Literature Review: Deep Learning for Wireless Low Resource Community Networks

Michael Gamsu

ABSTRACT

This project aims to build a deep-learning traffic classifier to optimise Quality of Service (QoS) and traffic engineering in low-resource community networks. These low resources pose significant challenges due to limited computational resources and network data encryption. Traditional methods often struggle to effectively classify encrypted traffic, making deep learning approaches increasingly relevant for improving QoS in community networks. Deep learning techniques like Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and Stacked Autoencoders (SAEs) show the potential to achieve high classification accuracy. However, the suitability of deep learning architecture depends on the type of classification and the trade-off between accuracy and computational complexity. Therefore, analysis of the various deep learning architectures used for traffic classification and assessing the potential and constraints specific to low-resource community networks is required. Furthermore, deep learning offers potential avenues for traffic classification in community networks; this review will examine the remaining vital considerations, including data quality, privacy, efficiency and model applicability.

KEYWORDS

Deep Learning, Community Networks, Traffic Classification, Neural Networks.

1 INTRODUCTION

Network traffic classification categorises network traffic into different classes through applications (Facebook, Gmail, YouTube) or protocols (TCP, UDP). Thus, it is essential in network traffic engineering, quality of service (QoS), and Network Security. Real-time traffic classification can assist network engineers and operators in prioritising the flow of different network applications for users on the network. Thus improving the quality of a service. Furthermore, it can provide insights into patterns and variables that identify data. In addition to QoS, traffic classification can detect intrusions and anomalies in the network. Thus providing community networks with safer connectivity while remaining cost-effective. In recent years, deep learning has significantly advanced in various fields. This literature review will provide context for network traffic classification, the dataset used, and the different approaches to processing and analysing the data.

1.1 Project Aims

The main objective is to investigate the effectiveness of various deep learning architectures for network traffic classification, considering the constraints of CPU, memory, and time typical in low-resource community networks. The primary focus is to compare different deep learning architectures, such as Stacked Autoencoders, Multilayer Perceptron (MLP), and hybrid architectures combining Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to capture spatial and temporal patterns in the network traffic data to increase QoS for real-time traffic classification. What are the key research questions that we will answer?

(1) How do different deep learning architectures such as Stacked Autoencoders, Multi-layer Perceptron, and hybrid architectures combining convolutional neural networks (CNNs) and Recurrent Neural networks provide the highest classification accuracy with the performance and data constraints in low resource community networks?
(2) How do the deep learning techniques perform under varying CPU and memory constraints?
(3) How effectively do hybrid architectures combining CNNs and RNNs learn spatial and temporal patterns in network traffic data?

Therefore, our project aims to use deep learning to classify traffic in low-resource community networks. By prioritizing critical applications and optimizing resource allocation, we can improve the Quality of Service (QoS) for users in these networks. This will help bridge the digital divide and promote digital inclusion in underserved areas. Additionally, an opportunity presents itself to further academic research in network management and optimization.

2 QUALITY OF SERVICE (QOS)

Quality of Service (QoS) is the comprehensive evaluation of telecommunications services based on four measurable parameters: latency, jitter, bandwidth, and packet loss. These parameters collectively determine the service's ability to meet its users' stated and implied needs, ensuring satisfactory performance and reliability in delivering data packets across the network [19]. In contrast, Quality of Experience (QoE) is measured beyond the QoS and focuses on the user's perception of the overall quality of telecommunication services. Thus, it provides further insights into network requirements' perceived satisfaction and preferences. QoE is based on user satisfaction with content access and application utility [4].

3 COMMUNITY NETWORKS

Community Networks offers a global solution to extend internet access to rural and low-resource areas. These networks are locally developed and managed, with communities establishing their network infrastructure and access points to connect to the internet at large [16]. Their infrastructure operates with neutrality, freedom, and openness principles despite being technically non-centralised and self-managed.

