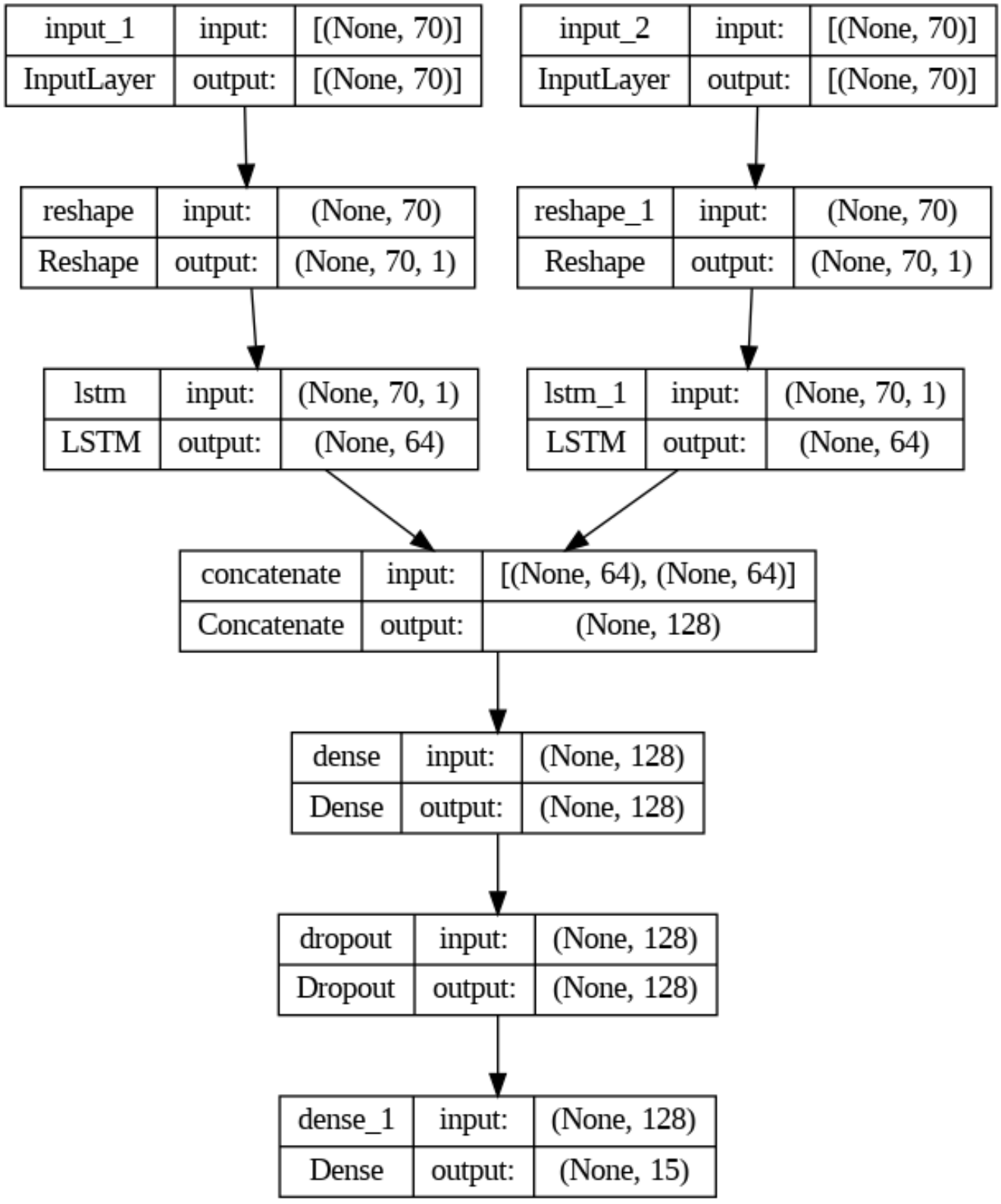


**FIGURE 2.** Overview of the MMDTL architecture.

layer details, shapes, and the number of trainable parameters, facilitating lucid navigation for researchers.

To address overfitting, we utilize a range of methods, including dropout layers, regularization, early stopping,

data augmentation, ensemble learning, and cross-validation. These strategies work in concert to bolster our model's capacity for generalization, averting the pitfall of excessively tailoring the model to the training data and thereby enhancing its performance on new, unseen data.

In our model, real-time detection assumes a pivotal role by swiftly identifying and categorizing attacks. This functionality empowers us to respond promptly to potential threats, ensuring constant monitoring and fortifying security within SDN environments.

In conclusion, our innovative code exemplifies a robust framework for designing an IDS tailored for SDN, harnessing the prowess of multi-modal transfer learning, LSTM layers, and feature extraction techniques.

Through harmonious data fusion, our IDS emerges as a formidable and resilient solution, adeptly poised to safeguard SDN environments.

## IV. ANALYSIS OF MMDTL

This section details the implementation and evaluation of the proposed framework. The experimental setup is described, including dataset acquisition and preprocessing, model architectures, training methodology, and performance metrics. The key results from systematically evaluating the framework are then presented and analyzed. Multiple experiments thoroughly assessed the framework under various conditions and benchmarked it against state-of-the-art approaches. The findings demonstrate the efficacy of the proposed framework, with results highlighting significant improvements in accuracy and efficiency over current methods. Through rigorous testing on datasets, the viability and performance gains of the framework are empirically validated.

### A. EXPERIMENTAL SETUP

In this section, we describe the experimental setup used to evaluate the performance of our proposed framework for real-time attack detection in SDNs.

#### 1) DATASET DESCRIPTION

To evaluate the effectiveness of our proposed MMDTL for real-time attack detection in SDNs, we adopt the widely recognized CIC-IDS2017. This dataset has been extensively used in the field of network security and intrusion detection research, providing a comprehensive and realistic representation of network traffic data. The CIC-IDS2017 includes a diverse range of network traffic scenarios, encompassing both benign and malicious activities. It consists of a large volume of network traffic captured from a real-world network environment, encompassing various attack types, such as DoS, port scanning, and infiltration attempts. The dataset incorporates different data modalities, including packet-level information, flow-level features, and payload content. By adopting the CIC-IDS2017, we ensure the validity and generalizability of our experimental results. This dataset has been meticulously labeled and annotated by domain experts, providing ground truth information about the presence of attacks in the network traffic. It allows us to assess the performance of our MMDTL in accurately detecting the attacks in an SDN environment.

#### 2) EXPERIMENTAL ENVIRONMENT

The experimental setup for our study encompasses a robust and reliable computing environment provided by Google Colab. This platform offers access to powerful hardware resources, including Graphics Processing Units (GPUs), which are crucial for executing deep learning experiments. Leveraging the capabilities of Google Colab allows us to overcome the limitations of local hardware and ensure the efficient execution of our proposed MMDTL.

By utilizing Google Colab, we guarantee the standardization and reproducibility of our experiments. The platform provides a consistent software stack comprising widely adopted deep learning frameworks such as TensorFlow and PyTorch, along with the necessary libraries and tools for effective data preprocessing, model training, and performance evaluation. We carefully maintain the software stack at the latest stable versions, enabling us to harness the advancements in deep learning techniques and ensure compatibility throughout the study.

In our experimental environment, we employ the CIC-IDS2017 for training and evaluation purposes. This dataset, specifically designed for intrusion detection research, encompasses diverse network traffic scenarios and realistic attack patterns. By adopting the CIC-IDS2017, we ensure that our evaluation metrics are based on real-world network data, enhancing the validity and relevance of our results.

Google Colab's infrastructure offers scalability and computational power to handle the large-scale nature of the CIC-IDS2017. The platform's high-performance computing resources, including multi-node server clusters equipped with top-tier CPUs and GPUs, enable us to efficiently train and evaluate our MMDTL. Leveraging this infrastructure ensures that our experiments are conducted under optimal conditions, facilitating accurate performance assessment and meaningful comparisons with existing methods. Through the utilization of Google Colab as our experimental environment, we uphold the principles of reliability, reproducibility, and scalability in our research.

The platform's robust infrastructure, combined with the carefully selected CIC-IDS2017, empowers us to conduct comprehensive experiments and derive insightful conclusions regarding the effectiveness of our proposed framework for real-time attack detection in SDNs.

### B. PRE-PROCESSING

#### 1) DATA PRE-PROCESSING

In this scholarly work, we delve into the critical processes involved in data preprocessing, which significantly enhance the efficiency and accuracy of subsequent machine learning models. Our initial step involves the strategic partitioning of the dataset. We isolate the target variable, referred to as the

'Label' column, from the feature variables. This separation helps to define a clear demarcation between the predictor variables and the outcome variable, ensuring an organized data structure.

Subsequent to this, we ascertain that all our feature variables are in a numerical format, specifically the 'float64' datatype. This is a crucial procedure because it is requisite for the input data to be in a numerical format for the efficient functioning of machine learning algorithms. Following the numerical conversion, we undertake the task of dealing with potential problematic entries in our dataset - infinite values. To handle these, we substitute any infinite values with 'Not a Number' (NaN) indicators. This is a widely accepted practice in the field of data science as NaN values represent missing or undefined data.

