Literature Review : Deep Learning for Wireless Community Networks

Boitumelo Mokoka
University of Cape Town
MKKBOI005@myuct.ac.za

ABSTRACT

Community networks are community-led projects which aim to establish and manage accessible network infrastructure. These networks address the need for telecommunications and internet connectivity in areas where these services are not available, they present a feasible alternative to commercial internet service providers. Traffic classification and the optimisation therein is an effective way to ensure quality of service in a network and traditionally networks rely on statistical methods to achieve this. Research has shown deep learning techniques to be particularly effective at this task and in this literature review we will be presenting the different implementations of deep learning, namely Multi-Layer perceptrons, Convolutional Neural networks and hybrid architectures, their advantages and disadvantages and evaluating their application to network traffic classification. we will be reviewing the existing literature on this topic with the aim of applying these techniques to community networks.

Keywords

Deep Learning, Networks, Traffic Classification, Neural Networks

1 INTRODUCTION

Network traffic classification is utilized in many ways in the pursuit of quality internet connectivity. Quality of service(QoS) is a methodology where the resources of a network are allocated by their corresponding priority and is used to improve network performance [23]. Accurately classified traffic allows for efficient resource management and the enhancement of many QoS services [13]. Classified traffic may be used by QoS devices in the form of different forwarding priorities, this is used to implement bandwidth optimization where critical packets are delivered first. This is especially effective in networks where resources are extremely limited. Traffic classification can also be utilised by security devices which implement traffic policing, intrusion detection and malware detection [16, 24]. Traffic classification's pertinence to quality internet services has resulted in extensive research into the topic and the various methodologies therein.

Classical approaches have applied statistical, traditional machine learning methods (such as k-nearest neighbour and random forests) and port-based methods to traffic classification but the constantly increasing volume and complexity of network traffic has resulted in the evolution of traffic classification methods. This has taken shape in the form of many deep learning methodologies, these deep learning methods cater for the increased complexity and increasing level of encryption in network data. The use of port numbers in traffic classification has declined in efficacy due to the increase in disguised port numbers in newer applications. The general increase in demand for privacy has also resulted in an increase in encryption in network traffic, this becomes a challenge to classify as encrypted data is pseudo-random [14]. Where classical techniques used port numbers to classify traffic, deep learning techniques are focused on finding patterns in data packets, this allows newer methods to handle more complex and encrypted data [16]. Traditional techniques requires experts to select features before training where deep learning techniques require less data preprocessing, allowing the trained models to be applied to more generalized and unprocessed datasets [14]. This is especially useful for traffic classification where data patterns are always changing. These are just a couple of examples demonstrating why deep learning has prevailed over traditional methods, in recent research, especially in the context of network traffic classification.

Research has shown Deep learning to be the most accurate approach to classifying network traffic. [14] demonstrated a stacked auto encoder and one-dimensional Convolutional Neural Network outperforming all similar works on the "ISCX VPN-nonVPN" traffic dataset, which has been used as the basis of many important papers demonstrating machine learning methods1. Deep learning models' performance is directly correlated to the amount of data they are trained on and this characteristic makes deep learning architectures applicable in our context as networks tend to have extremely large amounts of raw data[19].

The aim of the research project and this literature review is to tackle the task of applying network traffic classification to low resource networks. Our goal is to formulate a lightweight deep learning model which is accurate enough to be utilised for QoS features in a community network. Deep Learning will be approached with the aim of learning spatial and temporal patterns in the data. We will be providing the context of the research in the background section, with an overview of community networks, data collection/classification approaches and a brief explanation of the different deep learning architectures. We will then discuss and compare the performance and computational complexity of the different models in the context of network traffic classification. Lastly we will summarize the key conclusions.

2 BACKGROUND

The most popular approaches to traffic classification as well as an overview of community networks are presented here in the context our classification task.

2.1 Community Networks

Community networks are networks setup and maintained by a community for internet connectivity, this is also known as "bottom-up networking", these networks are decentralized systems and often set up using a mesh topology. Often citizens will pool resources to establish and operate open IP-based networks which provide local and internet networking, this has been the basis of hundreds of community networks operating across the globe [3]. Often times large ISPs will lack the financial incentive to establish internet connectivity in rural areas which is where community networks come in, they take the dependency away from large corporations for internet connectivity. Community networks have played a vital role in the battle against the digital divide by servicing under-serviced areas [1].