Consequently, challenges related to community networks are the administration, scalability, and ensuring service quality that remains prevalent in the deployment and maintenance of these networks [15]. Community Networks aim to close the digital divide by providing cheaper connectivity in rural and developing areas. The constraint of low-cost and inexpensive hardware within these networks is pertinent to traffic classification. Classification models

typically deployed on routers in such networks will encounter limitations in processors and memory. Thus, the main challenge to overcome is keeping computational complexity low while achieving high classification accuracy [26].

Unlike traditional networks, community networks have diverse protocols and traffic patterns that affect traffic capacity. Packet classification becomes more challenging on community networks due to the decentralised infrastructure that applications may use advanced port obfuscation techniques of embedding their information in well-known protocols' packets and randomisation of ports that evade ISPs' controlling processes [13]. The limitations of skills within a community and the quality and condition of the hardware vary, affecting network stability and performance. Thus providing no guarantee of QoS or connectivity to community networks [15].

Despite these challenges, traffic classification plays a crucial role in improving QoS, network security, and resource allocation within community networks. By prioritizing critical applications, such as VoIP or emergency services, administrators can ensure smoother user experiences and effective congestion management. Additionally, targeted security measures can be implemented to protect against threats like malware and DDoS attacks. Efficient resource allocation based on demand helps optimize network resources and minimize wastage, ultimately leading to cost savings for community network operators.

3.1 Support Vector Machine (SVM)

The Support Vector Machine (SVM) is a popular machine-learning model known for its ability to classify data quickly and accurately while using minimal memory. It is beneficial for binary classification problems, which seek to distinguish between two classes by finding the maximum-margin hyperplane [17]. This hyperplane is a straight line that divides the feature space's two classes. In cases where linear separation is impossible, a kernel function converts the data into a higher dimension[17].

One of the key benefits of the SVM algorithm is its ability to allow specific values to fall outside of the hyperplane, a phenomenon known as soft margin. This helps prevent over-fitting and ensures the model can generalise well to new data. When classifying multiple classes, a one-versus-rest method is typically used to separate each class from the others [26].

4 RELEVANT DEEP LEARNING APPROACHES

This section serves as an introduction to various deep-learning techniques frequently employed for network traffic classification. It explains each technique and gives an overview of their application in classifying network traffic observed in previous studies. Deep learning is considered a subset of machine learning, and the focus is specifically on training deep neural networks with multiple layers to learn complex patterns and representations directly from raw data. This section will focus on Multi-layer Perceptrons (MLP), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Stacked AutoEncoders (SAE) and hybrid models. Therefore, careful consideration of the evaluation metrics and training data is required to navigate the complexities of the various architectures.

4.1 Evaluation Metrics

Evaluation metrics serve as essential to assessing the performance of deep learning models. The key metrics typically used to evaluate the result are accuracy, precision, and recall, which offer valuable insights into the model's classification success [18]. Accuracy, precision and recall, although commonly used, have been known to fall due to population prevalence and label bias [18]. Precision and recall provide a fine-grained view of the model's ability to minimise false positives and false negatives. Additionally, the F1-score is an important metric that provides a balanced mean of precision and recall. The focus of the F1 score is to handle the imbalances of datasets. The F1-score provides a detailed view of a model's performance [18]. While particular research is not constrained by computational efficiency, this project will need metrics that analyse the computational efficiency, memory usage, and model parameters essential to the context of low-resource community networks. These metrics will provide an analysis of the model's performance and whether it is feasible to work in the scenario with limited CPU and memory resources.

4.2 Training Data Requirements

Effective deep-learning models rely heavily on the quality and diversity of training data used for traffic classification in community networks with limited resources. Training data that reflects the various network traffic patterns is crucial to developing robust models. To achieve this, a comprehensive dataset with samples spanning diverse applications, protocols, and network conditions is necessary. Accurate labelling of training data requires attention to detail to minimise errors and improve model efficacy. Data labelling is usually carried out by DPI tools given their high accuracy [5]. The preprocessing of encrypted traffic varies depending on the classification method preference (typically port, flow, and packet). Ma et al. [14] recommend preprocessing at the packet level when requirements for fine-grained characteristics in application classification. The methodology included five-tuple clustering, removal of MAC and IP addresses, packet splicing and bye conversion and normalisation, and dividing datasets. This preprocessing is critical to refine the training data and optimise its suitability for use in deep learning models. However, data privacy and security concerns must be considered, emphasising the need for anonymisation and encryption measures to protect sensitive user information. By adhering to these rigorous training data requirements, researchers and practitioners can develop robust and reliable deep-learning models tailored for traffic classification in low-resource community networks, potentially revolutionising network management paradigms[26].