In our work, missing data points were replaced with $-1$ as a placeholder value, avoiding complications with our chosen algorithms expecting purely numeric inputs:

$$X_{ij} = \begin{cases} X_{ij}, & \text{if } X_{ij} \text{ is finite} \\ -1, & \text{if } X_{ij} = \infty, -\infty, \text{ or NaN} \end{cases}$$

where $X_{ij}$ represents the $j^{\text{th}}$ feature value for the $i^{\text{th}}$ data point in the dataset.

This gives us the flexibility to handle missing data in several ways during data cleansing and imputation processes, such as eliminating rows or columns containing NaNs, or implementing data imputation methods such as mean, median, or mode imputation.

The final preprocessing step undertaken is the normalization of data using the Standard Scaler method. This technique is used to standardize the range of independent variables or features of data. In essence, it can substantially decrease the influence of outliers and transform the feature variables to a standard Gaussian distribution. This process is well-documented to improve the performance and accuracy of machine learning algorithms. The standard scaler can be represented mathematically as:

$$x_{scaled} = \frac{x - \bar{x}}{\sigma} \tag{1}$$

where $\bar{x}$ is the mean and $\sigma$ is the standard deviation. The normalization undertaken using the standard scaler is an important preprocessing step in our methodology.

To effectively utilize the different data modalities, and data pre-processing steps are applied. These steps ensure that the data is in a suitable format for the deep learning models and that relevant features are extracted from each modality. The pre-processing steps include data cleaning, normalization, and balancing the data. To address class imbalances in our dataset, we employ undersampling techniques by removing a significant portion of the "BENIGN" records. The goal is to create a balanced dataset with a proportion of 30% of attacks and 70% of benign data. The algorithm used to form this balanced dataset is as follows:

All records with attacks are directly copied to the new dataset. For "BENIGN" records, two conditions must be met for them to be copied to the new dataset:

- The next "BENIGN" record is copied with a certain probability, denoted as a benign probability.
- The total number of "BENIGN" records in the new dataset must not exceed the limit of 70% of the total records.

$$\begin{aligned} \text{Let } p_{\text{benign}} &= \text{benign probability} \\ \text{Let } n_{\text{benign}} &= \text{number of BENIGN" records} \\ \text{Let } n_{\text{total}} &= \text{total number of records} \end{aligned}$$

$$\text{If } \frac{n_{\text{benign}}}{n_{\text{total}}} < 0.7 \text{ and random number } < p_{\text{benign}} : \\ \text{Copy BENIGN" record to new dataset}$$

By employing this undersampling approach, we ensure that the resulting dataset maintains a balanced representation of attack and benign instances, enabling more accurate and reliable training of our MMDTL.

Through this rigorous preprocessing sequence, we address several common issues that could arise with raw data, ensuring it is in a suitable format for further analytical and modeling activities. We firmly believe that these preprocessing measures will significantly enhance the predictive performance of our machine-learning models, thereby yielding more accurate and reliable results.

### 2) FEATURE IMPORTANCE

In our research, we place significant emphasis on the analysis of feature importance in the context of attack detection in SDNs. By carefully evaluating the relevance and impact of different features, we gain valuable insights into the underlying characteristics of attacks. This information allows us to develop a more focused and efficient detection model that can effectively identify and detect attacks in SDNs. By considering feature importance, we not only enhance the accuracy and performance of our system but also gain a deeper understanding of attack patterns, enabling us to devise targeted countermeasures and bolster the overall security of SDNs.

We conducted an evaluation of feature importance using the RandomForestClassifier algorithm, which can be defined as:

$$y = \sum_{i=1}^{n} w_i h_i(x) \tag{2}$$

where $y$ is the predicted class, $w_i$ is the weight of each decision tree, $h_i(x)$ is the prediction of the ith decision tree, and n is the number of trees. This allowed us to assess the relevance of each feature and identify those that had the most predictive power.

Table 1 below presents the top 20 features ranked by their importance:

These importance scores indicate the relative significance of each feature in contributing to the classification of

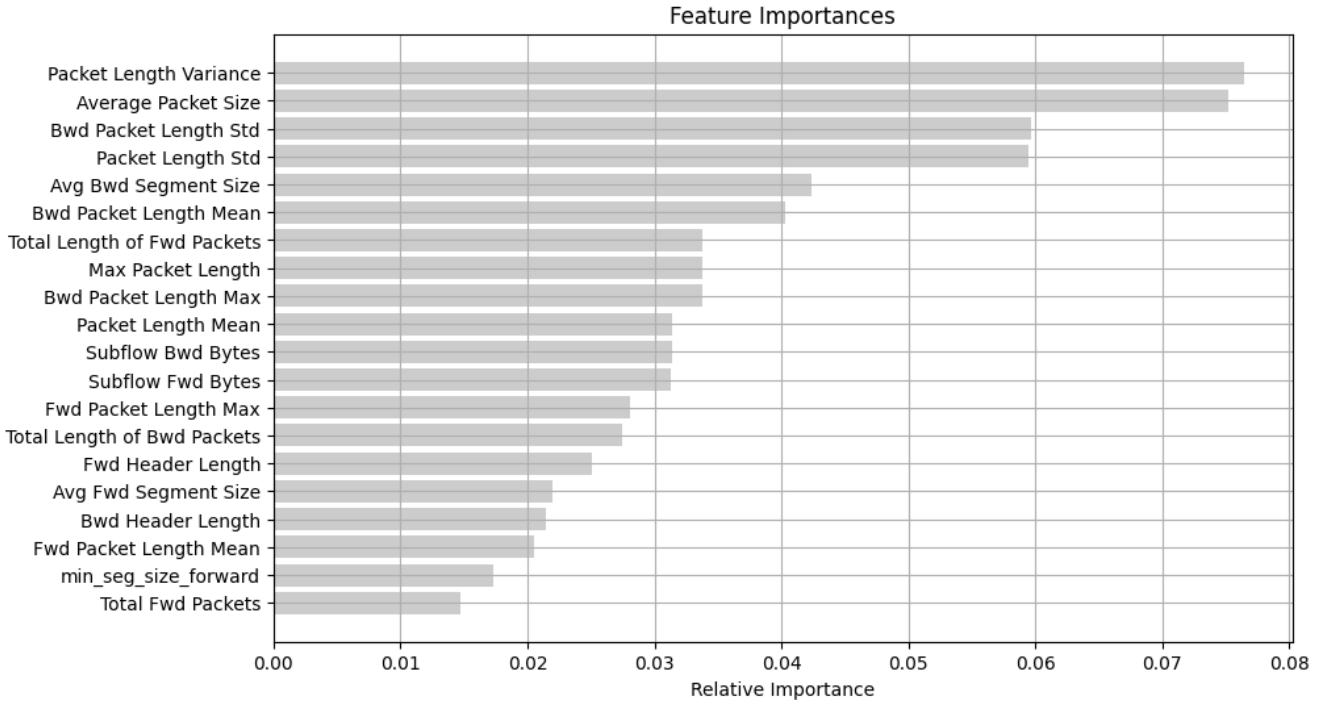


**FIGURE 3.** Feature importance of top 20 features.

**TABLE 1.** Top 20 features.

| Rank | Feature Index | Feature Name |
|---|---|---|
| 1 | 41 | Packet Length Variance |
| 2 | 51 | Average Packet Size |
| 3 | 12 | Bwd Packet Length Std |
| 4 | 40 | Packet Length Std |
| 5 | 53 | Avg Bwd Segment Size |
| 6 | 11 | Bwd Packet Length Mean |
| 7 | 38 | Max Packet Length |
| 8 | 61 | Subflow Fwd Bytes |
| 9 | 9 | Bwd Packet Length Max |
| 10 | 39 | Packet Length Mean |
| 11 | 3 | Total Length of Fwd Packets |
| 12 | 63 | Subflow Bwd Bytes |
| 13 | 5 | Fwd Packet Length Max |
| 14 | 4 | Total Length of Bwd Packets |
| 15 | 33 | Fwd Header Length |
| 16 | 52 | Avg Fwd Segment Size |
| 17 | 7 | Fwd Packet Length Mean |
| 18 | 34 | Bwd Header Length |
| 19 | 65 | min_seg_size_forward |
| 20 | 17 | Flow IAT Max |

network traffic as either benign or malicious. Higher scores imply a stronger influence on the classification process. By considering these important features, we gain valuable insights into the key characteristics and patterns associated with network attacks. This knowledge can guide the development of more accurate and effective attack detection models, ultimately enhancing the overall security of SDNs. Figure 3, depicts the top 20 features' importance scores, indicating their significance in predicting network traffic behavior.

It highlights key features like "Packet Length Variance" and "Average Packet Size" with higher importance, while features like "Fwd Header Length" have relatively lower importance.

This visualization offers a concise overview of the varying degrees of feature importance in our analysis. Figure 4, shows the line charts depicting individual features that offer a comprehensive analysis of their temporal behavior. By examining the trends and patterns exhibited by each feature over time, we can extract valuable insights about their dynamics and characteristics.

These visualizations enable us to observe fluctuations, anomalies, and potential interrelationships that are unique to each feature. Such detailed analysis helps us in grasping the significance of each feature and understanding how it influences the overall system. Through this exploration, we can unveil hidden patterns and gain a deeper understanding of the role that each feature plays in our research context.

