Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 21(1): 1527–1553.
DOI: 10.3934/mbe.2024066
Received: 01 September 2023
Revised: 05 December 2023
Accepted: 06 December 2023
Published: 29 December 2023

Research article

Advancements in enhancing cyber-physical system security: Practical deep learning solutions for network traffic classification and integration with security technologies

Shivani Gaba[1], **Ishan Budhiraja**[1], **Vimal Kumar**[1] **and Aaisha Makkar**[2,*]

[1] School of Computer Science Engineering and Technology, Bennett University, Greater Noida U.P., India

[2] Department of Computer Science, College of Science and Engineering, University of Derby, UK

* **Correspondence:** Email: a.makkar@derby.ac.uk.

Abstract: Traditional network analysis frequently relied on manual examination or predefined patterns for the detection of system intrusions. As soon as there was increase in the evolution of the internet and the sophistication of cyber threats, the ability for the identification of attacks promptly became more challenging. Network traffic classification is a multi-faceted process that involves preparation of datasets by handling missing and redundant values. Machine learning (ML) models have been employed to classify network traffic effectively. In this article, we introduce a hybrid Deep learning (DL) model which is designed for enhancing the accuracy of network traffic classification (NTC) within the domain of cyber-physical systems (CPS). Our novel model capitalizes on the synergies among CPS, network traffic classification (NTC), and DL techniques. The model is implemented and evaluated in Python, focusing on its performance in CPS-driven network security. We assessed the model's effectiveness using key metrics such as accuracy, precision, recall, and F1-score, highlighting its robustness in CPS-driven security. By integrating sophisticated hybrid DL algorithms, this research contributes to the resilience of network traffic classification in the dynamic CPS environment.

Keywords: network traffic classification; machine learning; deep learning; hybrid model

1. Introduction

In advanced communication networks, it is crucial to classify network traffic, which is the process of identifying various applications or traffic kinds by examining data packets that have been received. Accurate traffic classification is necessary for performing advanced network management activities including ensuring network quality-of-service (QoS) and discovering network anomalies. The payload-based approach, the machine learning (ML) approach, and the port-based approach [1] are the three currently in-use approaches for categorizing network traffic. Port-based traffic classification, which extracts port numbers from packet headers in transmission control protocol (TCP) or user datagram protocol (UDP) packets, is the simplest and oldest form of traffic classification. The payload-based method, also known as deep packet inspection (DPI), examines packet payloads using established patterns for different protocols. Although these two techniques may successfully classify traffic with high accuracy in some circumstances, they are constrained by the widespread use of encrypted data in contemporary communication networks [2]. For instance, the number of virtual private network (VPN) sessions significantly reduces the port-based technique's accuracy. The introduction of secure transmission protocols like hyper text transmission protocol (HTTPS) over secure socket layer (SSL) and secret file transfer protocol (SFTP) increases the difficulty of categorizing application types using the payload-based method. As a result, research interest in the machine learning approach to traffic classification has lately increased. This approach makes the assumption that encrypted packets feature certain intra-class and interclass discriminative patterns that can be recognized by ML techniques [3], rather than merely being sequences of utterly random bits [4, 5]. This method often makes use of a dataset that was created by someone else or made available online and is split into two parts, a training set and a test set, and contains network packets with precise labels. A statistical model, or classifier, is trained using the former, the training set, to predict the labels of the latter, the test set, for the purpose of performance assessment.

Although, progressive technologies are at risk of network intrusion from various aspects the defacement from cyber-attacks never stopped. The core challenge of cyber security not only requires exponential growth from the internet of things (IoT) but also requires architectures for detecting these kinds of potential attacks. To counter cyber attacks in cyber-physical systems (CPS), it is serious about consolidating the identification of cyber threats by executing various countermeasures for blocking potential risks.

1.1. General process of network traffic classification

The importance of network functionality and control is currently represented by a variety of services and applications. The functionality of the framework for categorizing network traffic is shown in Figure 2 [6]. This framework has several phases that are designed to gather data, extract attributes, mitigate and select attributes, and ultimately construct a framework. The entire process illustrates how network traffic classification (NTC) approaches can be used to identify or categorize unidentified types of network traffic using ML algorithms which is explained in Section five.

The input of a conventional traffic network might be employed to create a dataset for feature selection processing. Due to its impact on how well the traffic categorization performs, feature extraction (FE) and selection also play a significant role. One of the most crucial steps is FE, which enables measuring or computing features that could reveal information about the process's condition. In a nutshell, an FE process computes various metrics that reflect particular characteristics in the gathered data [7, 8]. The basic objective is to find words that more accurately describe the issue. A structured table made up of columns of attributes with each row representing a sample and an optional additional column showing the sample's current status is the end result of the FE process (usually known as label or class). When given a lot of features, ML models may experience issues. The models are pruned to, among other things, increase overfitting, decrease accuracy, and increase computing overhead [9–11]. These issues are frequently caused by the dimensionality curse [12, 13].

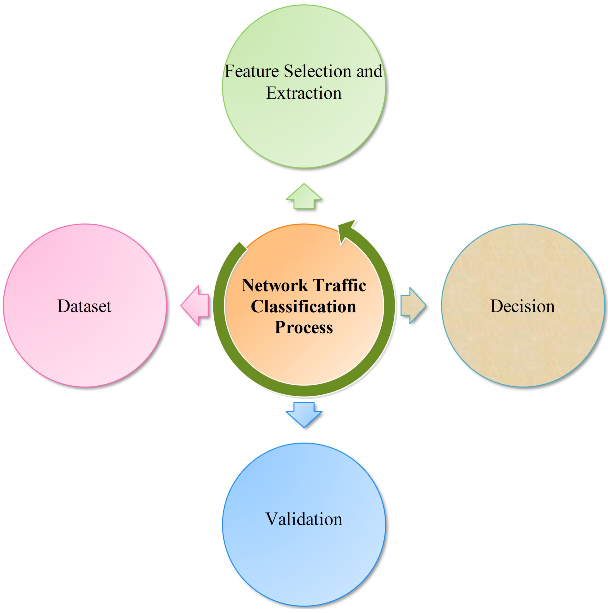

Feature Selection and Extraction
Dataset
Network Traffic Classification Process
Decision
Validation

Figure 1. Network traffic classification process.

Generally speaking, feature selection (FS)) is frequently used in this discipline to choose the most essential characteristics and to increase the precision of the ML models. Finding the best ML algorithm is crucial for traffic classification due to the large number of available ML techniques. In particular, the majority of the works have chosen their models by creating and testing a number of them until they locate the one with the best performance [14]. The decision process (DP) could then use ML approaches to determine the traffic classification class [15]. The validation process (VP), which assesses classification accuracy, is then utilized to validate the traffic categorization results.

1.2. Motivation

The motivation for NTC using the hybrid deep learning (DL) model is driven by:

- NTC is the foremost step for analyzing and identifying the various types of applications floating in the network.These classifications are helpful for the cyber attacks in CPSs.
- The increasing need for effective and efficient security solutions in CPS. CPS has become an integral part of serious infrastructures like power grids, transportation systems, and communication networks. Cyber attacks that target these classes can seriously harm both people and the environment.
- The effectiveness of item DL has been demonstrated in a number of applications, including speech recognition, natural language processing, and picture recognition. It has also been used for security concerns including malware categorization and intrusion detection.
- The application of DL for CPS has the ability to deliver precise and real-time threat identification, allowing for quick attack mitigation and minimizing the effect of cyber threats on CPS.

1.3. Contribution

- The development of more practical and efficient DL algorithms for identifying a variety of cyberattacks in CPS, enhancing the generalization capability of DL models, and investigating the integration of DL with other security technologies are just a few of the many open challenges and opportunities in this field.
- The computing cost and positional viability of DL approaches in actual CPS systems also need to be taken into account.
- We can contribute to this field by addressing these challenges and exploring new solutions to improve the security of CPS against cyber attacks.

1.4. Organization

There are several sections in the paper. The paper's literature review is discussed in section two. In section three, the architecture of cyber-physical systems is covered. The many DL-based cyber attack detection strategies used in cyber-physical systems are covered in section four. In section five, a few of the machine learning techniques that are currently in use for classifying network traffic are explained. Section six of the paper's methodology is described. Section seven discusses experimental results and performance evaluation. Section eight contains a description of the results and discussions. In Section nine, the paper is concluded. Figure 2 displays the paper's taxonomy.