4.3 Multi-Layer Perceptron

Multi-layer perceptron (MLP) is the foundation of the neural network structure. MLP architecture consists of an input layer, one or more hidden layers and an output layer. Each layer contains neurons, and neurons in adjacent layers are connected. The neurons compute a weighted sum of inputs, adjust for bias, and apply a non-linear activation function. The training data forms the input to the first layer, while each neuron computes the weighted sum of the outputs from the neurons of the previous layer with the

corresponding edge's weight thereafter. The output is used for predictions. The networks determines the optimal weight parameter values to ensure accuracy between the prediction and actual values. This adjustment is done through backpropagation [2].

However, MLP models are rarely used for traffic classification

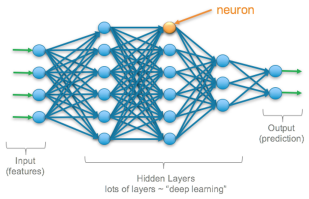

Figure 1: MLP

because of the computational complexity and low accuracy compared to other deep learning techniques [2]. However, MLP is also considered a benchmark used to compare other models. According to Aceto et al., MLP performs relatively poorly in traffic classification using one or two hidden layers compared to other models [3].

4.4 Convolutional Neural Networks

Convolutional neural networks are a more advanced deep learning model compared to MLP. The main difference is the input data is received through convolutional layers [13]. Convolutional layers use a set of kernel kernels to extract spatial patterns from input data. Furthermore, CNNs use pooling to determine if something exists by reducing the input by taking summary statistics of input values that are close together. Thus allowing the model to learn to shift-invariant features [9]. CNNs reduce the number of learnable parameters by reusing the same kernels across the entire input. Parameter sharing and sparse interactions further decrease the computational complexity compared to MLP [9]. CNNs are typically used in image recognition and object detection by two-dimensional CNNs (2D-CNNs). However, adjusting the model to one-dimensional CNN (1D-CNNs) uses filters to capture spatial patterns in the 1D vectors [26]. 1D-CNNs have shown significant success in traffic classification when using packet data. The convolutional layer recognises the spatial relationship between adjacent bytes or packets, resulting in effective traffic classification [22]. Furthermore, CNNs are shift-invariant for traffic classification as patterns are detected even if different sections are moved [20]. Two-dimensional CNN (2D-CNNs) filters use image recognition that captures two-dimensional features. In terms of traffic classification, 2D-CNNs require a transformation of the data into a 2D image. Aceto et al. [1] and Wang et al. [25] show that 1D-CNNs outperform 2D-CNNs as 1D-CNNs are better for sequential data [11].

4.5 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a neural network that stores RNNs designed for sequential data processing, where the current state depends on the current input and previous input. RNN achieve this by containing loops in the hidden layers to allow neurons to retain temporal information [21]. LSTM is a popular variant of conventional RNNs that addresses training issues such as gradient vanishing (which occurs during the backpropagation algorithm over gradient computation over multiple stages). Thus impacting the ability of RNNs to capture long-term dependencies in sequential data [21]. LSTM architecture incorporates a gated mechanism (including input forget and output gates) to regulate the flow of data within the model.

Wang et al. suggest that LSTMs are used instead of the conventional RNN because LSTMs are able to capture the long-term dependencies [24]. This is particularly useful when classifying flow-based traffic, as it can learn the temporal patterns between the individual packets. However, LSTM architecture is inherently sequential, which makes it hard to train and run in parallel.