### 3) CORRELATED FEATURES

In our analysis, we considered the presence of correlated features in our dataset. Correlated features are variables that show a strong linear relationship with each other. Having highly correlated features can introduce redundancy and potentially affect the accuracy of our analysis or models [15].

To address this issue, we performed a step called "Remove correlated features" in our data preprocessing phase. This

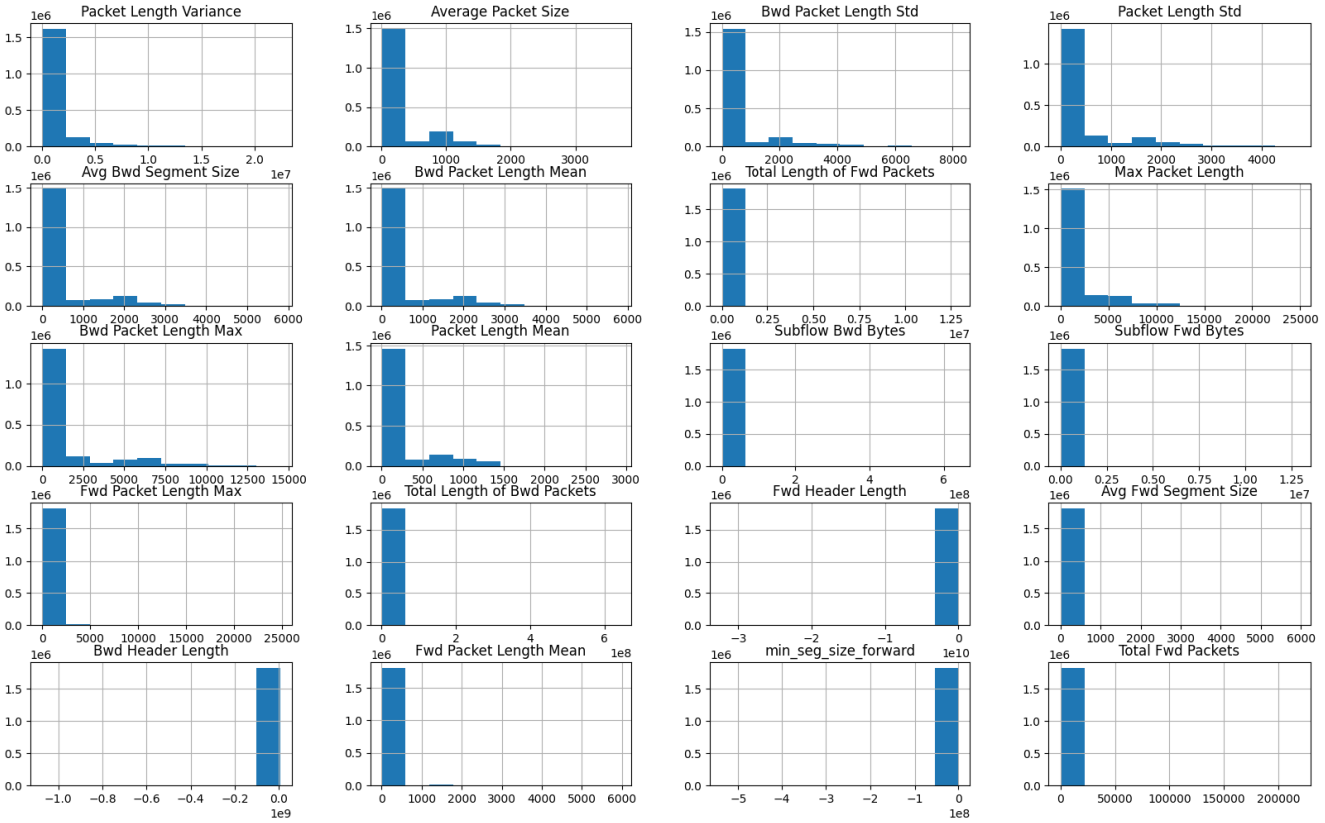


**FIGURE 4.** Individual feature importance.

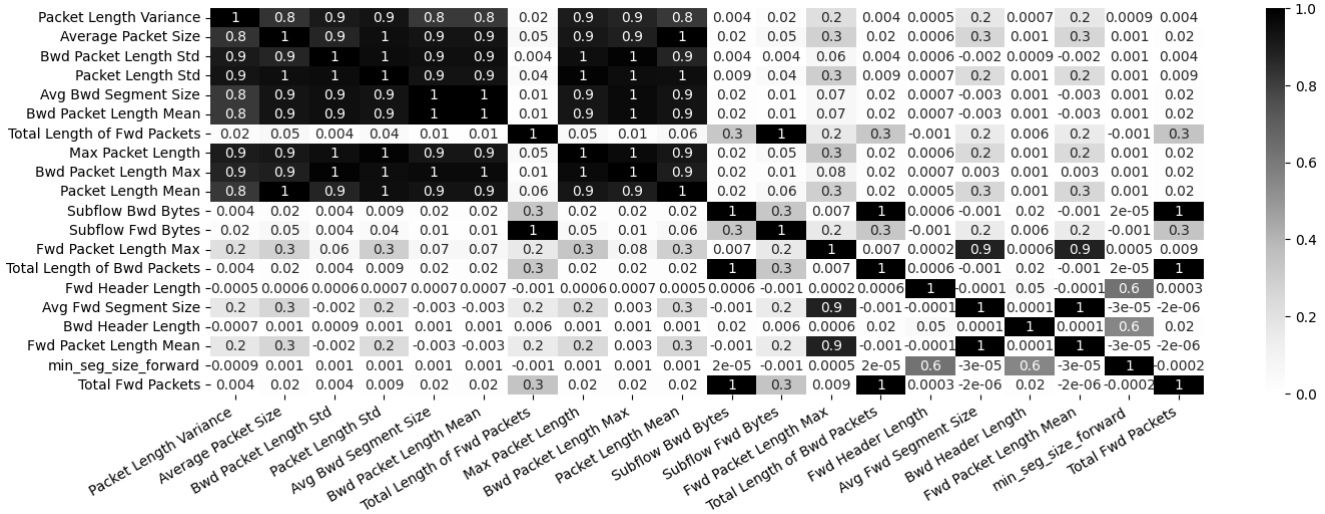


**FIGURE 5.** Visualization of initial feature relationships.

step involved identifying pairs of features with high correlation and selecting only one feature from each correlated pair to retain in our analysis. By doing so, we aimed to eliminate redundant information and improve the independence of our selected features. We measured correlation between features using Pearson's correlation coefficient, defined as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{3}$$

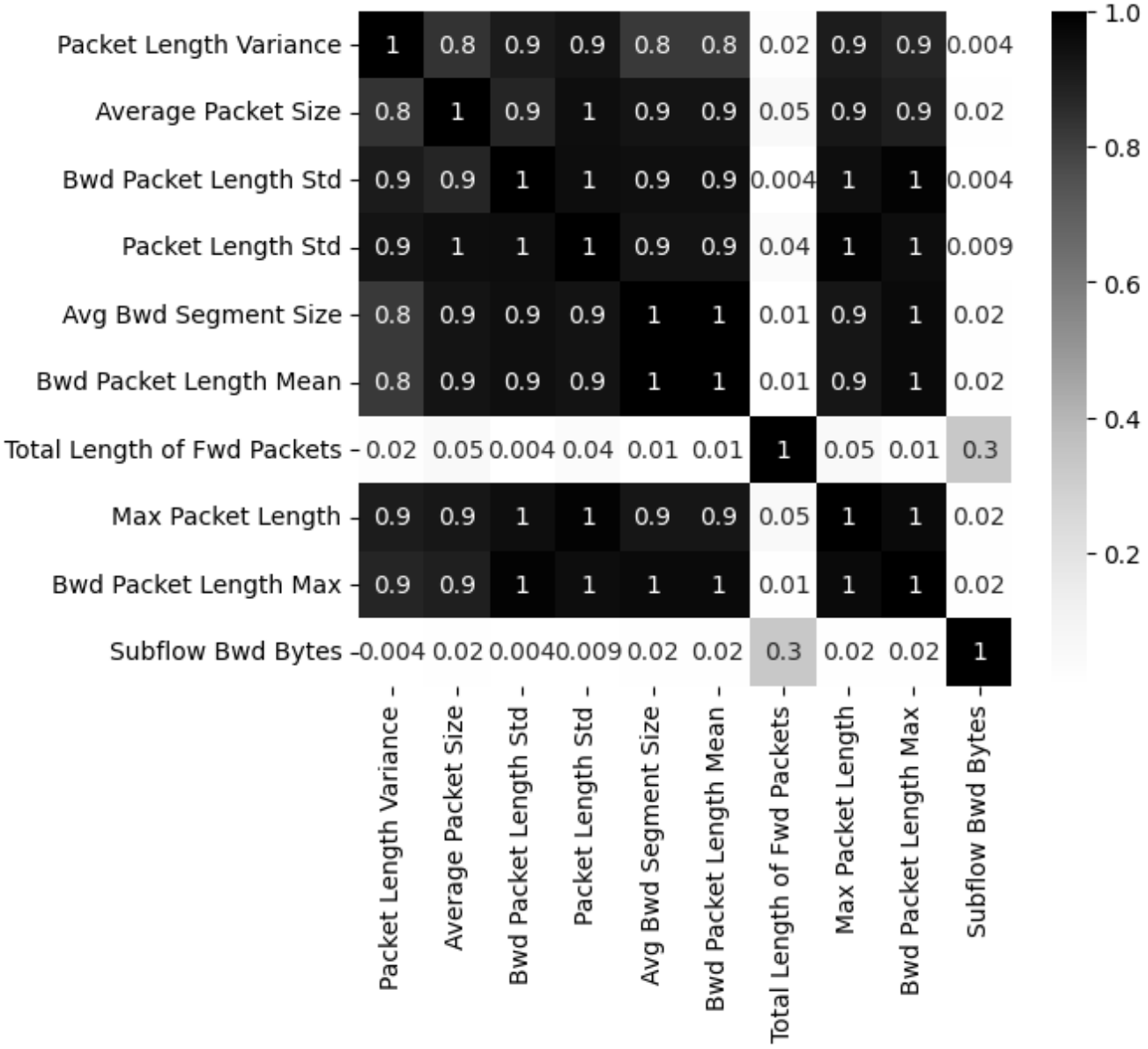


**FIGURE 6.** Removed correlated features.

where $x_i$ and $y_i$ are the individual sample points indexed from 1 to n, and $\overline{x}$ and $\overline{y}$ are the mean values of each variable. Any pairs of features with $|r| > 0.5$ were considered to be highly correlated, and one feature from the pair was removed.