Community networks are funded and managed by a local community or by a non-profit organisation which is why their resources are more constrained than traditional internet service providers. Hence the networks are often established with simple and low cost software and hardware [3]. When working with limited resources on a network it is important that auxiliary tasks such as traffic classification are optimised. Deep learning models often tend to be computationally expensive which weakens their suitability for low-resource networks, which is why the computational complexity of the different deep learning methods will be analysed in this review and research project, in great detail. This project and literature review will be using a dataset extracted from the iNethi community network based in Ocean View, South Africa.

2.2 Traffic Classification

There are multiple approaches to network traffic classification, in this section we will provide an overview of the most important and justify our use of package-based classification as an approach. Several protocols such as packet marking, integrated services (IntServ) and Multiprotocol Label Switching (MPLS) have been proposed for traffic differentiation but these proved to be too complex to implement and deploy, hence they never gained popularity [11]. Traditional Traffic classification methods used statistical analysis on information provided the headers of packets travelling through the network however a recent increase in the complexity of internet traffic has mitigated their effectiveness.

2.2.1 ***Deep Packet inspection*** . DPI is a technique where packets are inspected at a granular level, packet headers and application signatures are used to classify traffic, technological advancements have allowed for advanced packet inspections that check both the head and contents of a packet. Traditionally this approach was implement using statistical methods in combination with light-weight string matching algorithms [13]. DPI has been show to reach high accuracy levels however it comes with a heavy computational costs, privacy issues (as it requires access to user data) and struggles to deal with encrypted network traffic [7].

2.2.2 ***Port based classification*** . Alternatively port based classification is one of the simplest and oldest methods of traffic classification. Well known port numbers are assigned by the Internet Assigned Numbers Authority (IANA) for certain applications and protocols [15]. This classification approach extracts the port numbers from the TCP/UDP headers and compares them to their assigned IANA port numbers [14] Nowadays applications use dynamic port allocation , some networks hide port numbers for privacy reasons and mobile applications tunnel their traffic through protocols resulting in port obfuscation. For these reasons Port-based Classification has become unreliable and recent research has shown packet-based methods combined with machine learning methodologies to be the most effective approach.

2.2.3 ***Package-based Classification***. A payload based approach involves the use of patterns and characteristics from packet attributes to categorize the network traffic. Patterns are derived from the application layer of the packets and these are used to distinguish protocols from each other. Basic features such as packet sizes, arrival-times, number of bytes, bit rate are extracted from the application layer without the need to inspect the contents of the package. This approach is utilized in unsupervised learning methods which efficiently extract the relationships among these features [5]. The only drawback being that when protocols are updated the patterns need to be re-learned in order to be effective, this shortfall is addressed by code reuse and the CNN's ability to be re-trained for new activities (described below).

2.3 Deep Learning Techniques

In this section we will provide a brief overview on the different deep learning techniques which we will be utilising. This is just an introduction to the different approaches, further below in the application section we will discuss how these techniques have been applied to network traffic classification and in the discussion section we will evaluate their respective strengths and weaknesses in the context of our task.

Deep Learning is a term for Neural Networks with several hidden layers, this complexity is necessary to learn patterns in high-dimensional data [17]. Deep Learning is classified as a subset of Machine Learning and its ability to improve proportionally to the amount of data it is trained on has made it especially useful when learning patterns in extremely large datasets [19]. Some define Deep Learning as an AI functionality that mimics the human brain's processing of data, this is because DL networks are constructed using *neurons* (nodes) which are connected by weighted links [14]. It must be noted that Deep Learning trains models using a large number of parameters which translates to a longer training time, however this shortfall is balanced out by the model's efficiency and speed during testing.

2.3.1 Multilayer Perceptron. The Multilayer Perceptron (MLP) is the most popular and heavily researched neural network model to date [12]. Research on the MLP machine learning methodology traces back to 1960 with its ability to learn linear separation being showcased by Widrow et. al [20] in 1960 which is why it is considered the foundation of deep neural networks. It is classified as a Supervised deep learning technique as noted in 1, This means that the technique requires large sets of labeled data in order to train an accurate model. It consists of input and output layers which respectively receive input data and produce decisions/predictions based on this input. There are typically many hidden layers between these two and this is where the computational complexity arises [19]. The exact number of hidden layers and the number of perceptrons at each layer is up to the implementation of the neural network and its exact architecture is often determined at the hyper-parameter-tuning phase of training. The power of the MLP comes from its activation function, any non-linear function can be used for this except polynomial functions . The activation function allows non-linear transformations of input at each neuron in the network and this is the key to its ability to classify multidimensional data with a high accuracy. Popular activation functions include ReLu (Rectified Linear Unit), Sigmoid and Tanh (hyperbolic tangent) functions [12].