4.6 Stacked Autoencoder (SAE)

Autoencoders (AE) are a type of neural network architecture designed to reconstruct input data at the output layer, typically with fewer neurons in the hidden layers than the input and output layers. They are commonly utilised for dimensionality reduction and feature extraction tasks and also serve to initialise weights for other deep learning architectures [10]. Stacked Autoencoders (SAE) take this concept further by stacking multiple autoencoders together, where the output of one autoencoder serves as the input for the next [10]. This stacked architecture allows for more complex representations to be learned.

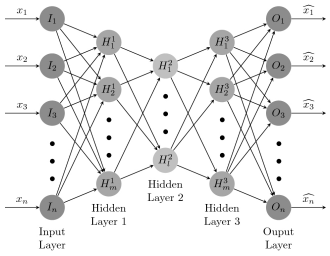

Figure 2: Autoencoder

4.7 Hybrid Models

The hybrid model integrates different techniques or models within a single framework to leverage their complementary strengths for improved performance. Bayat et al. [3] aimed to develop a deep learning model that network classifiers without decryption. This study utilises a hybrid model comprising Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs). GRU is a variation of the RNN that addresses the vanishing gradient problem and handles variable-length sequences. CNNs capture spatial dependencies in time series features and extract patterns. By combining both architectures, the model discerns intricate patterns in encrypted traffic. It learns from diverse features such as packet sequences, payload size, and inter-arrival times, addressing performance issues on specific inputs and high bias. Additionally, an ensemble method enhances robustness and accuracy by combining separate classifiers for each feature.

5 RELATED WORK

Network traffic classification is a crucial aspect of network management and optimisation. In recent years, deep learning and machine learning techniques have been extensively explored in this domain to enhance the accuracy and efficiency of traffic classification. The primary methods for performing traffic classification are deep packet inspection (DPI), port-based inspection, flow classification and packet classification. DPI inspects packets using headers and application signatures, but the challenges are complexity and privacy. Port-based inspection relies on TCP/UDP headers, yet it struggles with port obfuscation. Flow classification groups packets by characteristics, while packet classification dissects individual packets. These specific traffic classification decides how raw traffic is segmented into multiple discrete units [1]. This section will delve into these two types of traffic classification and review the deep-learning architectures that have been applicable in recent years.

5.1 Deep Packet Inspection (DPI)

Deep Packet Inspection (DPI) is a fine-grained level technique for inspecting data packets. Both packet headers and application signatures are used to categorise network traffic. While historically relying on statistical methods and lightweight string matching algorithms [8]. DPI has evolved into packet examining packet headers and application signature contents. Despite DPI's high accuracy in traffic classification, it is computationally expensive and has ethical privacy concerns due to it requiring access to user data. Furthermore, DPI struggles with encrypted network traffic, impacting its effectiveness, such as secure communication channels and virtual private networks (VPNs) [12]. Lotfollahi et al. [13] proposed a model that uses two different deep learning methods for DPI. They used 1D-CNN and SAE for application identification and traffic characterisation tasks. The research found that the 1D-CNN and SAE models achieved high F1 scores for application identification and traffic characterisation tasks. The 1D-CNN achieved a weighted average F1 score of 0.98 for application identification, while the SAE had a slightly low score of 0.95. In traffic characterisation, the 1D-CNN achieved an F1 score of 0.93, while SAE scored slightly less than 0.92 [13].

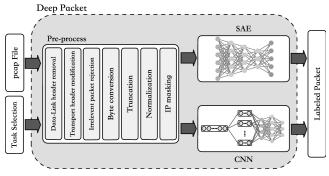

Figure 3: DPI tool kit

5.2 Port Based Inspection

Port-based inspection is considered the oldest and most widely recognised method for traffic classification via port number [7]. Port-based classification utilises the information from the TCP/UDF headers to extract port numbers associated with specific applications. The extracted port numbers are compared to the Internet Assigned Numbers Authority (IANA) for TCP/UDP applications to classify traffic. However, challenges with port obfuscation and network address translation (NAT) have diminished its accuracy. Thus, more sophisticated traffic classification approaches like flow and packet classification effectively classify [13].