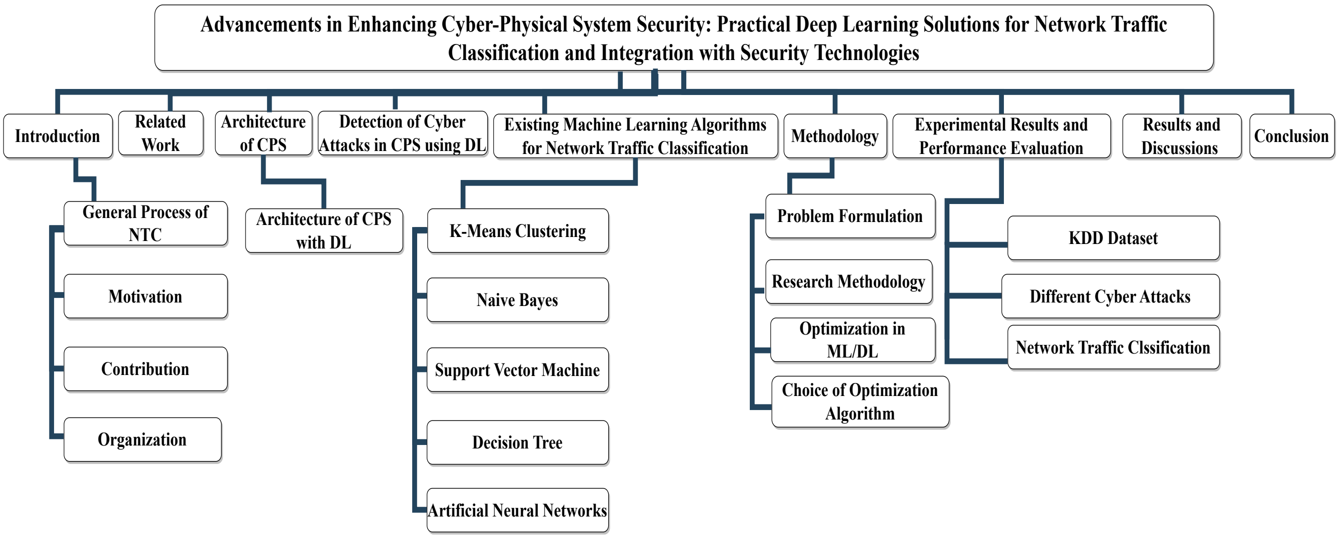

Advancements in Enhancing Cyber-Physical System Security: Practical Deep Learning Solutions for Network Traffic Classification and Integration with Security Technologies
Introduction
Related Work
Architecture of CPS
Detection of Cyber Attacks in CPS using DL
Existing Machine Learning Algorithms for Network Traffic Classification
Methodology
Experimental Results and Performance Evaluation
Results and Discussions
Conclusion
General Process of NTC
Motivation
Contribution
Organization
Architecture of CPS with DL
K-Means Clustering
Naive Bayes
Support Vector Machine
Decision Tree
Artificial Neural Networks
Problem Formulation
Research Methodology
Optimization in ML/DL
Choice of Optimization Algorithm
KDD Dataset
Different Cyber Attacks
Network Traffic Clssification

Figure 2. Taxonomy of paper.

Table 1. List of Acronyms/Abbrevations used in paper.

Acronyms/Abbreviations	Definition
ML	Machine Learning
DL	Deep Learning
NTC	Network Traffic Classification
CPS	Cyber Physical System
QoS	Quality of Service
TCP	Transmission Control Protocol
UDP	User Datagram protocol
DPI	Deep Packet Inspection
VPN	Virtual Private Network
HTTPS	Hyper Text Transmission Protocol over Secure Socket Layer
SFTP	Secret File Transfer Protocol
IoT	Internet of things
FE	Feature Extraction
FS	Feature Selection
DP	Decision Process
VP	Validation Process
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
GAN	Generative Adversarial Networks
LSTM	Long Short-Term Memory
MIMETIC	Mobile encrypted traffic classification using multimodal deep learning
IDS/IPS	Intrusion Detection System/Intrusion Prevention System
NB	Naive Bayes
SVM	Support Vector Machine
DT	Decision tree
ANN	Artificial Neural Network
NLP	Natural Language Processing
KDD	Knowledge Discovery in Databases
SGD	Stochastic Gradient Descent
DOS	Denial of Service
R2L	Remote to Local
U2R	User to Root
P2P	Peer to Peer
KNN	K-nearest neighbours

2. Related works

Here's a brief literature review on NTC using DL models that combine Convolutional neural network (CNN) and long short-term memory (LSTM) in the context of cyber-physical systems (CPSs):

The authors [1] proposed a deep learning framework for network traffic classification in CPS.The model combines CNN and LSTM to extract spatial and temporal features from network traffic data [16, 17]. The authors experiment with different network architectures, layer configurations, and hyperparameters to optimize classification accuracy. The proposed model achieves high accuracy in classifying different types of network traffic in CPS.

The authors [2] proposed a deep learning-based NTC approach for CPS. They combine CNN and LSTM layers to capture both spatial and temporal patterns in network traffic data. The model is trained and evaluated on a dataset collected from a CPS environment, achieving high classification accuracy. The review exhibits the viability of the consolidated CNN-LSTM model for NTC in CPS.

This paper [3] centers around NTC for interruption location in CPS utilizing DL techniques. The proposed model joins CNN and LSTM layers for capturing spatial and temporal patterns in network traffic data. The authors direct experiments on a real-world dataset and contrast the presentation and other classification methods. The results show that the consolidated CNN-LSTM model outperforms conventional exactness and identification rate strategies.

The study [4] proposes a hybrid DL model for NTC in CPS. The model joins CNN and LSTM layers to extricate spatial and temporal features from network traffic data. The authors likewise consolidate consideration systems to feature significant elements in the network traffic sequences. The proposed model accomplishes high classification precision and outperforms traditional Artificial Intelligence (AI) calculations in CPS environments.

These papers feature the viability of combined CNN and LSTM in DL models for NTC in CPS. The joining of spatial and transient features from CNN and LSTM separately assists in capturing intricate patterns in the network with trafficking information, empowering exact type and interruption recognition in CPS environments.

The authors [18] make a substantial contribution to the domain of NTC by the use of DL techniques. The main area of their study centers around the classification of encrypted mobile data, a topic that has become increasingly important as the use of encryption in mobile networks continues to rise.

The researchers have made notable contributions in the field, focusing on the advancement and evaluation of DL methods for the classification of encrypted mobile communications. The researchers carried out an extensive array of tests to assess the practical efficacy of these deep neural networks. The study conducted by the authors provides valuable insights derived from their experiments, which hold considerable relevance for scholars and professionals engaged in the domain of NTC. The results of their study contribute valuable insights and comprehension on the challenges and complexities associated with the use of DL methods in the field of traffic classification.

Furthermore, this research investigates the challenges associated with the categorization of encrypted data on cellular networks, therefore enhancing our understanding of the complexities associated with this endeavor. This study is of considerable significance since it addresses a practical issue within the domain of network administration and security. The information provided offers significant perspectives that can inform decision-making processes related to improving network performance and adopting robust security measures.

The authors in [19] provide a revolutionary technique in their article titled "Deep packet: A novel approach for encrypted traffic classification using deep learning" that makes a significant addition to the subject of NTC, particularly in relation to encrypted data. This research presents a novel methodology for the classification of encrypted network packets utilising advanced DL algorithms. The primary contribution of this research is in the creation of a novel technology referred to as "Deep Packet", which uses DL models to effectively categorize encrypted network data. The authors effectively tackle the complex problem of recognizing and categorizing encrypted packets, even in cases when the payload is hidden, by utilizing deep neural networks. The aforementioned approach has practical significance within the domain of network security and management, since the increasing frequency of encrypted communication has presented obstacles for conventional traffic categorization techniques. The methodology employed by the authors facilitates precise categorization of this particular form of network traffic, a critical aspect for a range of network-centric applications such as intrusion detection, QoS administration, and network optimization. The paper by Lotfollahi, Mohammad, et al. makes a substantial contribution to the field of NTC by introducing a novel and efficient method for classifying encrypted traffic. This approach enhances the abilities of network administrators and security experts in effectively managing and securing contemporary network environments. This study focuses on a significant issue within the subject of NTC and highlights the increasing significance of DL methodologies within this area.

The authors in [20] introduce a novel methodology for the categorization of mobile encrypted traffic in their scholarly article titled "MIMETIC: Mobile encrypted traffic classification using multimodal deep learning". The primary contribution of this study is in the advancement and use of multimodal DL methodologies to tackle the complexities related to the classification of encrypted data in mobile networks. The MIMETIC technique integrates many forms of information, including packet-level characteristics and flow-level data, in order to improve the precision of traffic categorization. The authors successfully enhance the categorization accuracy of encrypted communications by employing DL models that efficiently analyze and integrate diverse data modalities. This research holds significant relevance within the contemporary domain of network management and security, whereby the presence of encrypted communication presents a substantial obstacle for conventional categorization techniques. The authors offer a multimodal DL strategy that shows promise in reliably classifying the type of mobile encrypted data. This classification is crucial for many tasks such as network optimization, intrusion detection, and QoS monitoring. In brief, the study authored by Aceto, Giuseppe, et al. presents a noteworthy advancement in the domain of NTC with the introduction of the "MIMETIC" technique. This study enhances the current knowledge in encrypted traffic categorization by integrating multimodal DL algorithms. It provides practical approaches to tackle the intricate challenges present in modern mobile networks.

The authors [21] made a significant scholarly contribution in the domain of NTC, with a specific emphasis on the IoT sector. Their paper, titled "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things", was published in Institute of Electrical and Electronics Engineers (IEEE) Access. The primary contribution of this study is in the implementation of a traffic categorization methodology that integrates CNNs with recurrent neural networks (RNNs). The present hybrid neural network architecture has been specifically developed to efficiently categorize network traffic associated with the IoT, a domain that frequently poses distinctive obstacles owing to the varied and ever-changing characteristics of IoT devices and applications. The authors improve the system's capability to capture and analyze temporal and spatial patterns in network traffic data by utilizing both CNNs and RNNs. The aforementioned outcome leads to a heightened level of precision and resilience in the categorization of IoT data flow. This is of utmost importance for the effective administration of networks, the safeguarding of systems, and the allocation of resources within IoT settings. The research holds significance within the expanding IoT domain, where the accurate categorization of traffic plays a crucial role in enhancing network efficiency and safeguarding the integrity of IoT devices and applications. The study conducted by Lopez-Martin, Manuel, and their colleagues offers a helpful answer for effectively tackling the unique issues associated with the categorization of IoT data. Their research showcases the possibility of integrating DL techniques to augment the capabilities of network traffic classifiers within this particular field.