The learning process is based on the approach of minimizing errors between the network's outputs and the desired outputs, this process is enforced using a backpropogation algorithm. The process of training a MLP is as follows: First network weights are initialized to random values, then at each epoch (training iteration) a signal is propagated from the input layer to the output layer, during this process the outputs of hidden neurons are calculated using its input connections, the weights associated with these and the activation function. An error value is calculated at the output using the difference between the predicted and expected values and this error is back-propagated through the hidden layers to update the weights of each node. A few hyper parameters such as the learning rate (η) are used to determine by how much each weight changes, the weights are adjusted over a number of iterations in order to minimize the error. If the training error increases at any points the weights are reset to their best performance values and the process is started again. This process is repeated until a stopping condition is reached, this condition may be a desired accuracy or a specific number of epochs. Auxiliary methods such as an additional learning rate for each weight have been proposed to accelerate the training process [12]. A shortfall of the MLP is that training is quiet slow, complex problems often require tens of thousands of training iterations [12].

2.3.2 *Stacked Autoencoders.*

An auto-encoder Neural Network is an unsupervised deep learning approach (1) that aims to reconstruct the input at the output layer while minimizing the reconstruction error. An autencoder employs dimensionality reduction by using hidden layers that are smaller than the input and output layers while employing a softmax classifier in the output layer for classification. Using this approach the autoencoder attempts to learn the compressed representation of the dataset, this technique is generally used for feature extraction. The stacked autoencoder (SAE) is a more complex procedure where several auto-encoders are combined in a layer-wise method so the output of each one is the input of the successive layer [14]. This stacking allows the network to learn more complex data in a hierarchical fashion with each layer capturing increasingly complex features. The training process of SAEs involves separately training each AE layer using unsupervised learning, the different AE layers are then combined and fine-tuned using supervised learning [5]. The SAE is often combined with different neural network approaches such as CNNs and LSTMs, these auxiliary networks are placed either in between different AE modules or at the end of the last AE for the task of classification.

2.3.3 *Hybrid Architectures.*

Hybrid architectures combining different types of deep learning networks in many ways. Conventionally convolutions Neural Networks and Recurrent Neural networks are combined in order to effectively learn patterns in spatial and temporal patterns. Below we will explore the structure of each of these deep learning methods and the methods with which they are formed into hybrid architectures.

The Convolutional Neural Network is a supervised learning method (as noted in 1) that is effective at feature extraction due to its architecture. CNNs consist of multiple convolution and pooling stages which achieve feature extraction, this gives it the ability to learn directly from input without the need for human feature extraction [22]. The lower convolutional levels extract basic features while the higher levels extract more complex and abstract features, exploiting the features extracted by the lower layers. The pooling functionality exists between these convolutional layers, it reduces the dimensionality of the outputs and consequentially makes the features location-insensitive (meaning the network focuses more on the presence on the feature than its precise location within the data). The classification takes place using a feed-forwards neural network after the convolution and pooling stages, this process makes use of a softmax function which returns the output in terms of probabilities. This enhanced design has made CNNs extremely popular in the field of image processing and its ability to learn of of raw data without human intervention makes it more practical than other deep learning models. An advantage of CNNs is that existing models can be re-trained for new activities, so when network traffic protocols change it is not a challenge to retrain the existing network for the same purpose [5]. It must be noted that the features extracted by the deep layers of the CNN do not make a lot of sense to human observers which is why this method is considered a black-box method [14], this makes manual adjustments for fine-tuning the model challenging.

Recurrent Neural Networks are especially useful for learning time-series (temporal) data. They utilise "memory" and loops which allows them to use information from previous inputs unlike other learning methods which treat inputs and outputs as independent factors [19]. RNNs were designed to capture sequential data where outputs depend on the preceding values and this feature makes it especially useful in network traffic classification for contextual feature extraction and for understanding sequences of packets without implementing a flow-based classification approach. There are many derivetives and improved versions of the RNN which address its shortfalls in certain areas 1. To address the gradient fragility and inability to learn long-term dependencies the long short-term memory (LSTM) was introduced, this improved model mitigated the long-term dependency problem by controlling when information is stored or removed from the model's "memory" [16].