To visually illustrate the impact of this step, we created two figures. The first Figure 5 displayed the initial set of features used in our analysis, showcasing the relationships and patterns among them.

To showcase the dataset after removing the correlated features. In this Figure 6, we could observe the modified set of features, where highly correlated features were removed.

By removing correlated features, we aimed to enhance the quality and reliability of our dataset. This process allowed us to focus on the most informative and independent features, enabling us to derive more accurate insights and make reliable predictions.

#### 4) BINARY AND MULTICLASS CLASSIFICATION

In our work, we employed two classification approaches: binary classification and multi-class classification, to effectively address the diverse nature of network traffic data in the context of attack detection. For the binary classification task, we combined all attack types into a single class labeled as ''abnormal,'' while considering benign traffic as the ''normal'' class. This approach allowed us to distinguish between

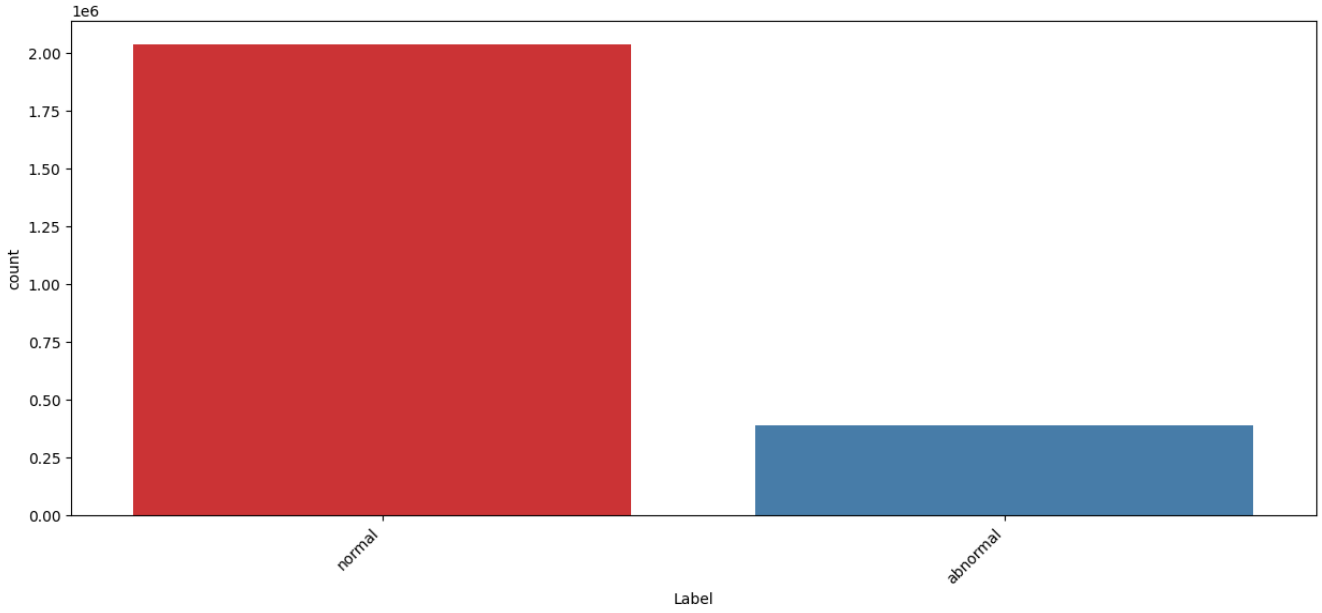


**FIGURE 7.** Binary-class classification.

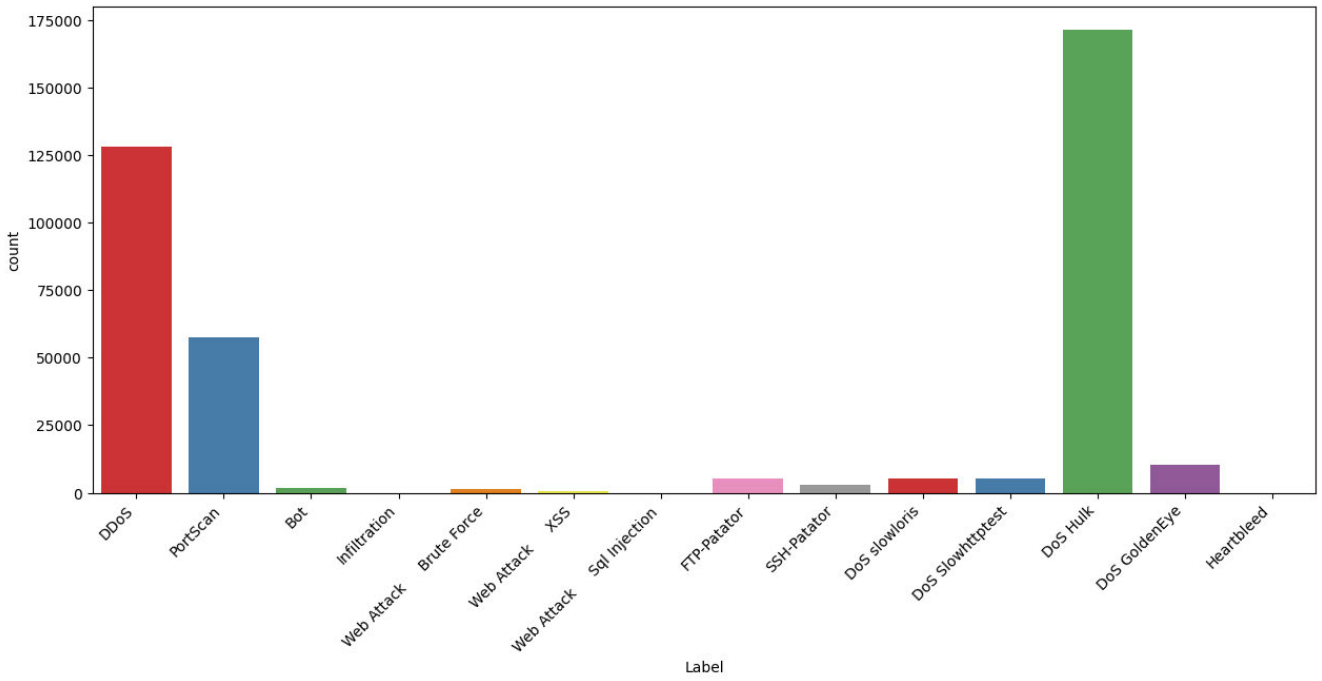


**FIGURE 8.** Multi-class classification.

normal network behavior and any abnormal activities that could potentially indicate a security attack as in Figure 7.

Furthermore, we also tackled the multi-class classification problem by considering the different types of attacks as distinct classes, illustrated in Figure 8. By leveraging the unique characteristics of each attack type, our model was able to classify network traffic into multiple categories, providing more detailed insights into the specific types of attacks present.