A large corpus of cutting-edge DL-based proposals for NTC (non-exhaustive list) are explained below by different authors and it is stated in Table 2.

3. Architecture of CPS

The architecture of a CPS typically consists of three main components: Physical layer, network layer, and application layer.

- **Physical Layer:** This is the bottom layer of the CPS's architecture and it comprises of physical devices and sensors that interact with the physical world. These devices and sensors collect and transmit data to the other layers of the system.
- **Network Layer:** The network layer is accountable for transmitting the data among the physical layer and application layer. It ensures the reliable and secure transmission of data and provides communication protocols for data transfer.
- **Application Layer:** The application layer is accountable for the processing and interpretation of the data received from the physical layer. It gives the associate for users to collaborate with the CPS and executes the control logic for the physical devices in the framework.

The architecture of a CPS is intended to be flexible and adaptable, considering the expansion or evacuation of physical devices and sensors depending on the situation. It is additionally designed to be secure, with reasonable safety efforts against cyber attacks, and guarantees the secrecy and concealment of penetrating information.

Table 2. Comparison between the state of the art existing schemes.

SNo	Methodology	Dataset	Results	Challenges Addressed	Limitations	Year & References
1	CNN-based feature extraction, classification.	Public network traffic dataset.	Achieved 95% accuracy in traffic classification.	Handling high-dimensional network data.	Limited evaluation on diverse attack scenarios.	2015 & [22]
2	Ensemble of CNN and LSTM networks.	Network intrusion detection dataset.	Improved accuracy in detecting network intrusions.	Integrating strengths of both CNN and LSTM architectures.	Ensemble techniques might increase computation.	2016 & [23]
3	LSTM-based sequence modeling.	Time series network traffic data.	Captured long-term dependencies in traffic data.	Capturing subtle anomalies in complex network traffic.	Dependency on sequence length for LSTM performance.	2017 & [24]
4	CNN for feature extraction, LSTM for sequence modeling.	Various time series datasets.	Demonstrated improved performance with hybrid model.	Improved handling of sequential and spatial patterns.	Computationally intensive due to dual architecture.	2018 & [25]
5	Combined CNN and LSTM architectures.	Intrusion detection dataset for CPS.	Achieved high accuracy in detecting intrusions.	Improved accuracy for CPS intrusion detection.	Limited evaluation of model's generalization.	2019 & [26]
6	CNN-LSTM architecture for intrusion detection.	Network intrusion detection dataset.	Detected a wide range of network intrusions.	Detecting zero-day attacks and new intrusion patterns.	Limited analysis of false positive rates.	2020 & [27]
7	Hybrid CNN-LSTM for anomaly detection.	Real-world CPS network traffic data.	Improved detection of anomalies in CPS networks.	Enhanced security of CPS by detecting novel threats.	Dependency on high-quality labeled training data.	2021 & [28]
8	CNN for spatial features, LSTM for temporal dependencies.	Industrial control system network traffic.	Achieved real-time and accurate traffic classification.	Addressing the security of industrial control systems.	Limited exploration of model robustness to adversarial attacks.	2022 & [29]
9	Attention-enhanced CNN-LSTM architecture.	CPS network traffic dataset.	Enhanced attack detection accuracy in CPS.	Improving focus on relevant features for attack detection.	Possible performance variation with different attention mechanisms.	2022 & [30]
10	Investigation of adversarial attacks.	Synthetic CPS dataset for adversarial attacks.	Demonstrated vulnerabilities of CNN-LSTM models.	Enhancing the understanding of model vulnerabilities.	Limited evaluation of countermeasures for adversarial attacks.	2023 & [31]
11	Deep Learning	-	Experimental, Lessons Learned, Challenges	Mobile Encrypted Traffic Classification	-	2019 & [18]
12	Deep Learning	-	Novel Approach for Encrypted Traffic Classification	Encrypted Traffic Classification, Network Security	-	2020 & [19]
13	Multimodal Deep Learning	-	Mobile Encrypted Traffic Classification	Mobile Encrypted Traffic Classification	-	2019 & [20]
14	CNN and RNN	-	Network Traffic Classification for IoT	IoT Traffic Classification, Network Optimization	-	2017 & [21]
15	Advancements in Enhancing Cyber-Physical System Security: Practical Deep Learning Solutions for Network Traffic Classification and Integration with Security Technologies	KDD Dataset	The model achieves accuracy of 97 percent which is compared with CNN, SVM and KNN.	Understanding the model vulnerabilities.	Transfer learning model for network traffic classification.	Our Paper

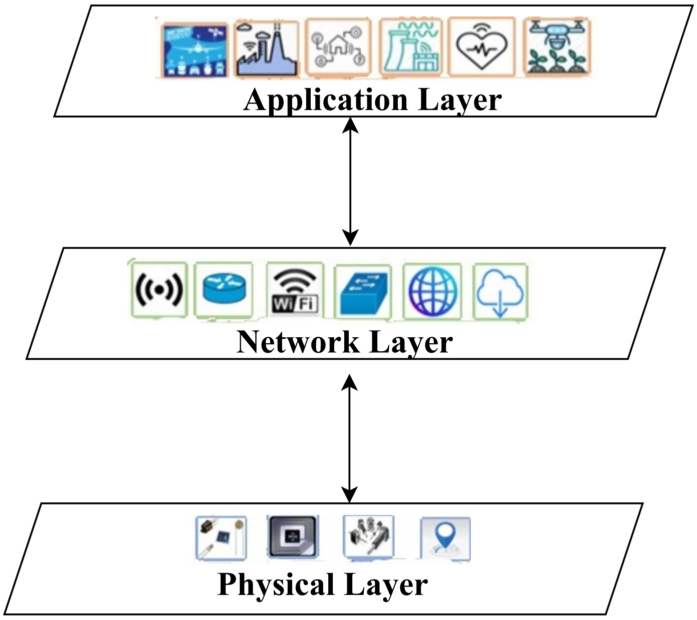


Figure 3. Architecture of CPS.

3.1. Architecture of CPS with DL

The architecture of a CPS with DL can be explained as follows:

- **Sensors and Actuators:** These are the physical components of the cyber-physical system that collects data from the physical environment and controls physical processes.
- **Communication Network:** This is the network that links the sensors and actuators to the central control system. The communication network should be secure and reliable to prevent cyber attacks.
- **Central Control System:** This is the core of the CPS, responsible for dispensation of the data from the sensors and transferring control commands to the actuators.
- Data Storage: This is where the data collected from the sensors and control commands directed to the actuators are stored. The data storage should be secure to prevent unauthorized access.
- **Data Preprocessing:** This is the procedure of altering raw data into a suitable format for training and using DL models. The preprocessing step may include cleaning, normalizing, and transforming the data.
- **DL Model:** This DL algorithm is prepared on the pre-handled information to gain the standard patterns to conduct and recognize deviations that might demonstrate a cyber attack. The DL model might be a CNN, RNN, or other appropriate architecture.
- **Model Deployment:** The trained DL model is deployed on the central control system to analyze new data in real time and detect any peculiarities that might show a cyber attack. The model can likewise be fine-tuned after some time as new information opens up and the threats to the CPS change and advance.

The architecture of a CPS with DL includes coordinating DL algorithms into the focal control framework to improve its protection from cyber attacks. The DL models are prepared on information gathered from the sensors and actuators and are utilized to recognize anomalies and distinguish assaults continuously.

4. Detection of cyber attacks in CPS using DL

Cyber attacks in CPS can be detected using DL methods. The following are some common approaches for detecting cyber attacks in CPS using DL:

- **Anomaly Detection:** One of the most common DL-based approaches for detecting cyber attacks in CPS is anomaly detection. This involves training a DL model on normal network behavior, then using the model to detect anomalies in the network traffic that may indicate an attack [31].
- **Intrusion Detection Systems (IDS):** Another common approach is to use DL-based intrusion detection systems (IDS). These systems are trained on a large dataset of network traffic, both normal and attack traffic, and are able to identify specific types of attacks by analyzing the network traffic.
- **Signature-based Detection:** Another approach is to use DL-based signature-based detection, which involves creating a signature for each type of attack and using DL models to detect attacks by matching the network traffic to the attack signatures.
- **Hybrid Methods:** Another approach is to use hybrid methods, which combine multiple DL-based techniques to detect cyber attacks in CPS. For example, a hybrid method may use anomaly detection and IDS together to achieve a higher level of accuracy in detecting attacks.

These are few common approaches for the detection of cyber attacks in CPS using DL, but the exact approach depends on the precise requirements and constraints of CPS, as well as the available data and computational resources.