The CNN and RNN networks are effective in their respective specialties (spatial and temporal features) but they have been demonstrated to perform much better when combined into hybrid architectures by recent research[5, 18]. One approach to implementing these architectures as hybrids is to implement them in the hidden layers of a stacked auto-encoder. [5] utilised multiple combination of these architectures in his work (discussed in the application section), using a CNN to extract spatial-features while using a LSTM recurrent network to extract temporal-features harnessing of the strengths of each model. An alternative is to implement CNN and RNN layers in sequence, in this implementation the output of CNN layers are fed as the input to RNN layers to learn spatial and temporal patterns from the data. Traditional MLP techniques such as backpropogation and dropout can be applied to this joint network in the pursuit of an optimal model.

Another widely-used approach is the fusion of different models at the decision level. Simply put, multiple models are trained independently on the same dataset and their seperate answers are combined by choosing an average answer. There are many sophisticated statistical techniques to combine models in order to obtain a more accurate prediction. Bagging (Bootstrap Aggregation) reduces the variance in classification models by extracting random samples (with replacement) from the dataset and training each model on a different subset. Once trained each model is asked to predict an answer the set of answers are combined either by using the mean or highest occurring answer. Random forests is a similar statistical approach which decorrelates the different models by using a random subset of the variables extracted from the dataset. In our context multiple models would be trained but each model would be trained on a different subset of the dataset's fields. When applied to deep learning this technique is called ensemble learning, it is effective because each model captures a different features of the data because each has a different "perspective" of the data. Random forests is effective in reducing overfitting, increasing parallelization (each network can be trained on different resources) and reduces sensitivity to noise [6].

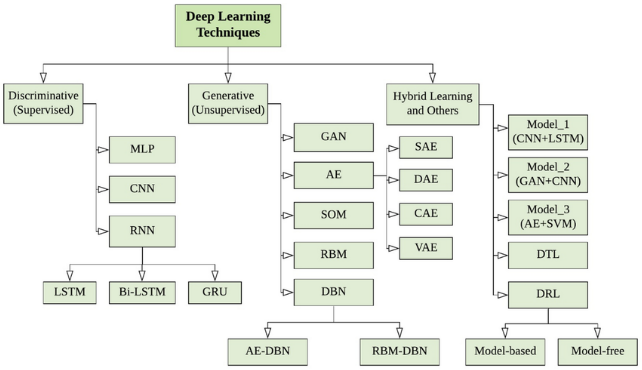

Figure 1: Basic taxonomy of Deep Learning (DL) methods [17]

3 APPLICATION

In this section we will be reviewing how the above techniques have been applied to the task of traffic classification. There have been many studies in the field of network traffic classification, however many focus on the implementation of traditional methods such as statistical and machine learning analysis.

Popular works have demonstrated the use of statistical properties in network traffic to classify flows. Bernaille et. al [4] achieved an application classification accuracy of 85% by using the size of the packages in SSL (Secure Socket Layer) encrypted connections. This is one example of a recent study that used payload-based approaches to workaround the increased encryption of modern network traffic. A shortfall of this literature was that it was only demonstrated to be effective on SSL connections, additional research into SSH and IPsec encryption would be necessary to label this work as a proof of concept. Both Zhang et. al [9] and Lin et. al [10] successfully demonstrated the effectiveness of traditional machine learning and statistical analysis to classify network traffic. Lin et. al [10] showed that pure application traffic could be utilised for the task by using a packet size distribution as an extracted feature, here the frequent packet sizes where used to classify traffic to different applications. This paper validated that the distribution of packet sizes is a good characteristic to classify applications and implemented the approach with an auxiliary *Port Association Table* which stored information on port locality. The literature demonstrated the effectiveness of combining a packet-based approach with a flow-based approach by using both package sizes and port information. They were successful in developing a classification algorithm without the need to inspect payload information and achieved a high accuracy of 96% on average with a false-negative rate of 4-5%. This method also worked well for encrypted applications like Skype. Unfortunately the work is extremely outdated and the nature of network traffic has changed drastically since these works were published, the use of port numbers has also become unreliable due to an increase in port obfuscation.

More recent and prevalent works have demonstrated the effectiveness of deep learning approaches when compared to traditional machine learning. [14] is a very recent work that has justified the deep learning approach by implementing both a Convolutional Neural Network (CNN) and Stacked Auto-Encoder (SAE). This approach outperformed all similar works on a very popular dataset 1, the "UNB ISCX VPN-nonVPN" dataset. Recent works such as [2] have also shown Deep learning algorithms to efficiently extract statistical features using raw network traffic as input further supporting the preference of deep learning models over traditional machine learning models for this application.