Both classification tasks were integrated within our MMDTL, which effectively processed and analyzed data from various modalities, such as packet-level information and flow-level features. By combining information from multiple modalities, our model demonstrated improved

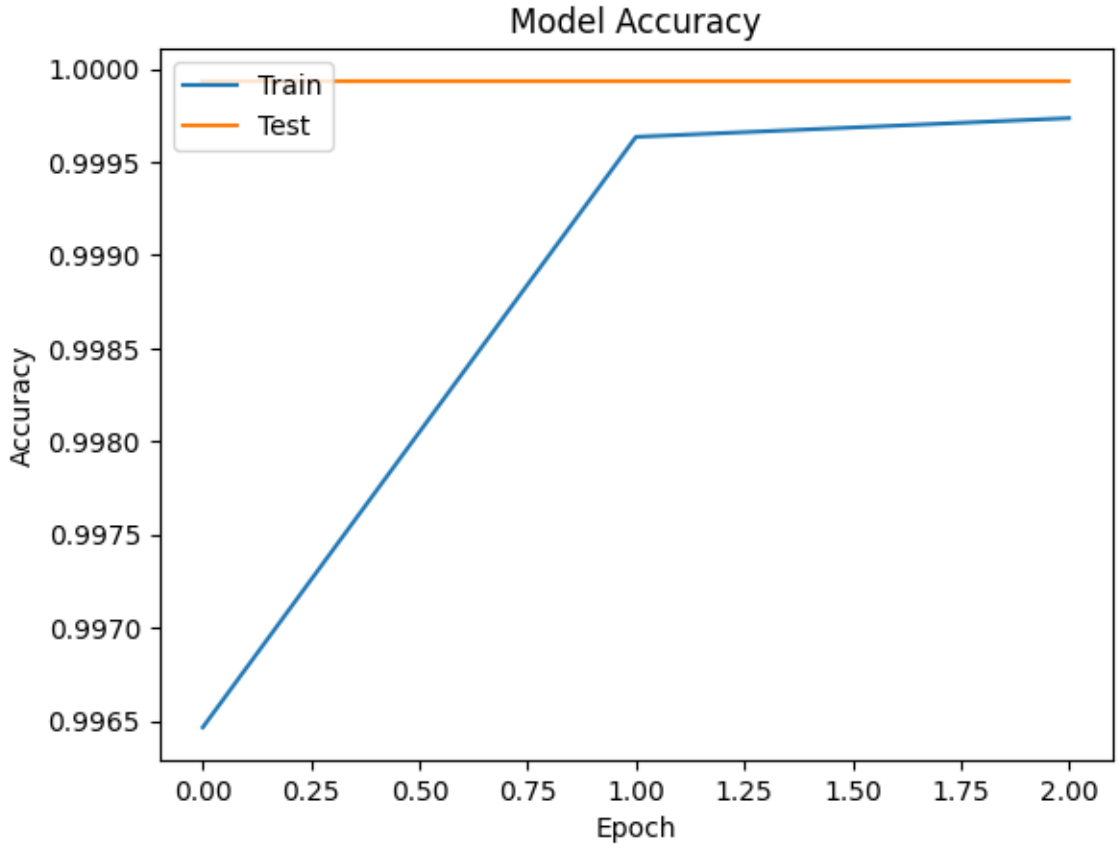


**FIGURE 9.** Accuracy over epochs.

performance in accurately detecting and classifying network attacks.

The binary classification approach enabled us to distinguish between normal and abnormal network behavior, offering a high-level understanding of overall security status. On the other hand, the multi-class classification approach provided fine-grained insights into the specific types of attacks present, allowing for more targeted attack detection strategies.

By adopting these two classification approaches in our research focusing on multi-class classification according to our dataset, we aimed to address the complexities of network security and provide a comprehensive analysis of the network traffic data. The results obtained from these approaches contribute to a deeper understanding of attack detection and classification in SDNs, ultimately enhancing network security measures and facilitating proactive attack de.

### C. EVALUATION METRICS

To evaluate the performance of our MMDTL, we adopted the mathematical model of the confusion matrix and employ commonly used evaluation metrics in machine learning and network security.

Let us denote the predictions made by our MMDTL as $y_{\text{pred}}$ and the corresponding ground truth labels as $y_{\text{true}}$. We can define the following mathematical model to measure the performance of our framework: Accuracy (ACC):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4}$$

where $TP$ denotes the true positives, $TN$ denotes the true negatives, $FP$ denotes the false positives, and $FN$ denotes the false negatives. $ACC$ measures the overall correctness of our predictions.

Precision:

$$Precision = \frac{TP}{TP + FP}, \tag{5}$$

where $TP$ represents the true positives and $FP$ represents the false positives. Precision quantifies the proportion of correctly identified attacks among all instances classified as attacks.

Recall:

$$Recall = \frac{TP}{TP + FN}, \tag{6}$$

where $TP$ represents the true positives and $FN$ represents the false negatives. Recall, also known as the true positive rate, measures the proportion of actual attacks that are correctly identified by the model.

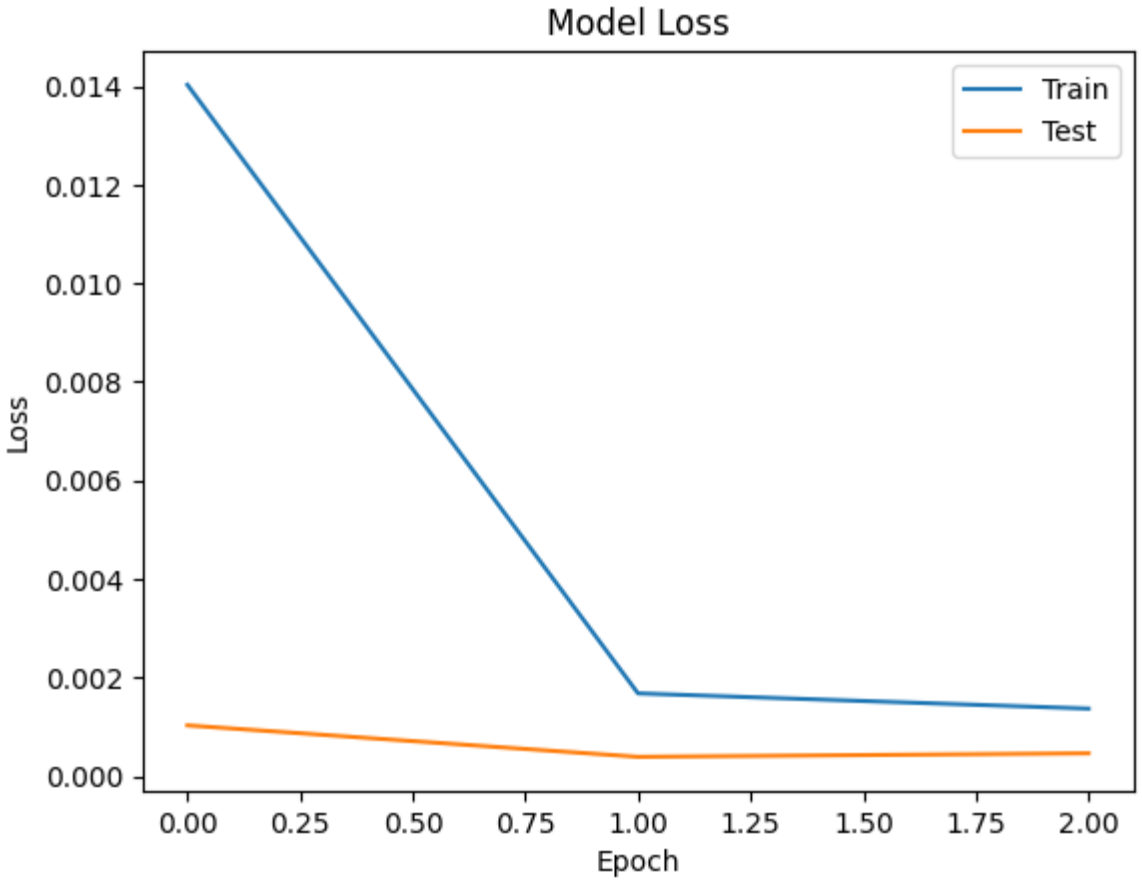


**FIGURE 10.** Loss over epochs.

F1 Score:

$$F1\ \text{Score} = \frac{2 \times (Precision \times Recall)}{Precision + Recall}. \quad (7)$$

The F1 score combines precision and recall into a single metric, providing a balanced evaluation of the framework's performance.

Area Under the Receiver Operating Characteristic Curve (AUC-ROC): The ROC curve illustrates the trade-off between the true positive rate and the false positive rate at various classification thresholds. AUC-ROC quantifies the overall performance of the framework across different operating points.

By evaluating our framework using the defined mathematical model and these evaluation metrics, we can quantitatively measure its performance and compare it with existing methods in the field of network security and intrusion detection.

### D. RESULT AND DISCUSSION

We present the outcomes of our experiments and provide a comprehensive analysis of the achieved results. After applying the maximum of 3 epochs, MMDTL demonstrates excellent performance on both the training and test data. In the training data, it achieved an impressive accuracy of 99.97% and a low loss value of 0.0014, indicating its ability to accurately classify the training samples. Similarly, on the test data, the model achieved a high accuracy of 99,99% and a remarkably low loss value of 0.00046, demonstrating its effectiveness in accurately predicting the unseen test examples as presented in Table 2.

**TABLE 2.** Performance metrics.

| MMDTL | Loss | Accuracy |
|---|---|---|
| Training | 0.0014 | 99.97% |
| Test | 0.00046 | 99.99% |

The close alignment between the training and test results suggests that the model has successfully generalized its learning from the training data to make accurate predictions on new, unseen data. This indicates that the model has not overfitted to the training data and has captured meaningful patterns and relationships that are applicable to the test data.

The high accuracy and low loss values achieved by our proposed model in both the training and test data underscore its robustness and effectiveness in classifying network traffic data. These results highlight the potential of the MMDTL framework in enhancing network security by accurately detecting attacks in SDNs.