5. Existing ML algorithms for NTC

When the statistical data pertaining to the specific application traffic is reviewed, ML algorithms are able to capture and identify the traffic data packets. In this topic, diverse solutions utilizing various ML algorithms are frequently found. Finding the best ML algorithm is crucial for traffic classification due to the large number of available ML techniques [32]. In particular, the majority of the works have chosen their models by creating and testing a number of them until they locate the one with the best performance. Following are some typical ML techniques for classification network traffic:

1) K-means Clustering: This method is one of the most used unsupervised ML-based algorithms. It could find unlabeled data in a variety of clusters, as shown by [33]. To perform K-means clustering, two key processing requirements—the dataset and the number of clusters—must be met. When there are K clusters, the clustering problem is solved using the K-means algorithm in three steps: initializing the K cluster, using the distance function at each network node that is closest to the center node, assigning a new centroid while taking into consideration the current node, and stopping the classifier [34]. Few researchers remove outliers and provide a robust dataset for ML-based algorithms using the K-means clustering technique. As a classifier, K-means clustering is used by other researchers to distinguish between nodes' legitimate and malicious conduct in traffic networks.

2) Naive Bayes (NB): The Bayesian algorithm's main responsibility is to handle prediction problems. This method produces useful learning algorithms and incorporates the observed data. The NB algorithm helps by providing a meaningful perception that allows the learning methods to be computed and comprehended. This approach is used to compute precise hypothesis probabilities and remove noise from input data [35]. It is not particularly difficult to create a huge dataset for traffic categorization because it uses the Bayes network theorem. Also, the NB technique is dependable and effective for identifying and classifying complex traffic.
3) Support Vector Machine: The support vector machine (SVM) is a trusted machine learning-based technique. Large volumes of traffic data as well as internet traffic can be recognized and categorized by it. It is used in regression and classification, which rely on the hyperplane's separation. However, it is most usually used to solve a classification problem because of its ability to split two classes using a hyperplane. The fundamental objective of SVM is to find a hyperplane that can clearly categorize the data. Hyperplanes are often referred to as decision boundaries. They help to classify the data points [36]. Closer to the hyperplane data points, called support vectors, influence the location and direction of the hyperplane. The SVM approach performs well when there are more attributes and fewer sample cases.
4) Decision tree (DT): A class of supervised ML-based algorithms includes the decision tree (DT) method. It may handle continuous and categorical forms of input and output variables. The DT is made up of numerous leaves, each of which contains a number of branches where you can represent different traffic classes. There are two methods for developing ML-based DTs, such as C4.5 and ID3. These tree methods were developed by introducing the entropy notion to the DT and the training dataset. Two types of variables, such as continuous variables and categorical variables, make up the DT. The phrases root node, splitting node [37], terminal node, decision node, etc. are all used in relation to DTs. The DT approach is used to construct new variables, develop links between existing variables, and identify and categorize the goal variable. The DT just requires a little amount of traffic statistics and may be used for both category and numerical variables.
5) Artificial Neural Network (ANN): One of the most popular ML-based techniques, the artificial neural network (ANN) technique models the relationship between input and output data. It is thought to be particularly trustworthy for traditional regression and statistical data modeling. The primary benefit of the ANN approach is the reliable processing on a massively parallel implementation, which substantially advances and meets the demand for ML-based technique development. The ANN attracts a lot of attention from researchers with backgrounds in computer and network technology since it can process artificial neurons that can carry out a variety of computational operations on the applied inputs [38]. The ANN model becames so well known among other ML-based techniques at the same time that the SVM method was invented. Nevertheless, processing application inputs with ANN and SVM algorithms takes a long time.

6. Methodology

6.1. Problem formulation

The problem statement for detecting cyber attacks in CPS using DL can be defined as:

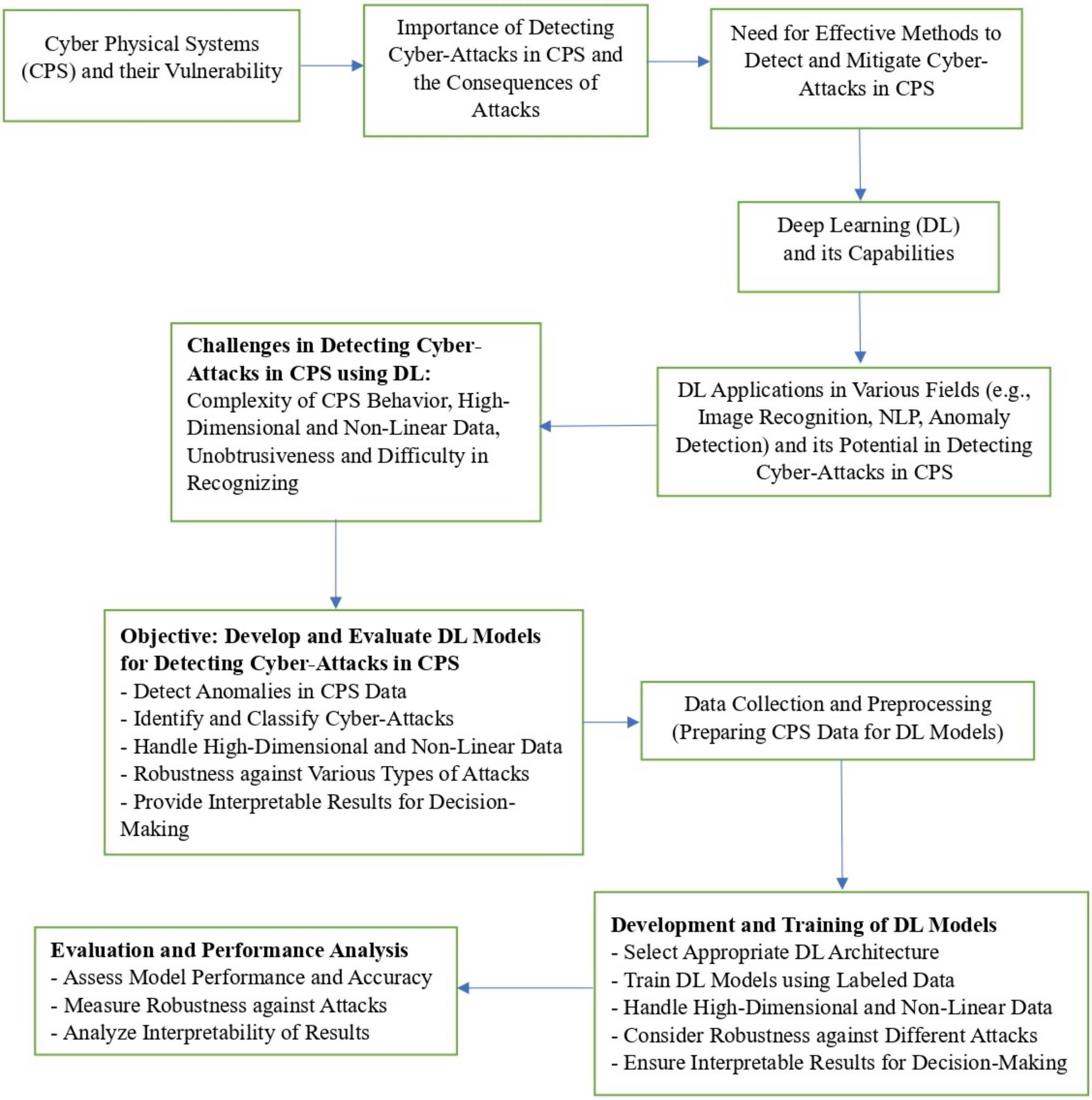

Cyber Physical Systems (CPS) and their Vulnerability
Importance of Detecting Cyber-Attacks in CPS and the Consequences of Attacks
Need for Effective Methods to Detect and Mitigate Cyber-Attacks in CPS
Deep Learning (DL) and its Capabilities
DL Applications in Various Fields (e.g., Image Recognition, NLP, Anomaly Detection) and its Potential in Detecting Cyber-Attacks in CPS
Challenges in Detecting Cyber-Attacks in CPS using DL: Complexity of CPS Behavior, High-Dimensional and Non-Linear Data, Unobtrusiveness and Difficulty in Recognizing
Objective: Develop and Evaluate DL Models for Detecting Cyber-Attacks in CPS
- Detect Anomalies in CPS Data
- Identify and Classify Cyber-Attacks
- Handle High-Dimensional and Non-Linear Data
- Robustness against Various Types of Attacks
- Provide Interpretable Results for Decision-Making
Data Collection and Preprocessing (Preparing CPS Data for DL Models)
Development and Training of DL Models
- Select Appropriate DL Architecture
- Train DL Models using Labeled Data
- Handle High-Dimensional and Non-Linear Data
- Consider Robustness against Different Attacks
- Ensure Interpretable Results for Decision-Making
Evaluation and Performance Analysis
- Assess Model Performance and Accuracy
- Measure Robustness against Attacks
- Analyze Interpretability of Results

Figure 4. Problem statement: Detecting cyber attacks in CPSs using deep learning.

- CPS are becoming progressively pervasive, encompassing various aspects of our daily lives such as transportation systems, power grids, and healthcare systems. Due to their widespread use and critical importance, CPSs are becoming increasingly vulnerable to cyber attacks, which can have serious consequences for both the system and society as a whole. Therefore, there is a pressing need to develop effective methods to detect and mitigate cyber attacks in CPS.
- DL has been exhibited to be compelling in a broad collection of applications, including image recognition, Natural language processing (NLP), and anomaly detection. In contrast, the

capability of DL concerning getting CPS from cyber attacks has yet to be thoroughly investigated. The challenges lie in how CPS behaves can be complicated, and the information produced by these frameworks can be high-dimensional and nonlinear. In addition, cyber attacks on CPS can be unpretentious and hard to recognize, making it trying to develop powerful techniques to distinguish them.