3.1 Dataset preprocessing

An essential step in the training of deep neural networks is the preprocessing of data, this step ensures that the data is in a format that the network can accept. This process usually consists of data collection and, in cases where neural networks require it, feature selection. The dataset we will be working with contains raw packet data (PCAP files) drawn from Wireshark. Research often utilizes popular and pre-labelled datasets, for example the extremely popular "ISCX VPN-nonVPN" dataset, however our implementation of deep learning will require some preprocessing in order to ensure effective training procedures.

There are many approaches to transforming raw datasets and a common solution is to use a Deep Packet Inspection tool to transform encrypted traffic into a labeled dataset. There are a range of free DPI modules such as nDPI which can be used to label datasets however it is noted in [16] that these methods are not effective for encrypted traffic. Another shortfall of these open-source tools is that the Neural Network's accuracy is dependent on the accuracy of the labelling tool, this problem can be minimized by utilizing neural networks that can handle raw data, such as the CNN architecture. Imbalanced training datasets lead to reduced classification performance and a simple statistical solution would be to train deep learning models on a sample of the dataset that represents each category evenly, this approach will decrease the amount of data which may hinder the accuracy of the model but it also prevents the model from overfitting to dataset-specific features while guiding it more towards traffic specific features [16].

[8] took a unique package-based approach to preprocessing their dataset. They extracted the DNS packages, converted the data into hexadecimal and then into binary form. This binary format was then transformed into a black and white image where each binary 0 and 1 represented a sequential black or white pixel. A CNN was used on this dataset and effectively extracted features and characterized protocols, this was due to the fact that CNNs are extremely proficient at processing and classifying images. This approach is worth investigating as it takes advantage of CNN's specialized architecture and feature extraction capabilities.

3.2 hyper-parameters and optimisation

Deep Learning approaches come with many hyper-parameters and the optimal configuration of these values are found using cross-validation. [5] considered over 5 million different combinations of hyper-parameters in the search of an optimal network configuration. Additionally many training techniques are utilized to improve the training process: early-stopping is a technique which stops the training process when the loss function remains unchanged for several iterations. Dropout is a methodology to reduce over fitting of a network, it works by randomly nullifying a certain portion of the nodes in the input and hidden layers at each iteration during training.

Dimension reduction protocols are used when training unsupervised models to reduce the complexity of input data and aid in extracting features by disposing of redundant data. These algorithms are traditionally utilized in the pooling layers of CNN architectures. Two popular dimensionality reduction algorithms are the t-SNE (t-distributed stochastic neighbour embedding) and PCA (Principle Component Analysis) algorithms which are both effective for identifying features in data. [8] found a combination of the T-SNE and the PCA algorithms to be the most effective for clustering features in their implementation of a CNN for network traffic classification. Although the combination did not perform much better than the standalone T-SNE algorithm, the calculation time was reduced the most when both algorithms were combined. This dimensionality reduction algorithm has been demonstrated to ensure accuracy while maintaining performance.

3.3 Deep learning models

In this section we will be over viewing recent implementations of the different deep learning methods in the context of network traffic classification.

3.3.1 *Multilayer Perceptron.*

Aceto et. al [7] investigated the applicability of MLP to network traffic classification and found MLPs to be the poorest performing model when compared to other deep learning models. This architecture is still worth investigating because it can be utilized as a hybrid or auxiliary network as demonstrated in the implementation of the Hybrid DBN-CNN-MLP hybrid network [18].

3.3.2 *Stacked Auto-Encoders.*

Dangelo et. al [5] investigated network traffic classification using self-learned spatial-temporal features using different combinations of CNNs, Recurrent network models and Stacked encoders. The dataset used contained over 1,4 million 6-dimensional features and the different models were used to categorize the traffic into 4 categories. It is also worth noting the they used Amazon Web service machines to perform the task, 4 instances with 4 vCPU Intel Broadwell E5-2686v4 @ 2.3 GHz, and 16 GB RAM were utilised. This is a relatively low-resource specification and is worth noting because of the resource constraints which we will be testing in our implementations. This implementation was achieved using a package-based technique and 6 features of each packet-flow were organized as a vector and used as the input to the networks. 6 different architectures were tested, the best performing hybrid architectures included a Stacked-CNN-LSTM-SAE-NN, a ConvLSTM-SAE-NN and a Deep-SAE-NN.