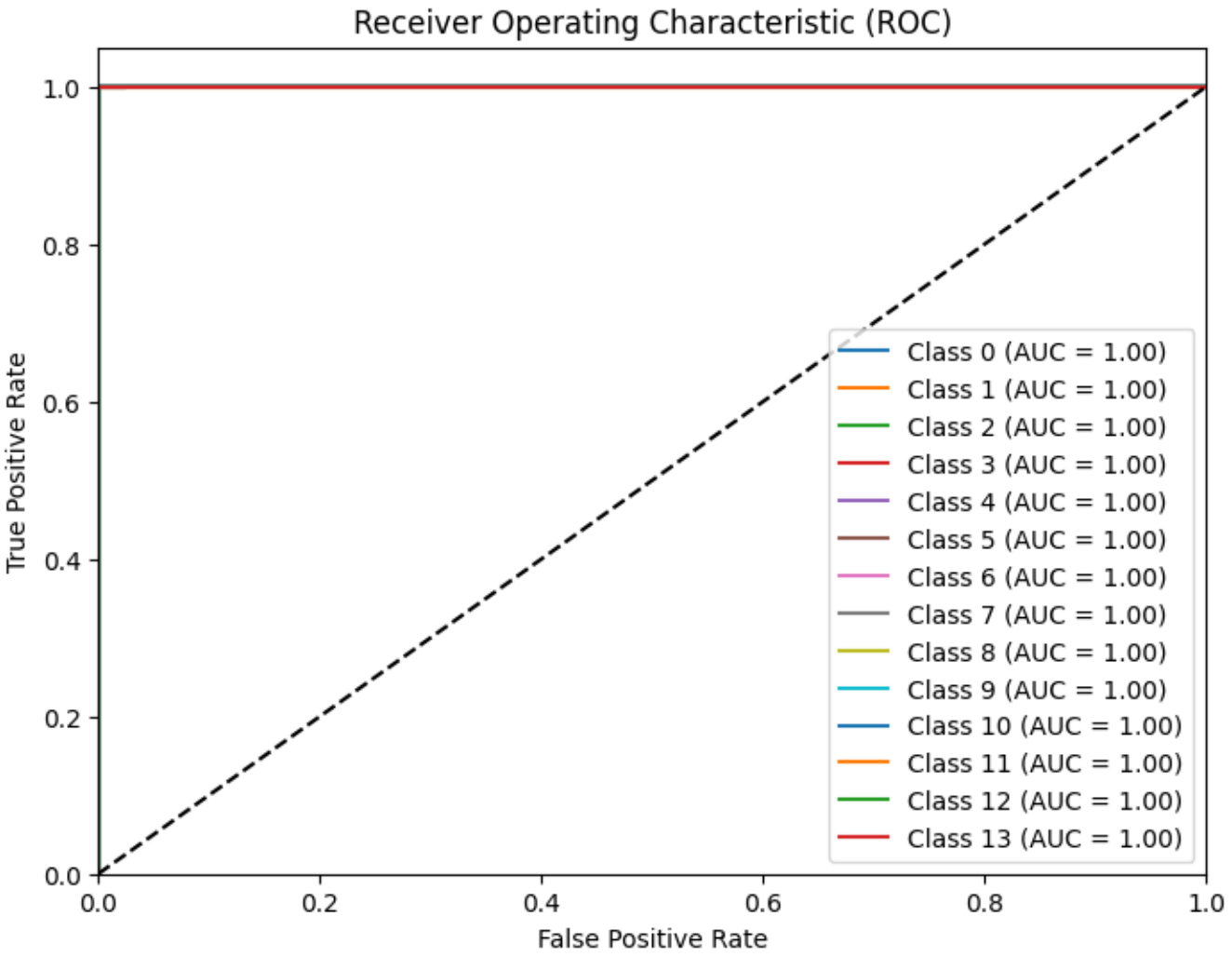


**FIGURE 11.** ROC curve for network traffic classification.

These metrics indicate the effectiveness of our proposed MMDTL in accurately classifying network traffic data. To visualize the performance of the model during training, we have plotted two key figures. The first figure shows the accuracy of the model across different epochs, indicating how well it improves over time.

The second figure represents the loss value, illustrating the convergence and optimization of the model during training. Figure 9 showcasing the modal accuracy over epochs demonstrates that the model steadily improves its accuracy as the training progresses. This indicates that our proposed framework effectively captures the underlying patterns and features of the data, enabling accurate classification of benign and attack samples.

Similarly, Figure 10 depicting the loss value shows a consistent decrease in the loss over epochs. This signifies that the model successfully minimizes the discrepancy between the predicted and actual labels, optimizing its performance and enhancing its ability to make precise classifications.

These figures collectively provide valuable insights into the training process and validate the effectiveness of our MMDTL approach in achieving high accuracy and low loss. To assess the overall predictive capability of our proposed MMDTL, we calculated the average ROC curve and AUC.

The ROC curve illustrates the trade-off between true positive rate and false positive rate for different classification thresholds. Remarkably, our model achieved an AUC value of 1, indicating excellent discriminatory power and perfect separation between benign and attack network traffic data. Figure 11 presents the average ROC curve, visually demonstrating the outstanding performance of our model in accurately classifying network traffic. The curve is positioned close to the top-left corner, indicating high sensitivity and specificity across various classification thresholds.

These findings further validate the robustness and efficacy of our proposed framework in network traffic classification. It showcases our model's ability to accurately distinguish

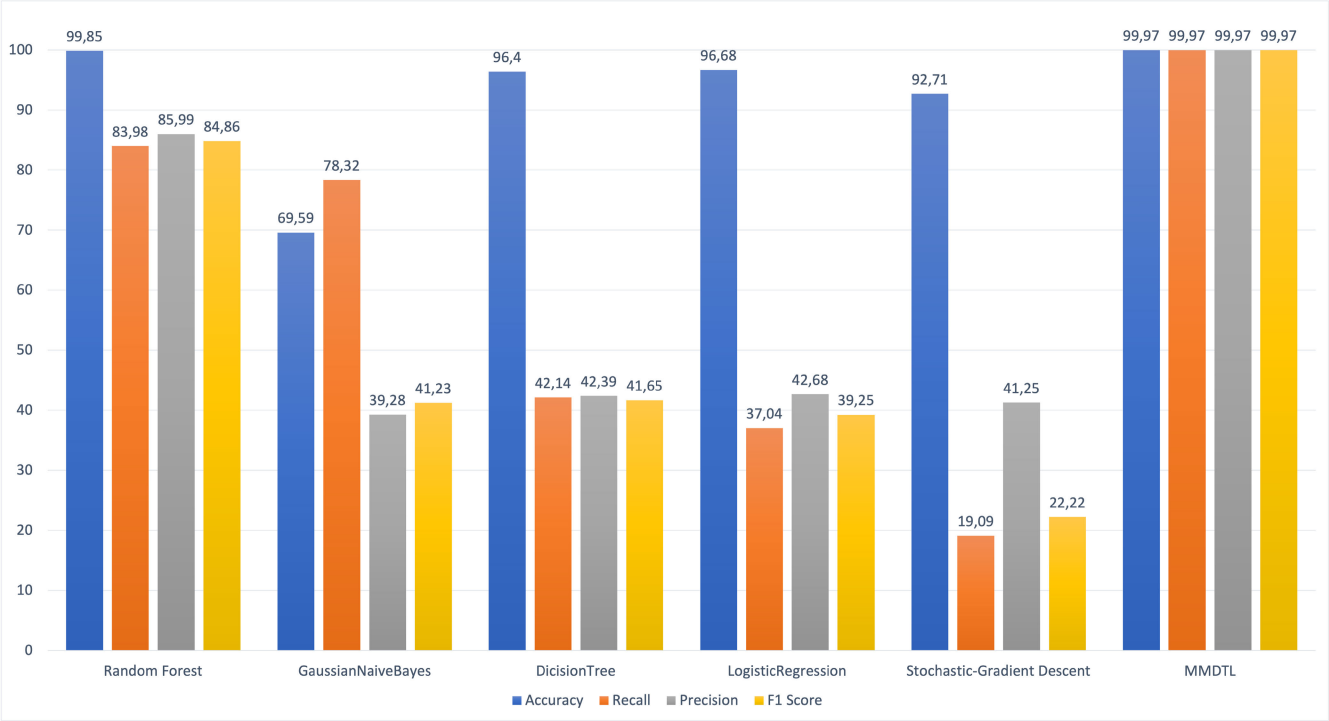


**FIGURE 12.** Multi-class performance comparison: MMDTL vs traditional ML models.

between benign and attack data, emphasizing its relevance and significance in enhancing network security in SDNs.

In the following sections, we will provide a comprehensive analysis and comparison of the implications of these results, highlighting the contributions and advancements our research brings to the field of network security.

### *E. PERFORMANCE COMPARISON*

In this section, we present the performance comparison of MMDTL with traditional machine learning algorithms for multi-class classification, including Gaussian Naive Bayes, Decision Tree, Logistic Regression, and Gradient Boosting. The evaluation was conducted on the same multivariate dataset and under the same conditions to ensure fair and consistent comparisons.

The results of the multi-class classification comparison are summarized in Figure 12:

Our proposed MMDTL model demonstrates superior multi-class classification performance compared to the traditional machine learning algorithms. It achieves an accuracy of 99.97% on the training dataset and 99.99% on the test dataset.

In addition to multi-class classification, we evaluated MMDTL on binary classification tasks using the same dataset. For binary classification, MMDTL achieved an accuracy of 99.99% on both the training and test sets. The binary classification results are summarized visually in Figure 13.