In this context, for detecting cyber attacks in CPS using DL is to develop and evaluate DL models that can effectively detect anomalies in CPS data and identify cyber attacks in a fast and reliable manner. The models should be able to handle high-dimensional and nonlinear data, be robust against various types of attacks, and provide interpretable results to support the decision-making process.

6.2. Research methodology

This research study depends on the classification of network traffic. Preprocessing, feature extraction, classification, and performance analysis are some of the phases in the categorization of network traffic. The procedures are described below:

1) Dataset Input and Preprocessing: The dataset was gathered from an official KDD source. According to the examination of the KDD training and test sets, there are approximately 78 percent and 75 percent of duplicate data in each set, respectively [39, 40]. Due to the abundance of unnecessary records in the training set, the learning algorithm may favor the most prevalent records. Therefore, it is vital to prohibit the sporadic and dangerous recordings. The networks may suffer from these recordings. Since repeated records are present in the test set, approaches with enhanced detection rates on regular records promote the generation of biased findings. Additionally, this study labels the records of the whole KDD training and test sets using 21 trained machines and an analysis of the complexity level of records in the KDD dataset. So, for each record, 21 projected labels are provided. The KDD dataset is a collection of unbalanced data that requires preprocessing before data cleansing.
2) FE and Classification: For FE, the CNN model is used, and for classification, the LSTM is used. By merging short-term and long-term thinking, LSTM networks utilize a micro-gate control and offer a partial solution for gradient disappearance. There are three explicit frameworks in it: An input gate, an output gate , and a forget gate. LSTM may add or delete information from the cell state using a structure known as a gate. The information from the cell state that should be discarded as decided in the first phase. The forget gate layer serves to accomplish this goal. The forget gate outputs a vector between zero and one after reading the hidden state of the previous instant $h_(t-1)$ and the current input data x_t. This vector's value, which ranges from zero and one, indicates the degree to which information is rejected or reserved in the cell state c_t. A number of zero means that all information is rejected, but a value of one means that all information is retained.

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

The second phase seeks to quantify how much new information has been introduced to the cell state. This involves two steps. Using $h_(t-1)$ and x_t, this process first determines which information needs to be updated by an input gate operation then, a new candidate cell state is obtained through the atanh layer using $h_(t-1)$ and x_t.

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

$$z = tanh(W.[h_{(}t - 1), x_t] + b) \tag{3}$$

Next, the cell state is updated as follows:

$$c_t = z_f \odot c_{(}t - 1) + z_i \odot z \tag{4}$$

The output value must be determined as the last step. The portion of the cell state that will be the output dependent on the inputs $h_{(}t - 1)$ and x_t is chosen [40] after the cell state has been changed. In order to get the judgement conditions, the input must travel through a sigmoid layer known as the output gate. After that, the cell state must be sent via the tanh layer to obtain a vector between negative one ad one. By multiplying this vector by the analytical requirements met by the output gate, the output is finally produced.

$$z_o = \sigma(W_O.[h_{(}t - 1), x_t] + b_0) \tag{5}$$

$$h_t = z_o \odot tanh(c_t) \tag{6}$$

The forget gate is represented by z_i in the formula above, the input gate by z_f and the output gate by z_o. In addition, z represents the input through a tanh layer. This is referred to as a candidate cell state as well. Last but not least, $\odot$ denotes a multiplication function of the associated matrix elements. The categorization process uses the CNN model. The input of one layer flows into the output of the following layer. Learning environments can be unsupervised, supervised, or semi-supervised. A representation learning technique can be used to define DL. Algorithms for representation learning optimize to discover the best practical method to represent the data. DL does not require separating the FE and classification operations since the model automatically extracts the features while being trained. Its four main layers are the completely connected layer, pooling layer, activation function layer, and convolutional layer [41, 42]. In Figure 6, a general CNN architecture is displayed.

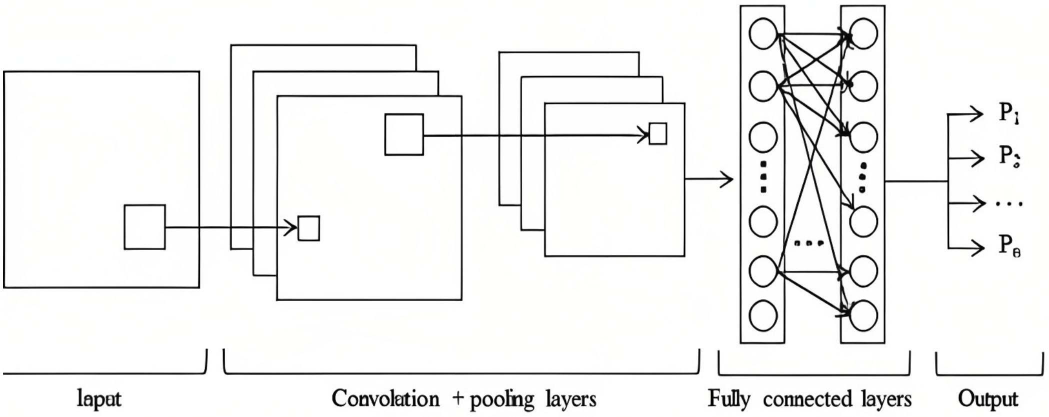


Figure 5. A CNN architecture.

- Convolutional Layer: The convolution layer is where CNN gets its name. To extract the input feature map in this layer, a number of mathematical procedures are carried out.
- Pooling Layer: Following the convolution layer, the pooling layer is often applied. In this layer, the convolution layer's output matrix size is condensed. Although different sized filters can be used in the pooling layer, the most common size is a 2×2 filter. This layer supports the usage of a number of functions, including max pooling, average pooling, and L2-norm (Euclidean norm) pooling. The biggest value in the sub-windows is chosen, and this value is then moved to a new matrix to perform max pooling.
- Activation Layer: In ANN, the activation function creates a curvilinear connection between the input and output layers. The network's performance is affected. The network may learn nonlinearly thanks to the activation function. CNN commonly utilizes the nonlinear Rectified Linear Unit (ReLU) activation function, while other activation functions include linear, sigmoid, and hyperbolic tangent. (1) does not change values greater than zero, but it does change values less than zero into zero.

$$f(x) = \begin{cases} 0, & \text{if } x < 0. \\ x, & \text{otherwise.} \end{cases}$$

- Fully Connected Layer: After convolution, pooling , and activation processes are complete, the final generated matrix is sent as an input into the fully connected layer. In this layer, recognition and classification are carried out. The hybrid DL algorithm for NTC is shown as algo 1.

6.3. Optimization in ML/DL

Optimization plays a crucial role in ML and DL models, as it determines the concert and precision of the models. Optimization refers to the process of finding the optimal set of parameters for a given model that minimizes the error among the predicted and actual values. The goal of optimization is to find the best set of parameters that leads to the best generalization performance on new, unseen data.

To tune the model parameters, traditional optimization methods like gradient descent and its derivatives are widely used in ML. It can be difficult to execute these optimization techniques on complicated models, such deep neural networks, because of their high dimensionality and nonlinear nature.

As an example of an optimization algorithm used commonly by DL, consider stochastic gradient descent (SGD) with momentum. These optimization techniques are crucial for training big, complicated DL models since they are made to handle the enormous dimensionality and nonlinearity of deep neural networks. The effectiveness and precision of DL models are significantly impacted by the optimization techniques used in the models. In addition to determining how quickly and effectively the models reach the ideal set of parameters, they also have an impact on how well the models generalize to fresh, untested data.

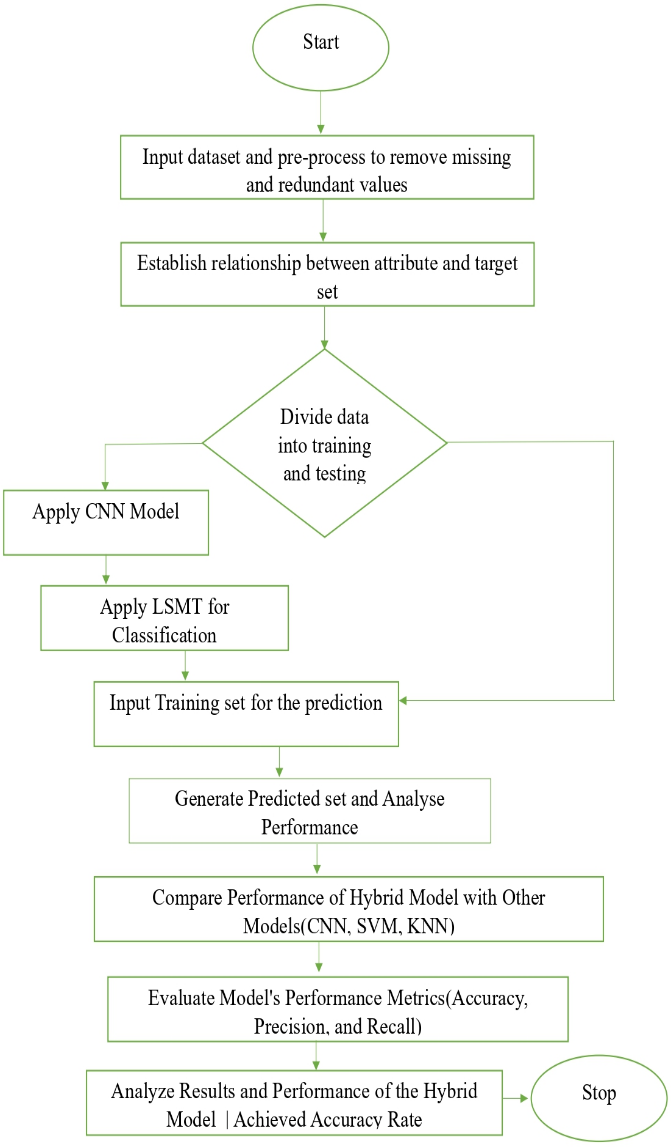
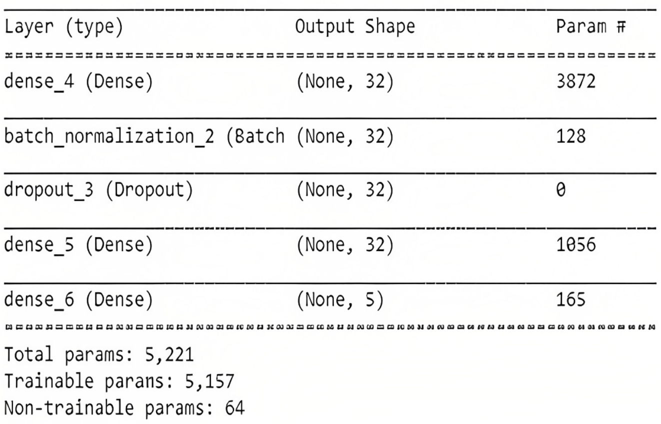