5.3 Flow Classification

Aceto et al. [1] defines a flow as comprising all packets sharing the same 5-tuple: source IP, source port, destination IP, destination port, and transport-level protocol. Previous literature discusses various approaches to flow classification. One approach involves directly extracting raw data, in the form of bytes, from specific packets within the flow [25]. Alternatively, another method involves extracting raw data directly from the flow itself, focusing solely on the initial N bytes while disregarding individual packets [24]. The third approach utilises time series data derived from individual packets, incorporating packet sizes, directions, and inter-arrival times [21]. Lastly, flow statistics represent the fourth method for extracting data from a flow: statistics on the packets, packet sizes and arrival times[26]. This approach uses a lot more packets from flow to minimise the variance. However, Rezaei et al. [21] do not recommend this methodology if the aim is for real-time classification. The structure of the raw data extracted from flows made LSTM and CNNs useful for traffic classification. These deep learning models find long and short-term temporal connections in the data. Wang et al. [25] 1D-CNN and a 2D-CNN to classify flows from the first 785 or 1000 bytes of the flow. These bytes were transformed into a 1D vector and 2D image, respectively. The 1D-CNN achieved an accuracy of 91.25%, while the 2D-CNN had an accuracy of 90% [26]. As previously discussed, it is expected that 1D-CNN is better for sequential data. Zeng et al. [27] investigated various deep learning architectures for flow-based traffic classification. They examined three models: CNN, LSTM, and SAE. The results of all three models achieved approximately 99% classification accuracy on encrypted data. The LSTM model demonstrated slightly better accuracy than the SAE and marginally worse accuracy than the CNN. This indicates the viability of LSTM for flow-based traffic classification in the context of low-resource community networks. Bayat et al. [3] utilised a CNN-GRU (RNN) hybrid model for flow classification, training on bidirectional flow classification of raw data and inter-arrival times. The combined CNN-GRU for packet, payload, and inter-arrival time sequences achieve 77.1%, 78.1%, and 63.2%, respectively, compared to the CNN-based if 62.4%. This combination effectively captured spatial and temporal features. Thus demonstrating the possibility of a hybrid of CNNs and RNNs. However, computational costs associated with this model were not explored and could provide challenges.

5.4 Packet Classification

Packet classification is a highly detailed method for individual packet analysis, making it more precise than flow classification [13]. However, when dealing with sequential data, packet-by-packet analysis becomes redundant. On the other hand, deep learning can be trained using packet data, allowing it to learn high-dimensional data. Lotfollahi et al. [13] used the first 1480 bytes of the IP payload and IP header as input but concealed IP addresses due to limited

host and server usage [13]. Unfortunately, this limitation may have resulted in less reliable outcomes from their research. Meanwhile, Wang et al. [23] ignored the IP header and used identical data input to Lotfollahi et al. Lotfollahi et al. [13] researched the effectiveness of convolutional neural networks (CNN) and support vector machines (SAE) for packet-level classification [13]. They trained the SAE and equipped it with a soft-max layer to classify traffic. The model achieved F1 scores of 95% and 92% for application classification and traffic characterisation, respectively, just below the 1D-CNN [26]. Using a 1D-CNN, individual packets were classified into classes, achieving an F1 score of 98% for application classification and 93% for traffic categorisation. In a related study on application categorisation, Wang et al. [23] compared the MLP, CNN, and SAE models and MLP achieved an F1 score of 96.5%. The limitation of this MLP was that it was a small model with only two hidden layers and six neurons each, which could have reduced performance. In comparison, CNN and SAE achieved an F1 score of 98.4% and 98.8%, respectively [26].