Dangelo et. al [5] outperformed many traditional machine-learning methodologies which were applied to the same dataset. The literature compared the use of traditional machine learning approaches with deep learning methods and took this a step further by comparing the traditional deep learning architectures with their respective hybrid counterparts. First a traditional machine learning method was tested against the CNN and LSTM deep learning approaches, the standard CNN and LSTM configurations performed 2% and 26% (respectively) better than the machine learning classifier but the introduction of SAE increased the performance further. Both the CNN and LSTM architectures were tested against their hybrid counterparts and they found that the introduction of SAE increased the performance significantly. The accuracy of the CNN-SAE and LSTM-SAE increased by 9% and 8% when compared to their CNN and LSTM counterparts. The configuration with the highest performance was a stacked-CNN-LSTM-SAE neural network, scoring close to 100% for every metric. This study demonstrated that the introduction of CNN and LSTM layers to a Stacked Auto Encoder improves its classification performance substantially. While the pure CNN and LSTM showed low performance in distinguishing the classes considered, the introduction of an SAE demostrated significant improvement in the models.

3.3.3 *Hybrid Architectures.*

Xue et. al [8] utilized the popular ResNet-50 CNN architecture and achieved an accuracy of 89.78% and a weighted average precision of 89.66% across the 3 protocols tested, this was the highest performing out of the 6 popular CNN architectures tested in that literature. Lotfollahi et. al [14] was able to achieve an accuracy of 98% on the "ISCX VPN-nonVPN" dataset using a one-dimensional CNN. This implementation used two CNN layers, a pooling layer, a flattening layer and a three-layered fully connected network in that order. The CNN's ability to accurately learn patterns from raw encrypted network traffic without the need for deep packet inspection was demonstrated to be substantial. It is noted that the model was not able to classify traffic which was encrypted before transmission, it was only effective in classifying traffic based on encryption patterns evident in the application layer. A shortfall of this deep learning approach is its inability to classify new classes of traffic. One suggestion is to design a multi-level classification algorithm which detects whether traffic belongs to a known class or a new "unknown" class, furthermore the ability to cluster these unknown classes using an unsupervised clustering algorithm will make them useful for future classification via human intervention.

Data fusion is the act of combining datasets from different sources to address the lack of sufficient labelled data in a single dataset, this method increases the accuracy of deep learning models trained on the dataset due to the larger volume. Decision level data fusion combines the results of independent classifiers to achieve a higher accuracy and this method is implemented using a Bayesian model s in [18]'s traffic classification approach. Here three deep learning models, a one-dimensional CNN, a deep belief network (DBN) and an MLP are used to classify the traffic in the ISCX VPN-nonVPN dataset, the results of all three models are then fused using the Bayesian fusion method to form the final decision.

Author	Model	Precision	Accuracy
Lotfollahi et. al [14]	1D CNN	0.93	0.98
Izadi et. al [18]	Hybrid DBN-CNN-MLP		0.97
Lotfollahi et. al [14]	SAE	0.92	0.95
Yamansavascilar et al. [21]	KNN		0.94

Table 1: Comparison of performance using the "ISCX VPN-nonVPN" dataset

4 DISCUSSION

4.1 Comparison of the deep learning models

It is mildly difficult to compare the performance of different different deep learning implementations because each literature uses a unique dataset. Luckily multiple works have utilised the popular ISCX VPN-nonVPN dataset for network traffic classification, making this comparison a little easier.

Firstly it must be noted that the MLP has barely been implemented for network traffic classification. The few studies found displaying its capabilities demonstrated that MLPs are the worst performing deep learning model for network traffic classification. It's poor accuracy mitigates it as an option for a standalone network traffic classifier although its relatively low computational complexity makes it a candidate to be implemented as a component in a hybrid network. Izadi et. al [18] proved this to be true with the best performing model being a combination of CNN, DBN and MLP with decision level fusion.

Implementations of the SAE architecture proved to be proficient in network traffic classification, falling just behind the CNN implimentation when tested on the ISCX VPN-nonVPN dataset (1). Dangelo et. al [5] demonstrated that SAE implementations can harness the strengths of other deep learning methodologies by integrating them as layers in the architecture. As a standalone implementation the autoencoder was sufficient but when improved and combined with CNN and MLP layers it demonstrated the best performance.

CNNs and hybrid implementations therein have been extensively studied in the context of network traffic classification and have repeatedly been demonstrated to be an extremely effective deep learning model for this task. As noted in 1 the standalone CNN and the hybrid CNN models scored the highest accuracy out of the recent works on the ISCX VPN-nonVPN dataset. CNNs are evidently powerful for the task of network traffic classification and their ability to process raw data makes them an even stronger candidate for the most effective deep learning model. Xue et. al [8] also demonstrated the ability to modify existing CNN architectures for the task of traffic classification with ease, by using the popular ResNet-50 implementation. The CNN's ability to be re-trained and reused for different tasks make it adaptable and robust, these features make it the ideal candidate for community networks.