Start
Input dataset and pre-process to remove missing and redundant values
Establish relationship between attribute and target set
Divide data into training and testing
Apply CNN Model
Apply LSMT for Classification
Input Training set for the prediction
Generate Predicted set and Analyse Performance
Compare Performance of Hybrid Model with Other Models(CNN, SVM, KNN)
Evaluate Model's Performance Metrics(Accuracy, Precision, and Recall)
Analyze Results and Performance of the Hybrid Model | Achieved Accuracy Rate
Stop

Figure 6. Proposed flowchart.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 32)                3872
_________________________________________________________________
batch_normalization_2 (Batch (None, 32)                128
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_5 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_6 (Dense)              (None, 5)                 165
=================================================================
Total params: 5,221
Trainable params: 5,157
Non-trainable params: 64
```

Figure 7. Proposed model.

6.4. Choice of optimization algorithm

For DL models, the choice of an optimization technique depends on a number of variables, including the model's complexity, the volume of training data, the available computing power, and the desired convergence speed. Some of the most popular optimization strategies for DL models are as follows:

- **SGD:** The model parameters are informed by the simple and efficient optimization procedure through the negative gradient of the loss function with respect to the parameters. Due to its computational efficiency and ability to handle huge datasets and complicated models, SGD is frequently utilized in DL models.
- **Momentum SGD:** A moving average of the gradients is used in the SGD version known as momentum SGD to give a more steady update direction. Momentum SGD is often used in DL models to help avoid getting stuck in local minima, and it is particularly useful for optimizing deep neural networks.
- **Adagrad:** It adapts the learning rate to the parameters, based on the historic gradient information. Adagrad is commonly used in DL models to help avoid overfitting, and it is particularly useful for sparse data.
- **Adadelta:** Adadelta is a more recent optimization algorithm that is similar to Adagrad, but it practices a moving average of the squared gradients, rather than the historical gradients, to adapt the learning rate. Adadelta is commonly used in DL models because it is computationally efficient and it has good convergence properties.
- **Adam:** Adam is a widely used algorithm that combines the ideas from Adagrad and momentum SGD. Adam uses an adaptive learning rate that adapts to the historical gradient information, and it incorporates a moving average of the gradients to provide a more stable update direction.