6 DISCUSSION

Deep learning techniques for traffic classification require selecting appropriate models based on the nature of the data and the classification task. MLPs serve as a foundational structure, but CNNs and RNNs are more suitable for traffic classification. Hybrid models integrate different deep learning architectures, offering an approach for leveraging the complementary strengths of individual models. Deep learning techniques have shown significant advancements over DPI and port-based inspection approaches. It explored the limitations of DPI dealing with encrypted data traffic. In comparison, port-based inspection accuracy has diminished due to port obfuscation and network address translation (NAT). The analysis highlights the potential of deep learning techniques, particularly CNNs, LSTMs and SAE, as they all achieve high classification accuracy. CNNs are ideal to capture spatial dependencies in traffic data. LSTMs handle long-term dependencies in the sequential data. SAE can be used for dimensionality reduction or feature extractions. Selecting the appropriate deep learning technique depends on various factors, including understanding the strengths and limitations of each technique and the trade-off between accuracy and computational complexity chosen. Therefore, this section will review several challenges and considerations for deep learning approaches for traffic classification in community networks.

6.1 Data Gathering and Selection

The fundamental problem with the study of traffic classification is that there is no standardised dataset to asses models [21]. Researchers tend to use their dataset or publicly available datasets accumulated to meet the specifics of their research, e.g., the dataset used for intrusion detection. Therefore, comparing the different models and the conclusions drawn from these studies is difficult. It is vital that data gathering is completed in a controlled environment to ensure accurate labelling and sampling for all classes. Deep learning models are limited in their accuracy of application. The challenge identified with raw data is that it has high dimensionality. This refers to a dataset with a large number of features or variables. The higher dimensionality results in increased computational complexity. Thus, using the raw data for traffic classification on low-resource community networks may be challenging. Preprocessing will be required on the packet or flow to ensure that the computational complexity is minimised. Furthermore, Preprocessing allows the model to learn the dynamic structure of the flow, which can lead to an increase in predictive capabilities [6]. Deep learning models, especially those with large numbers of parameters like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), can be computationally intensive. In low-resource community networks with limited computational resources such as CPU, memory, and power, other options may be available than deploying complex models. Therefore, there is a trade-off between model complexity and resource constraints. Hybrid architectures or lightweight models may need to be explored to balance accuracy with computational efficiency. One challenge in applying deep learning models for traffic classification in community networks is ensuring their generalisation and adaptability to diverse network environments. Models trained on one network may not perform well on another due to variations in traffic patterns, hardware configurations, and network topologies. Therefore, evaluating model performance across different network scenarios and possibly fine-tuning or retraining models to ensure their effectiveness in real-world deployments is essential.

7 CONCLUSION

In conclusion, deep learning techniques have potential applications for traffic classification in low-resource community networks. Thus, it addresses critical considerations and explores possible research directions for deep learning models for network traffic classification. Thus, by leveraging neural network architectures such as CNNs, RNNs, and hybrid models, it is possible to improve the accuracy and efficiency of traffic classification while addressing the constraints of CPU, memory, and power typical in community networks. Additionally, it can be utilised in networking management to increase QoS and overall user experience on these networks. However, challenges such as model complexity, generalizability, and privacy considerations must be carefully addressed to successfully deploy deep learning-based traffic classification systems in real-world community network environments.

REFERENCES

[1] Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, and Antonio Pescapé. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE Transactions on Network and Service Management*, 16(2):445–458, 2019.

[2] Tom Auld, Andrew W Moore, and Stephen F Gull. Bayesian neural networks for internet traffic classification. *IEEE Transactions on neural networks*, 18(1):223–239, 2007.

[3] Niloofar Bayat, Weston Jackson, and Derrick Liu. Deep learning for network traffic classification. *arXiv preprint arXiv:2106.12693*, 2021.

[4] Hector Fabio Bermúdez, JL Arciniegas, and E Astaiza. Estado del arte de los métodos de evaluación de qoe y entornos de emulación para el servicio de video en redes lte. *Entre Ciencia e Ingeniería*, 10(20):66–75, 2016.

[5] Tomasz Bujlow, Valentín Carela-Español, and Pere Barlet-Ros. Independent comparison of popular dpi tools for traffic classification. *Computer Networks*, 76:75–89, 2015.