Although it is difficult to compare the performance of the different literaature pieces reviewed in this paper due to the use of many different datasets, one dataset called the "ISCX VPN-nonVPN" was used extensively. A comparison of the performance of the different model architectures is tabulated in ??. Yamansavascilar et al. [21] used time-related features to achieve application classification. Deep Packet [14] outperformed other proposed approaches on the dataset with their CNN proving to be the best performing model.

4.2 Gaps in Current Research

It is widely noted that the need for extremely large volumes of data in order to utilize deep learning is a shortfall which reduces its applicability. The amount of required data in training affects the accuracy of classification and [18] suggests the investigation of new classification methods that are more robust and applicable to datasets with insufficient samples for network traffic classification. Their suggested approach is data fusion, the combination of different datasets from different sources. This methodology is worth investigating for our implementation of deep learning. When combined with sampling and dataset-balancing methods such as random forests and bagging these statistical techniques reduce the variability of the outcome and produce more uniform results. Another aspect is accurately labelling raw datasets in order to be used in supervised learning models, the online nDPI software has been utilized by recent literature but its inaccuracy when handling encrypted traffic may affect the performance of the neural networks trained on this data.

Attacks on machine learning methods have been widely studied, these kinds of attacks exploit vulnerabilities in machine learning models by intentionally causing misclassification. These attacks are not well researched in network traffic classification and to ensure the security and robustness of a deep learning network traffic classifier in a community network this area will need to be addressed. The complex nature of deep learning architectures makes the especially vulnerable to malicious attacks but various defence mechanisms and emerging software aims to address this issue. This is a topic that will be further addressed in later stages of the literature.

A few pieces of literature implemented impromptu simulations to test how the models will perform when implemented in a live network. A standard benchmark and simulation that mimics the conditions of a mesh/community network is necessary to benchmark the performance of the different models. Web services such as amazon allow you to benchmark performance on specific computer architectures and this is an approach worth considering for our literature. A standard format with which to compare other deep learning implementations would be ideal.

5 CONCLUSION

In this paper the implementation of a deep learning model for the task of classifying network traffic in a community network was considered, this is to be used to improved the QoS in the network. The need for an efficient yet accurate model was motivated in order to accommodate hardware and overall resources constraints. A package based approach was decided upon and various deep learning architectures were evaluated for this purpose. In the background section the context for the literature, the justification for package-based classification as an approach and an overview of the deep learning models was presented. Purely packet-based classification prevailed as the best option due to the increasingly complex nature of network traffic and the improved performance. All three deep learning approaches are heavily researched and present their own strengths and weaknesses although the best fit for the task was shown to be CNN due to its ability to process raw data effectively. In the application section the various models were evaluated using the recent research in the context of network traffic classification. Aspects such as dataset preprocessing and model optimisation were also briefly discussed in order to gain insight into the process of implementing a deep learning model. While the nDPI software is widely used to label network traffic its inaccuracy on encrypted traffic have provided motivation to explore other methods. The three model approaches were each evaluated using recent studies and implementations, these studies provided conclusive evidence and motivation for the different models. MLP were shown to be the weakest of the 3 and its lack of accuracy eliminates it as a suitable candidate for a standalone implementation however it was found to be useful when combined in hybrid architecture and for that reason it is still considered worth investigating. The SAE implementations were extremely effective and the implementation of CNN and MLP layers in between convolutions proved to be powerful improvements to the model. The different hybrid networks were also shown to be extremely accurate and the CNN model was demonstrated to be the most accurate of all in 1. Statistical techniques such as bagging and random forests are effective ways to combine different architectures and harness the strengths of each model without increasing computational complexity too much. A sequential implementation of a CNN and RNN (LSTM) network is a promising way to utilize the benefits of both but no prior implementations of this exact model in the context of traffic classification were located. Finally the Discussion section compared the different deep leraning models and explored the gaps in current research worth addressing in the literature.

REFERENCES

[1] Andrea Kavanaugh, John M. Carroll, M. B. R.-T. T. Z. D. D. R. Community networks: Where offline communities meet online. *Journal of Computer-Mediated Communication 10*, 4 (07 2017).