This high level of accuracy highlights the versatility of MMDTL in handling both multi-class and binary classification problems. The consistent results on both task types validate the robustness of MMDTL and its ability to generalize well across different network traffic classification domains, whether requiring multi-label or binary predictive modeling.

To further validate the robustness and efficacy of our proposed MMDTL framework, we performed a direct comparative evaluation against the state-of-the-art method recently published by [7], utilizing identical experimental conditions and datasets. Reference [7] previously proposed a CNN model for attack detection. When benchmarked under the same constraints, Our MMDTL framework achieved superior performance over [7] model on both training and test data, attaining over 99.97% accuracy on training data compared to 98.14% by [7] method, while on test data our MMDTL framework achieved 99.99% accuracy compared to 97.99% by [7] approach, as summarized in Figure 14. This direct comparison on an even playing field confirms the significant performance gains and state-of-the-art results of our proposed MMDTL, demonstrating its effectiveness for network traffic classification.

These results highlight the effectiveness of our MMDTL in accurately classifying network traffic data for multi-class prediction. The comparisons validate that MMDTL outperforms other machine learning approaches for the complex multi-label classification task.

Let's take a closer look at the performance of each method:

1- Random Forest: The Random Forest classifier stands out with remarkable performance, displaying high accuracy

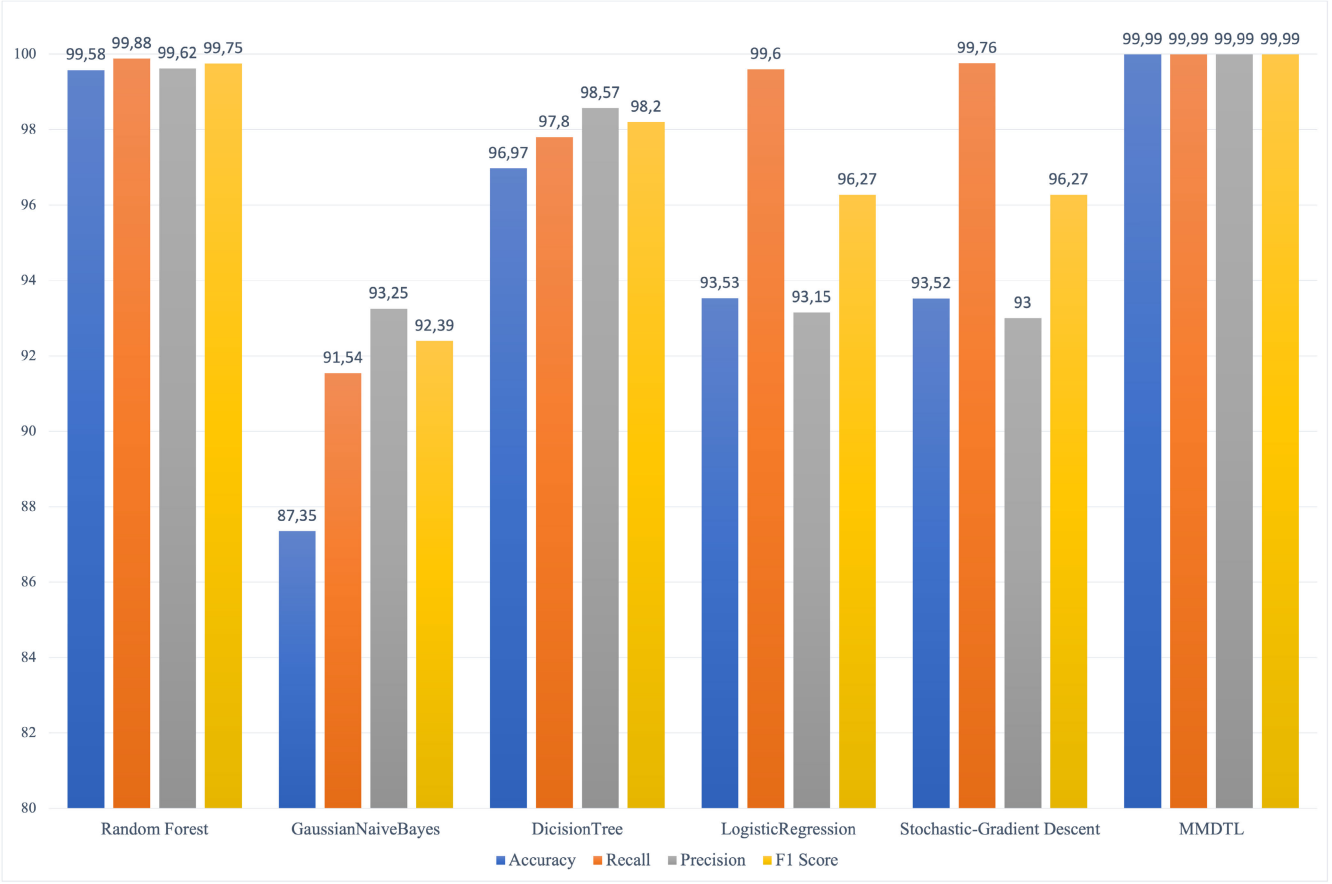


**FIGURE 13.** Binary performance comparison: MMDTL vs traditional ML models.

and an impressive F1 Score. These results indicate its strong overall performance in classifying the data.

2- Gaussian Naive Bayes: The Gaussian Naive Bayes classifier, while showing some limitations, still demonstrates acceptable accuracy. However, its F1 Score and precision are comparatively lower, suggesting challenges in distinguishing between different classes.

3- Decision Tree: The Decision Tree classifier exhibits good accuracy, but its Recall and Precision scores are relatively lower. This observation suggests that it may face difficulties in capturing class-specific patterns effectively.

4- Logistic Regression: The Logistic Regression classifier performs decently in terms of accuracy, but it struggles with Recall. This indicates that the model may encounter challenges in correctly identifying instances of certain classes.

5- Stochastic Gradient Descent: The Stochastic Gradient Descent classifier shows lower accuracy and Recall scores, suggesting difficulties in identifying instances of multiple classes accurately.

6- CNN [7]: The CNN demonstrates promising performance with high accuracy, Recall, Precision, and F1 Score. These results showcase the model's ability to learn complex features, making it well-suited for image-based classification tasks.

7- MMDTL (Proposed Method): Our proposed MMDTL method truly stands out with exceptional accuracy and outstanding Recall, Precision, and F1 Score. These impressive results highlight the effectiveness of our approach in multiclass classification. In fact, the MMDTL outperforms all other classifiers, solidifying its potential as a robust and reliable model for the task at hand.

Generally, the MMDTL method shows superior performance compared to the other classifiers, demonstrating its efficacy in addressing the multiclass classification challenge with exceptional accuracy and strong generalization capabilities. The promising outcomes of this study warrant further research and experimentation to explore its potential in real-world applications and across diverse datasets. Our proposed MMDTL model opens new avenues for enhancing multiclass classification accuracy, and we are enthusiastic about its potential impact in the field of machine learning.

The significant improvement in accuracy provided by our proposed model emphasizes its capability to handle the complexities and variations in network traffic patterns, outperforming traditional machine learning approaches. These