```
Algorithm 1 Hybrid Deep Learning for NTC
Data Preparation:
  1: Select KDD dataset for NTCD.
  2: Preprocess data for deep learning $D_{preprocessed}$.
  3: Split dataset into training and validation sets $D_{train}, D_{validation}$
  4: Define hybrid deep learning model $M_{hybrid}$ = CNN+ LSTM
Model Training and Validation:
  5.Train hybrid model using training dataset $M_{hybrid}, D_{train}$.
  6.Validate model using validation dataset $M_{hybrid}, D_{validation}$.
  7.Monitor and record training and validation loss ($L_{train}, L_{validation}$) and accuracy ($A_{train}, A_{validation}$) for each epoch.
Model Assessment:
  8. Compare performance of hybrid model with other models (e.g., CNN, SVM, KNN).
  9. Evaluate model's performance metrics (accuracy A, precision P, recall R)
Result Analysis and Conclusion:
  10.Analyze results and performance of the hybrid model ($Results_{hybrid}$)
  11.Document achieved accuracy rate ($A_{hybrid}$)
  12.Draw conclusions on model's reliability ($Conclusion_{reliability}$)
Note:
  D,
  $D_{preprocessed}, D_{train}, D_{validation}$ represent datasets. $M_{hybrid}$ represents the hybrid deep learning model.
  $L_{train}, L_{validation}, A_{train}, A_{validation}$ represent loss and accuracy values.
  $Results_{hybrid}, A_{hybrid}, Conclusion_{reliability}$ represent analysis outcomes
```

7. Experimental results and performance evaluation

7.1. KDD dataset

KDD Cup 1999 dataset is an extensively used benchmark dataset for testing IDSs. The dataset was created by processing the TCPdump network traffic and was collected by the Massachusetts Institute of Technology (MIT) Lincoln Laboratory. The KDD Cup 1999 dataset consists of a set of preprocessed network traffic logs that have been labeled as either normal or malicious. Commonly, the dataset is used to evaluate the combination of IDSs and ML algorithms for detecting cyber attacks.

This dataset is extensively utilized due to its size and labeled data availability. The dataset contains approximately 4 million instances of 41 distinct categories of attacks, making it an invaluable tool for testing and evaluating IDSs. It is important to note, however, that the KDD Cup 1999 dataset is dated and may not accurately reflect the types of cyber attacks that are prevalent today. Nonetheless, it remains a valuable benchmark dataset for evaluating IDS and ML algorithms for cyber attack detection. In the past, both RNNs and CNNs have been used with the KDD Cup 1999 dataset for intrusion detection in CPSs.

RNNs are often used for intrusion detection in CPS because they are well-suited for processing sequential data such as network traffic logs. RNNs can learn patterns and dependencies in sequential data and make predictions based on these patterns.

CNNs are additionally utilized for interruption recognition in CPS since they are appropriate for handling grid-structured data such as image data. CNNs can learn designs in image data and do computations based on these patterns. Whether to involve an RNN or a CNN for interruption recognition in CPS relies upon the information being utilized, and the issue being settled. For instance, assuming that the issue is distinguishing irregularities in network traffic logs, an RNN might be more suitable. If the issue is to identify peculiarities in image data, a CNN might be more appropriate [43]. Generally, both RNNs and CNNs have been utilized with progress for interruption location in CPS, and the choice of calculation will rely upon the exact issue being tackled and the classification of information being used.

The sort of dataset doesn't decide if CNN or RNN ought to be utilized. The decision of which model to utilize relies upon the issue you are attempting to tackle and the kind of information you have. For the model, if the KDD dataset is in a text design, for example, network log information, an RNN might be a decent decision. RNNs are appropriate for handling consecutive information, like time series or text information, and can learn examples and conditions in the data. On the other hand, if the KDD dataset is in a framework-organized design, for example, image data, a CNN might be a definitive decision. CNNs are appropriate for grid-structured data and can learn designs in the information by utilizing convolutional filters. In general, both RNNs and CNNs have been used successfully for intrusion detection in CPS and the choice of algorithm will depend on the specific problem being solved and the type of data being used [44].

7.2. Different cyber-attacks

Benign, Denial of Service (DoS), remote-to-local (R2L), and user-to-root (U2R) are all categories of cyber attacks that target computer systems and networks. These categories are typically used to categorize different types of attacks based on their goals, methods, and impact. These categories provide a useful way of categorizing different types of cyber attacks, and are commonly used by researchers, security professionals, and law enforcement agencies to describe and understand different types of cyber threats.

1) **Benign:** "Benign" refers to something that is not harmful or dangerous. In the background of cybersecurity and CPS, benign refers to network traffic, exercises, or information that are not noxious or planned to harm. For instance, harmless traffic could be a primary HTTP demand for a web page or a transfer of specific information between devices, while malicious traffic could be a cyber assault like a Distributed Denial of Service (DDoS) assault or a phishing endeavor. Safety efforts in CPS aim to identify and prevent malicious or destructive activities and permit harmless activities to continue typically.
2) **DoS:** It is a type of cyber attack that goals to make a network reserve or a website unreachable to its envisioned users. The attackers do this by devastating the embattled network or system with a huge quantity of traffic, making it unable to handle the requests and resulting in a DoS to the planned users. This kind of assault can be innervated from a solitary computer or numerous computers simultaneously, where it is expressed as a DDoS assault. The objective of the assaults is to upset the standard working of the designated framework and cause hassle to its users.

Protecting against DDoS attacks is a central issue for CPSs, as it can disturb their typical functioning and truly harm them in many cases.

3) **PROBE:** In the context of cybersecurity, "probe" refers to a type of cyber attack that aims to gather information about a target system, network, or website. Probing can take many forms, such as pinging a target to see if it is online, attempting to access known vulnerabilities, or attempting to log into a system with a brute-force attack. The goal of the attacker is to gain information that can be used to plan and execute a more advanced attack in the future. Probes can also be used for reconnaissance purposes, such as gathering information about a target's network and systems, before launching a full-scale attack. To protect against probes, CPSs and networks should have robust security measures in place, including firewalls, IDSs, and regular software updates to fix known vulnerabilities.
4) **Remote to Local (R2L):** It is a kind of cyber attack that targets computer systems and networks. Here, the attacker gains access to a remote system and then exploits vulnerabilities to escalate their privileges, eventually gaining administrative-level control of the system. This can be accomplished through various means, such as exploiting software vulnerabilities, cracking passwords, or tricking a user into installing malware. Once the attacker has administrative control, they can perform a range of malicious actions, like installing supplementary malware, stealing sensitive information, or using the compromised system as a launching pad for further attacks. To protect against R2L attacks, it is significant to contrivance robust sanctuary measures, such as firewalls, intrusion detection systems, and regular software updates, as well as educating users about safe computing practices.
5) **User to Root (U2R):** It is a kind of cyber attack that aims at computer systems and networks. Here, the attacker starts by compromising a user account, either through social engineering, exploiting software vulnerabilities, or cracking passwords. Once the attacker has access to a user account, they can then use that access to escalate their privileges and eventually gain administrative-level control of the system. This kind of assault can bring about critical damage, as the assailant can perform a scope of dreadful activities, for example, interfacing with malware, stealing sensitive data , or using the compromised framework as a take-off platform for additional attacks. To safeguard against U2R attacks, it is influential to have contraption vigorous asylum measures , like firewalls, interruption location frameworks , and standard programming refreshes, as well as teaching teach users about safe processing rehearses. Also, it is vital to dissect users' movements to identify suspicious behavior and respond rapidly to security incidents.

7.3. NTC

NTC involves examining network traffic information and arranging it into various classes or types depending on its characteristics. This is a basic task for network security as it helps distinguish and relieve cyber attacks and other security threats. NTC includes the utilization of ML and DL strategies to automatically investigate and order network traffic given its protocol, payload, source , and destination.

A few types of NTC incorporate distinguishing web traffic, email traffic, peer-to-peer (P2P) traffic, and streaming traffic. The classification can be performed at various levels of the network protocol stack, including the application, transport, network, and connection layers.

Effective NTC is essential for securing CPS, which relies on network communication for their operation. By analyzing the network traffic in a CPS, it is possible to identify potential security threats and take appropriate measures to mitigate them.

It is possible to implement network traffic categorization using both ML and DL models. NTC may be done using ML models like K-nearest neighbours (KNN), Random Forest, SVM, and NB. On the other hand, DL models like LSTM, RNN, and CNN can be used for the same problem. The particular needs of the application and the complexity of the data will determine which model is best.

8. Results and discussions

This research work is based on NTC using DL models. The proposed model is the hybrid model which is the combination of CNN and LSTM. The LSTM model is used for the classification and the CNN model is used for the FE. The Dataset is KDD is used for the simulation results and performance metrics accuracy, precision and, recall are used to test model reliability.

As shown in Figure 8, the hybrid DL model training loss and validation loss is shown versus Epoch values. The training loss and validation loss is approximately 0.2 on the 30th Epoch.

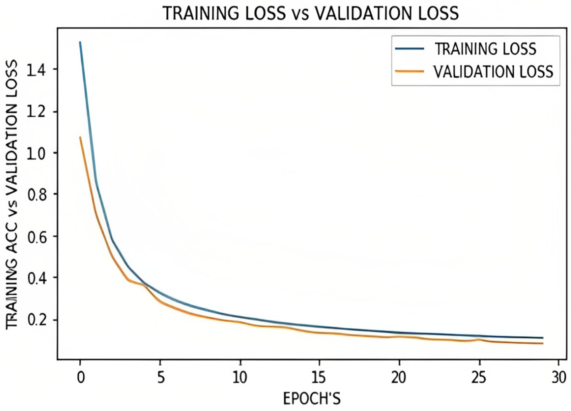


Figure 8. Model training loss.

As shown in Figure 9, the training and validation accuracy is shown in this figure versus Epoch values. The training and validation accuracy is achieved up to 97 percent for the NTC.

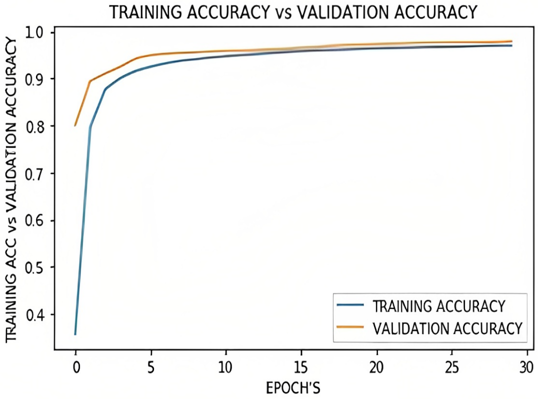


Figure 9. Training and validation accuracy.

Table 3. Performance analysis.

Model	Accuracy	Precision	Recall
SVM	88.16 Percent	88 Percent	88 Percent
KNN	76.15 Percent	76 Percent	76 Percent
CNN	90.14 Percent	90 Percent	90 Percent
Proposed	97.18 Percent	97 Percent	97 Percent

As shown in Figure 10, the performance of the proposed hybrid DL model is compared with CNN, SVM and KNN. It is analyzed that the proposed model achieves maximum accuracy of 97% for NTC.

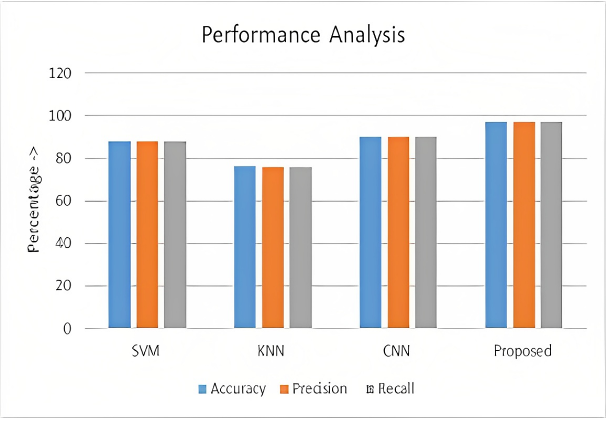