[2] Azab, A., Khasawneh, M., Alrabaee, S., Choo, K.-K. R., and Sarsour, M. Network traffic classification: Techniques, datasets, and challenges. *Digital Communications and Networks* (2022).

[3] Bart Braem, Christoph Barz, F. F. C. B.-H. R. L. N. J. B. P. E. A. L. K. S. P. R. B. V. A. N. B. T. I. V. i. B. M. M. A case for research with and on community networks. *ACM SIGCOMM Computer Communication Review 43* (07 2013), 68–73.

[4] Bernaille, L., and Teixeira, R. Early recognition of encrypted applications. *PAM 2007 - 8th Internatinoal Conference on Passive and Active network Measurement* (Apr 2007), 165–175.

[5] D'Angelo, G., and Palmieri, F. Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial–temporal features extraction. *Journal of Network and Computer Applications 173* (2021), 102890.

[6] Gareth James, Daniela Witten, T. H., and Tibshirani, R. *An Introduction to Statistical Learning with Applications in R.* Springer, 2023.

[7] Giuseppe Aceto, Alberto Dainotti, W. d. D., and Pescape, A. Portload: Taking the best of two worlds in traffic classification. pp. 1 – 5.

[8] Jingliang Xue, Yingchun Chen, O. L., and Li, F. Classification and identification of unknown network protocols based on cnn and t-sne. *Journal of Physics: Conference Series* (July 2020).

[9] Jun Zhang, Yang Xiang, Y. W. W. Z.-Y. X., and Guan, Y. Network traffic classification using correlation information. *IEEE Transactions on Parallel and Distributed Systems 24*, 1 (2013), 104–117.

[10] Lin, Y.-D., Lu, C.-N., Lai, Y.-C., Peng, W.-H., and Lin, P.-C. Application classification using packet size distribution and port association. *Journal of Network and Computer Applications 32*, 5 (2009), 1023–1030.

[11] M. Said Seddiki, Muhammad Shahbaz, S. D. S. G. M. P. N. F., and Song, Y.-Q. FlowQoS: QoS for the Rest of Us. In *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN'2014)* (Chicago, United States, Aug 2014).

[12] MARIUS-CONSTANTIN POPESCU, VALENTINA E. BALAS, L. P.-P., and MASTORAKIS, N. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems 8*, 7 (2009), 579–588.

[13] Michael Finsterbusch, Chris Richter, E. R., Muller, J.-A., and Hanssgen, K. A survey of payload-based traffic classification approaches. *IEEE Communications Surveys Tutorials 16*, 2 (2014), 1135–1156.

[14] Mohammad Lotfollahi, Ramin Shirali Hossein Zade, M. J. S., and Saberian, M. Deep packet: A novel approach for encrypted traffic classification using deep learning.

[15] Ola Salman, Imad H. Elhajj, A. K., and Chehab, A. A review on machine learning–based approaches for internet traffic classification. *Annals of Telecommunications 75* (06 2020), 673–710.

[16] Rezaei, S., and Liu, X. Deep learning for encrypted traffic classification: An overview. *IEEE Communications Magazine 57*, 5 (2019), 76–81.

[17] Roya Taheri, Habib Ahmed, E. A. Deep learning for the security of software-defined networks: a review. *Cluster Computing 26* (July 2023), 3089–3112.

[18] Saadat Izadi, M. A., and Rajabzadeh, A. Network trafc classifcation using deep learning networks and bayesian data fusion. *Journal of Network and Systems Management (2022)* (January 2022).

[19] Sarker, I. H. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science (2021) 2*, 420 (August 2021).

[20] Widrow, B., Hoff, M. E., et al. Adaptive switching circuits. In *IRE WESCON convention record* (1960), vol. 4, New York, pp. 96–104.

[21] Yamansavascilar, B., Guvensan, M. A., Yavuz, A. G., and Karsligil, M. E. Application identification via network traffic classification. In *2017 International Conference on Computing, Networking and Communications (ICNC)* (2017), pp. 843–848.

[22] Yann LeCun, Y. B., and Hinton, G. Deep leraning. *Nature 521* (May 2015), 436–444.

[23] Yung-Fa Huang, Chuan-Bi Lin, C.-M. C., and Chen, C.-M. Research on qos classification of network encrypted traffic behavior based on machine learning. *Electronics 10*, 12 (2021).

[24] Zheng Wu, Yu-ning Dong, X. Q., and Jin, J. Online multimedia traffic classification from the qos perspective using deep learning. *Computer Networks 204* (2022), 108716.