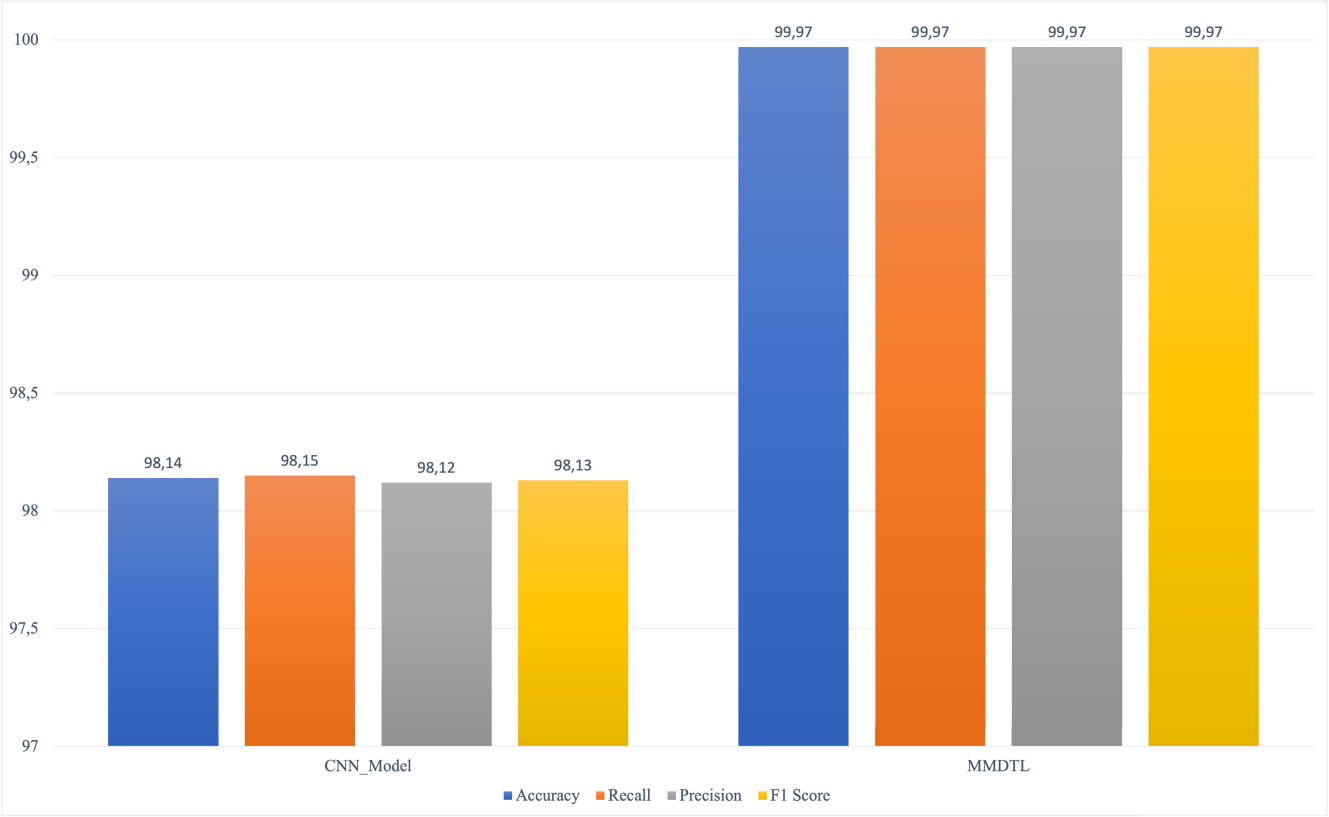


**FIGURE 14.** Comparison of classification accuracy: MMDTL vs [7] model.

findings reinforce the value of incorporating MMDTL in the context of network security in SDNs.

## V. CONCLUSION AND FUTURE WORK

In this research, we have proposed a novel MMDTL framework for real-time attack detection in SDNs. By integrating multiple data modalities, such as network traffic data, system logs, and user behavior data, MMDTL captures a comprehensive view of network behavior, leading to improved attack identification. The incorporation of transfer learning techniques further enhances the framework's performance by leveraging pre-trained knowledge from related domains, increasing its generalization capabilities and detection accuracy.

Our comprehensive evaluation demonstrates the superiority of the MMDTL framework over existing methods, achieving an exceptional accuracy of 99.99% on test data and 99.97% on train data. The framework's scalability and efficiency make it well-suited for deployment in real-world SDN environments with varying data volumes and complexities. Additionally, we have conducted an insightful analysis of the contribution of each modality to the overall performance, highlighting the benefits of leveraging multi-modal learning for SDN security.

Overall, the contributions of this research significantly advance the field of attack detection in SDNs. The MMDTL framework offers a comprehensive and effective solution to address the challenges posed by diverse attack types, enhancing network security. By bridging the gaps in the literature and considering a wide range of attacks, our research offers a more realistic and practical approach to SDN attack detection.

While our research has made notable contributions, there are several promising directions for future exploration and development in the field of SDN attack detection:

1- Expand Modalities and Data Sources: Investigate additional data modalities and sources to further enrich the understanding of network behavior and enhance attack detection capabilities. Exploration of non-traditional data sources, such as system resource utilization, may provide valuable insights for improved attack identification.

2- Adaptive Transfer Learning: Explore adaptive transfer learning approaches that dynamically adjust the knowledge transfer process based on the evolving attack landscape. This would enable the framework to adapt to new attack types and scenarios, ensuring continuous efficacy in real-world SDN environments.

3- Real-Time Implementation: Develop a practical implementation of the MMDTL framework in a real SDN environment, considering the resource constraints and latency requirements. Implementation challenges, such as model size and computational overhead, need to be addressed to ensure efficient real-time attack detection.

4- Real-World Deployment and Validation: Deploy the MMDTL framework in actual SDN infrastructures to validate its performance and effectiveness in real-world scenarios. Collaborations with industry partners and stakeholders would be valuable in conducting large-scale trials and gathering practical insights.

In conclusion, the proposed MMDTL framework represents a significant advancement in the field of SDN attack detection. With further exploration of the above-mentioned avenues, we believe that this research lays the foundation for continued advancements in network security and contributes to the establishment of robust and efficient attack detection solutions for SDNs.

## ACKNOWLEDGMENT

This work is a part of the Ph.D. thesis titled ''Designing a Smart Security Framework for Software Defined Networks'' at the Institute of Graduate Studies, Istanbul University-Cerrahpaşa, Istanbul, Türkiye.

## REFERENCES

[1] K. Nsafoa-Yeboah, E. T. Tchao, B. Yeboah-Akowuah, B. Kommey, A. S. Agbemenu, E. Keelson, and M. M. Khan, ''Software-defined networks for optical networks using flexible orchestration: Advances, challenges, and opportunities,'' *J. Comput. Netw. Commun.*, vol. 2022, pp. 1–40, Aug. 2022.

[2] D. B. Rawat and S. R. Reddy, ''Software defined networking architecture, security and energy efficiency: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 325–346, 1st Quart., 2017.

[3] H. Liu and B. Lang, ''Machine learning and deep learning methods for intrusion detection systems: A survey,'' *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019.

[4] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, ''A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019.

[5] B. Lampe and W. Meng, ''A survey of deep learning-based intrusion detection in automotive applications,'' *Expert Syst. Appl.*, vol. 221, Jul. 2023, Art. no. 119771.

[6] H. Sayadi, Y. Gao, H. M. Makrani, J. Lin, P. C. Costa, S. Rafatirad, and H. Homayoun, ''Towards accurate run-time hardware-assisted stealthy malware detection: A lightweight, yet effective time series CNN-based approach,'' *Cryptography*, vol. 5, no. 4, p. 28, Oct. 2021.

[7] A. H. Janabi, T. Kanakis, and M. Johnson, ''Convolutional neural network based algorithm for early warning proactive system security in software defined networks,'' *IEEE Access*, vol. 10, pp. 14301–14310, 2022.

[8] L. Yang and J. Teh, ''Review on vulnerability analysis of power distribution network,'' *Electr. Power Syst. Res.*, vol. 224, Nov. 2023, Art. no. 109741.

[9] K. P. Tran, H. D. Nguyen, and S. Thomassey, ''Anomaly detection using long short term memory networks and its applications in supply chain management,'' *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2408–2412, 2019.

[10] H. Elubeyd and D. Yiltas-Kaplan, ''Hybrid deep learning approach for automatic DoS/DDoS attacks detection in software-defined networks,'' *Appl. Sci.*, vol. 13, no. 6, p. 3828, Mar. 2023.

[11] S. A. Wagan, J. Koo, I. F. Siddiqui, N. M. F. Qureshi, M. Attique, and D. R. Shin, ''A fuzzy-based duo-secure multi-modal framework for IoMT anomaly detection,'' *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 1, pp. 131–144, Jan. 2023.

[12] M. Dib, S. Torabi, E. Bou-Harb, and C. Assi, ''A multi-dimensional deep learning framework for IoT malware classification and family attribution,'' *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1165–1177, Jun. 2021.

[13] B. Xue, H. Zhao, and W. Yao, ''Deep transfer learning for IoT intrusion detection,'' in *Proc. 3rd Int. Conf. Comput., Netw. Internet Things (CNIOT)*, May 2022, pp. 88–94.

[14] N. Marastoni, R. Giacobazzi, and M. Dalla Preda, ''Data augmentation and transfer learning to classify malware images in a deep learning context,'' *J. Comput. Virol. Hacking Techn.*, vol. 17, no. 4, pp. 279–297, Dec. 2021.

[15] K. Yan and D. Zhang, ''Feature selection and analysis on correlated gas sensor data with recursive feature elimination,'' *Sens. Actuators B, Chem.*, vol. 212, pp. 353–363, Jun. 2015.

**HANI ELUBEYD** received the M.S. degree in computer engineering from Eastern Mediterranean University, Famagusta, Cyprus, in 2016. He is currently pursuing the Ph.D. degree in computer engineering with Istanbul University-Cerrahpaşa, Istanbul, Türkiye.

His research interests include data science methodologies and their applications, the evolution and impact of artificial intelligence systems, and the challenges and strategies associated with cybersecurity. He has contributed significantly to the advancement of these fields through rigorous research and collaboration.

**DERYA YILTAS-KAPLAN** received the B.Sc., M.Sc., and Ph.D. degrees in computer engineering from Istanbul University, Istanbul, Türkiye, in 2001, 2003, and 2007, respectively.

She completed the Postdoctoral Research with North Carolina State University. She is currently an Associate Professor with the Department of Computer Engineering, Istanbul University-Cerrahpaşa. Specializing in computer sciences, her research is notably focused on computer networks and data routing, making significant contributions to engineering and technology. She received a Postdoctoral Research Scholarship from The Scientific and Technological Research Council of Türkiye (TUBITAK).

**ŞERİF BAHTİYAR** received the bachelor's degree in control and computer engineering and the master's degree in computer engineering from Istanbul Technical University, in 2001 and 2004, respectively, and the Ph.D. degree in computer engineering from Boğaziçi University, in 2011.

He completed a Postdoctoral Research with Technische Universität Berlin, in 2012 and 2013. With a research portfolio encompassing information security and cryptology, algorithms and computation theory, computer and communication networks, intelligent systems, and financial systems, he has solidified his reputation as a leading mind in the world of computer sciences and technology.