Figure 10. Performance analysis.

9. Conclusions

In this study, we have proposed a DL model for NTC, leveraging the power of CNN for FE and LSTM for classification. We evaluated our model's performance using the KDD dataset and achieved an impressive accuracy rate of 97 percent, surpassing the results of traditional methods like CNN, SVM, and KNN. As we look forward to the future of research in the field of DL and NTC, there are several promising avenues to explore. One compelling direction is the concept of adaptability through incremental learning. As network traffic patterns continually evolve, developing models that can adapt and learn from new data in real time will be crucial. This could lead to more robust and efficient classification systems. Moreover, enhancing the interpretability of DL models in NTC is another intriguing research area. The deployment of explainable AI (XAI) methods can help us understand the decision-making process of our models, making them more transparent and trustworthy. Interpretable models are essential for security applications and for gaining insights into how and why specific traffic is classified in a certain way.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. Guo, M. Cui, C. Hou, G. Gou, Z. Li, G. Xiong, et al., Global-aware prototypical network for few-shot encrypted traffic classification, in *2022 IFIP Networking Conference (IFIP Networking)*, (2022), 1–9. https://doi.org/10.23919/IFIPNetworking55013.2022.9829771

2. S. Stryczek, M. Natkaniec, Internet threat detection in smart grids based on network traffic analysis using lstm, if, and svm, *Energies*, **16** (2023), 329. https://doi.org/10.3390/en16010329

3. H. Liu, B. Lang, Network traffic classification method supporting unknown protocol detection, in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, (2021), 311–314. https://doi.org/10.1109/LCN52139.2021.9525009

4. A. Barnawi, S. Gaba, A. Alphy, A. Jabbari, I. Budhiraja, V. Kumar, et al., A systematic analysis of deep learning methods and potential attacks in internet-of-things surfaces, *Neural Comput. Appl.*, **2023** (2023), 1–16. https://doi.org/10.1007/s00521-023-08634-6

5. A. Yadav, S. Gaba, H. Khan, I. Budhiraja, A. Singh, K. K. Singh, Etma: Efficient transformer-based multilevel attention framework for multimodal fake news detection, *IEEE Trans. Comput. Soc. Syst.*, **2023** (2023), forthcoming. https://doi.org/10.1109/TCSS.2023.3255242

6. R. Moreira, L. F. Rodrigues, P. F. Rosa, R. L. Aguiar, F. de Oliveira Silva, Packet vision: a convolutional neural network approach for network traffic classification, in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, (2020), 256–263. https://doi.org/10.1109/SIBGRAPI51738.2020.00042

7. K. Lin, X. Xu, Y. Jiang, A new semi-supervised approach for network encrypted traffic clustering and classification, in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, (2022), 41–46. https://doi.org/10.1109/CSCWD54268.2022.9776310

8. J. Zhao, X. Liu, Q. Yan, B. Li, M. Shao, H. Peng, Multi-attributed heterogeneous graph convolutional network for bot detection, *Inf. Sci.*, **537** (2020), 380–393. https://doi.org/10.1016/j.ins.2020.03.113

9. P. Singh, G. Bathla, D. Panwar, A. Aggarwal, S. Gaba, Performance evaluation of genetic algorithm and flower pollination algorithm for scheduling tasks in cloud computing, in *International Conference on Signal Processing and Integrated Networks*, (2022), 139–154. https://doi.org/10.1007/978-981-99-1312-1_12

10. S. Gaba, I. Budhiraja, V. Kumar, S. Garg, G. Kaddoum, M. M. Hassan, A federated calibration scheme for convolutional neural networks: Models, applications and challenges, *Comput. Commun.*, **192** (2022), 144–162. https://doi.org/10.1016/j.comcom.2022.05.035

11. A. Aggarwal, S. Gaba, J. Kumar, S. Nagpal, Blockchain and autonomous vehicles: Architecture, security and challenges, in *2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), IEEE*, (2022), 332–338. https://doi.org/10.1109/CCiCT56684.2022.00067

12. Y. Wang, X. Yun, Y. Zhang, C. Zhao, X. Liu, A multi-scale feature attention approach to network traffic classification and its model explanation, *IEEE Trans. Network Serv. Manage.*, **19** (2022), 875–889. https://doi.org/10.1109/TNSM.2022.3149933

13. J. Zhao, M. Shao, H. Wang, X. Yu, B. Li, X. Liu, Cyber threat prediction using dynamic heterogeneous graph learning, *Knowl. Based Syst.*, **240** (2022), 108086. https://doi.org/10.1016/j.knosys.2021.108086

14. Q. Ma, W. Huang, Y. Jin, J. Mao, Encrypted traffic classification based on traffic reconstruction, in *2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD), IEEE*, (2021), 572–576. https://doi.org/10.1109/ICAIBD51990.2021.9459072

15. Y. Zeng, Z. Qi, W. Chen, Y. Huang, Test: an end-to-end network traffic classification system with spatio-temporal features extraction, in *2019 IEEE International Conference on Smart Cloud (SmartCloud), IEEE*, (2019), 131–136. https://doi.org/10.1109/SmartCloud.2019.00032

16. A. Aggarwal, S. Gaba, S. Nagpal, A. Arya, A deep analysis on the role of deep learning models using generative adversarial networks, in *Blockchain and Deep Learning: Future Trends and Enabling Technologies, Springer*, (2022), 179–197. https://doi.org/10.1007/978-3-030-95419-2_9

17. S. Nagpal, A. Aggarwal, S. Gaba, Privacy and security issues in vehicular ad hoc networks with preventive mechanisms, in *Proceedings of International Conference on Intelligent Cyber-Physical Systems: ICPS 2021, Springer*, (2022), 317–329. https://doi.org/10.1007/978-981-16-7136-4_24

18. G. Aceto, D. Ciuonzo, A. Montieri, A. Pescapé, Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges, *IEEE Trans. Network Serv. Manage.*, **16** (2019), 445–458. https://doi.org/10.1109/TNSM.2019.2899085

19. M. Lotfollahi, M. J. Siavoshani, R. S. Hossein Zade, M. Saberian, Deep packet: A novel approach for encrypted traffic classification using deep learning, *Soft Comput.*, **24** (2020), 1999–2012. https://doi.org/10.1007/s00500-019-04030-2

20. G. Aceto, D. Ciuonzo, A. Montieri, A. Pescapé, MIMETIC: Mobile encrypted traffic classification using multimodal deep learning, *Comput. Networks*, **165** (2019), 106944. https://doi.org/10.1016/j.comnet.2019.106944

21. M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, J. Lloret, Network traffic classifier with convolutional and recurrent neural networks for Internet of Things, *IEEE Access*, **5** (2017), 18042–18050. https://doi.org/10.1109/ACCESS.2017.2747560

22. J. Li, V. S. Sheng, Z. Shu, Y. Cheng, Y. Jin, Y. F. Yan, Learning from the crowd with neural network, in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, (2015), 693–698. https://doi.org/10.1109/ICMLA.2015.14

23. X. Y. Zhang, G. S. Xie, C. L. Liu, Y. Bengio, End-to-end online writer identification with recurrent neural network, *IEEE Trans. Human Mach. Syst.*, **47** (2016), 285–292. https://doi.org/10.1109/THMS.2016.2634921

24. X. Shi, H. Qi, Y. Shen, G. Wu, B. Yin, A spatial–temporal attention approach for traffic prediction, *IEEE Trans. Intell. Transp. Syst.*, **22** (2020), 4909–4918. https://doi.org/10.1109/TITS.2020.2983651

25. Y. Saadna, A. Behloul, An overview of traffic sign detection and classification methods, *Int. J. Multimedia Inf. Retr.*, **6** (2017), 193–210. https://doi.org/10.1007/s13735-017-0129-8

26. D. Kaur, A. Anwar, I. Kamwa, S. Islam, S. M. Muyeen, N. Hosseinzadeh, A Bayesian deep learning approach with convolutional feature engineering to discriminate cyber-physical intrusions in smart grid systems, *IEEE Access*, **11** (2023), 18910–18920. https://doi.org/10.1109/ACCESS.2023.3247947

27. A. Aldweesh, A. Derhab, A. Z. Emam, Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues, *Knowl. Based Syst.*, **189** (2020), 105124. https://doi.org/10.1016/j.knosys.2019.105124

28. J. Bhardwaj, J. P. Krishnan, D. F. L. Marin, B. Beferull-Lozano, L. R. Cenkeramaddi, C. Harman, Cyber-physical systems for smart water networks: A review, *IEEE Sens. J.*, **21** (2021), 26447–26469. https://doi.org/10.1109/JSEN.2021.3121506

29. M. S. Akhtar, T. Feng, Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time, *Symmetry*, **14** (2022), 2308. https://doi.org/10.3390/sym14112308

30. D. D. Godsey, Y. H. Hu, M. A. Hoppa, A Multi-layered Approach to Fake News Identification, Measurement and Mitigation, in *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC)*, (2021), 624–642. https://doi.org/10.1007/978-3-030-73100-7_45

31. Y. Jang, N. Kim, B. D. Lee, Traffic classification using distributions of latent space in software-defined networks: An experimental evaluation, *Eng. Appl. Artif. Intell.*, **119** (2023), 105736. https://doi.org/10.1016/j.engappai.2022.105736

32. A. V. Jain, Network traffic identification with convolutional neural networks, in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), IEEE*, (2018), 1001–1007.

33. S. Dong, Multi class svm algorithm with active learning for network traffic classification, *Expert Syst. Appl.*, **176** (2021), 114885. https://doi.org/10.1016/j.eswa.2021.114885

34. Y. Guo, G. Xiong, Z. Li, J. Shi, M. Cui, G. Gou, Combating imbalance in network traffic classification using gan based oversampling, in *2021 IFIP Networking Conference (IFIP Networking), IEEE*, (2021), 1–9. https://doi.org/10.23919/IFIPNetworking52078.2021.9472777

35. F. Al-Obaidy, S. Momtahen, M. F. Hossain, F. Mohammadi, Encrypted traffic classification based ml for identifying different social media applications, in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), IEEE*, (2019), 1–5. https://doi.org/10.1109/CCECE.2019.8861934

36. X. Ren, H. Gu, W. Wei, Tree-rnn: Tree structural recurrent neural network for network traffic classification, *Expert Syst. Appl.*, **167** (2021), 114363. https://doi.org/10.1016/j.eswa.2020.114363

37. W. Liu, C. Zhu, Z. Ding, H. Zhang, Q. Liu, Multiclass imbalanced and concept drift network traffic classification framework based on online active learning, *Eng. Appl. Artif. Intell.*, **117** (2023), 105607. https://doi.org/10.1016/j.engappai.2022.105607

38. Y. Pan, X. Zhang, H. Jiang, C. Li, A network traffic classification method based on graph convolution and lstm, *IEEE Access*, **9** (2021), 158261–158272. https://doi.org/10.1109/ACCESS.2021.3128181

39. C. Gijón, M. Toril, M. Solera, S. Luna-Ramírez, L. R. Jimenez, Encrypted traffic classification based on unsupervised learning in cellular radio access networks, *IEEE Access*, **8** (2020), 167252–167263. https://doi.org/10.1109/ACCESS.2020.3022980

40. X. Jing, J. Zhao, Z. Yan, W. Pedrycz, X. Li, Granular classifier: Building traffic granules for encrypted traffic classification based on granular computing, *Dig. Commun. Networks*, **2022** (2022), forthcoming. https://doi.org/10.1016/j.dcan.2022.12.017

41. S. Ahn, J. Kim, S. Y. Park, S. Cho, Explaining deep learning-based traffic classification using a genetic algorithm, *IEEE Access*, **9** (2020), 4738–4751. https://doi.org/10.1109/ACCESS.2020.3048348

42. J. Zhang, J. Zhou, N. Zhou, Network traffic classification method based on subspace triple attention mechanism, in *2022 3rd International Conference on Information Science, Parallel and Distributed Systems (ISPDS), IEEE*, (2022), 312–316. https://doi.org/10.1109/ISPDS56360.2022.9874195

43. A. S. Iliyasu, H. Deng, Semi-supervised encrypted traffic classification with deep convolutional generative adversarial networks, *IEEE Access*, **8** (2019), 118–126. https://doi.org/10.1109/ACCESS.2019.2962106

44. L. K. Ramasamy, F. Khan, M. Shah, B. V. V. S. Prasad, C. Iwendi, C. Biamba, Secure smart wearable computing through artificial intelligence-enabled internet of things and cyber-physical systems for health monitoring, *Sensors*, **22** (2022), 1076. https://doi.org/10.3390/s22031076

AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)