[6] Zhitang Chen, Ke He, Jian Li, and Yanhui Geng. Seq2img: A sequence-to-image based approach towards ip traffic classification using convolutional neural networks. In *2017 IEEE International conference on big data (big data)*, pages 1271–1276. IEEE, 2017.

[7] Alberto Dainotti, Antonio Pescape, and Kimberly C Claffy. Issues and future directions in traffic classification. *IEEE network*, 26(1):35–40, 2012.

[8] Michael Finsterbusch, Chris Richter, Eduardo Rocha, Jean-Alexander Muller, and Klaus Hanssgen. A survey of payload-based traffic classification approaches. *IEEE Communications Surveys & Tutorials*, 16(2):1135–1156, 2013.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[10] Jonas Höchst, Lars Baumgärtner, Matthias Hollick, and Bernd Freisleben. Unsupervised traffic flow classification using a neural autoencoder. In *2017 IEEE 42Nd Conference on local computer networks (LCN)*, pages 523–526. IEEE, 2017.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[12] Ying-Dar Lin, Chun-Nan Lu, Yuan-Cheng Lai, Wei-Hao Peng, and Po-Ching Lin. Application classification using packet size distribution and port association. *Journal of Network and Computer Applications*, 32(5):1023–1030, 2009.

[13] Mohammad Lotfollahi, Mahdi Jafari Siavoshani, Ramin Shirali Hossein Zade, and Mohammdsadegh Saberian. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Computing*, 24(3):1999–2012, 2020.

[14] Xiuli Ma, Wenbin Zhu, Jieling Wei, Yanliang Jin, Dongsheng Gu, and Rui Wang. Eetc: An extended encrypted traffic classification algorithm based on variant resnet network. *Computers & Security*, 128:103175, 2023.

[15] LM Martínez-Cervantes and R Guevara-Martínez. Community networks and the quest for quality. 2020.

[16] Panagiota Micholia, Merkouris Karaliopoulos, Iordanis Koutsopoulos, Leandro Navarro, Roger Baig Vias, Dimitris Boucas, Maria Michalis, and Panayotis Antoniadis. Community networks and sustainability: a survey of perceptions, practices, and proposed solutions. *IEEE Communications Surveys & Tutorials*, 20(4):3581–3606, 2018.

[17] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[18] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[19] ITUT Recommendation. E. 800, definitions of terms related to quality of service. *International Telecommunication Union's Telecommunication Standardization Sector (ITU-T) Std*, 2008.

[20] Shahbaz Rezaei and Xin Liu. How to achieve high classification accuracy with just a few labels: A semi-supervised approach using sampled packets. *arXiv preprint arXiv:1812.09761*, 2018.

[21] Shahbaz Rezaei and Xin Liu. Deep learning for encrypted traffic classification: An overview. *IEEE communications magazine*, 57(5):76–81, 2019.

[22] Justine Sherry, Chang Lan, Raluca Ada Popa, and Sylvia Ratnasamy. Blindbox: Deep packet inspection over encrypted traffic. In *Proceedings of the 2015 ACM conference on special interest group on data communication*, pages 213–226, 2015.

[23] Pan Wang, Feng Ye, Xuejiao Chen, and Yi Qian. Datanet: Deep learning based encrypted network traffic classification in sdn home gateway. *IEEE Access*, 6:55380–55391, 2018.

[24] Wei Wang, Yiqiang Sheng, Jinlin Wang, Xuewen Zeng, Xiaozhou Ye, Yongzhong Huang, and Ming Zhu. Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE access*, 6:1792–1806, 2017.

[25] Wei Wang, Ming Zhu, Jinlin Wang, Xuewen Zeng, and Zhongzhen Yang. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In *2017 IEEE international conference on intelligence and security informatics (ISI)*, pages 43–48. IEEE, 2017.

[26] Shane Weisz. Network tra ic classification using two-dimensional convolutional neural networks for community networks.

[27] Yi Zeng, Huaxi Gu, Wenting Wei, and Yantao Guo. $deep - full - range$: a deep learning based network encrypted traffic classification and intrusion detection framework. *IEEE Access*, 7:45182–45190, 2019.