Technological University Dublin
ARROW@TU Dublin

Conference papers | School of Electrical and Electronic Engineering

2020-06-04

Intelligent SDN Traffic Classification Using Deep Learning: Deep-SDN

Ali Malik
Technological University Dublin, ali.malik@tudublin.ie

Ruairí de Fréin
Technological University Dublin, ruairi.defrein@tudublin.ie

Mohammed Al-Zeyadi
University of Essex, ma18964@essex.ac.uk

See next page for additional authors

Follow this and additional works at: https://arrow.tudublin.ie/engscheleart

 Part of the Computer and Systems Architecture Commons, and the Digital Communications and Networking Commons

Recommended Citation

Malik, A., De Frein, R., Al-Zeyadi, M., & Andreu-Perez, J. (2020). Intelligent SDN traffic classification using deep learning: deep-SDN. *The 2nd IEEE International Conference on Computer Communication and the Internet (ICCCI)* doi:10.21427/tbet-8c64

This Conference Paper is brought to you for free and open access by the School of Electrical and Electronic Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 4.0 License
Funder: Science Foundation Ireland (SFI)

Authors
Ali Malik, Ruairí de Fréin, Mohammed Al-Zeyadi, and Javier Andreu-Perez

This conference paper is available at ARROW@TU Dublin: https://arrow.tudublin.ie/engscheleart/314

Intelligent SDN Traffic Classification using Deep Learning: Deep-SDN

Ali Malik[1], Ruairí de Fréin[1], Mohammed Al-Zeyadi[2], Javier Andreu-Perez[2]
[1]*School of Electrical and Electronic Engineering*, *Technological University Dublin*, *Ireland*
{ali.malik, ruairi.defrein}@tudublin.ie
[2]*School of Computer Science and Electronic Engineering*, *University of Essex*, *United Kingdom*
{ma18964, javier.andreu}@essex.ac.uk

***Abstract*—Accurate traffic classification is fundamentally important for various network activities such as fine-grained network management and resource utilisation. Port-based approaches, deep packet inspection and machine learning are widely used techniques to classify and analyze network traffic flows. However, over the past several years, the growth of Internet traffic has been explosive due to the greatly increased number of Internet users. Therefore, both port-based and deep packet inspection approaches have become inefficient due to the exponential growth of the Internet applications that incurs high computational cost. The emerging paradigm of software-defined networking has reshaped the network architecture by detaching the control plane from the data plane to result in a centralised network controller that maintains a global view over the whole network on its domain. In this paper, we propose a new deep learning model for software-defined networks that can accurately identify a wide range of traffic applications in a short time, called Deep-SDN. The performance of the proposed model was compared against the state-of-the-art and better results were reported in terms of accuracy, precision, recall, and f-measure. It has been found that 96% as an overall accuracy can be achieved with the proposed model. Based on the obtained results, some further directions are suggested towards achieving further advances in this research area.**

***Index Terms*—SDN, deep learning, big data, traffic analysis, traffic classification, network management.**

I. Introduction

The massive growth of the Internet users over the past years has been accompanied with a rapid evolution of networking systems, e.g. cloud computing and network virtualisation, along with the flourish of communication technologies, e.g. Internet of Things and smart cities [1]. This has led to an exponential growth in data traffic over the heterogeneous networks of the Internet [2].

On one hand, identifying the traffic application types is an essential function of networking systems that facilitates fine-grained management through classifying the network traffic flows. Currently, there are some widely-used approaches to identify and predict the networks traffic, such as: port-based approaches, deep packet inspection and machine learning [3]. Port-based approaches, which are also known as payload-based approaches, are no longer effective due to the dynamic usage of port numbers by applications nowadays [4]. Deep packet inspection is an expensive approach due to its high computational cost [5] in addition to its inability to inspect encrypted traffic. Therefore, in order to cope with these challenges, more intelligence needs to be deployed into networking appliances for learning purposes.

On the other hand, traditional computer networks are composed of a multitude of forwarding elements, i.e. routers and/or switches, that aer governed by a multitude of protocols and run a wide range of applications. Such a heterogeneity in the infrastructure increases the complexity of the network management and performance optimisation. However, traditional networks are inherently distributed systems where each forwarding element maintains a local view over the whole network. Therefore, applying machine learning techniques on a system whose elements have a limited view is another big challenge [6].

The emerging paradigm of Software-Defined Networking (SDN) provides a clean-slate approach to redesigning the network architecture [7], in which the control plane, i.e. controller, is decoupled from the data plane, i.e. forwarding elements. In this new architecture, the logically centralised controller maintains a global view of the entire network. SDN provides a great opportunity for machine learning techniques to be applied, to learn from network traffic data. In this paper, we exploit the advantage of SDN's architecture and propose a new framework that utilises Deep Learning (DL) with the aim of identifying the applications that give rise to network traffic. DL is a promising approach in dealing with significant massive data towards extracting the hidden patterns such as traffic features for classification and prediction purposes. Our goal is to build a SDN network model that is capable of achieving high prediction accuracy in terms of application types within a short time scale.

This paper is organised as follows. Related works are presented in Section II. In Section III, we introduce our proposed method and framework. Performance evaluation and simulation results are presented in Section IV. Finally, the conclusion and future work are presented in Section V.

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/SIRG/3459.

II. Related Work

Recently, DL has gained a surge of interest due to its proven efficacy in solving complex problems in various areas such as networking, robotics, computer vision and speech recognition [8]. Network traffic analysis in SDNs has motivated the application of a number of machine learning applications. We discuss related studies that have utilised machine/DL in traffic classification in this section.

In [9] the authors introduced SBAR, a SDN flow-based monitoring and application recognition framework, as a combination of deep packet inspection and machine learning. SBAR classifies the network traffic load at two levels: (1) application protocol based classification for the monitored traffic and (2) deep packet inspection to identify the application of web and encrypted flows. Compared with traditional machine learning and DNS classifiers, the performance of SBAR was much better in terms of classification accuracy.

The authors in [10] proposed DL as a hybrid deep neural network–based application classification method for SDNs. The proposed model is composed of two components, namely stacked autoencoder and softmax regression layer. The experiment results showed that DL outperformed the Support Vector Machine (SVM) where the overall accuracy was up to 91.2%.

In [11] the authors proposed a QoS-aware traffic classification framework for SDNs. The proposed framework utilised both deep packet inspection and semi-supervised machine learning to classify the network traffic into different classes based on the QoS requirements. The accuracy of the proposed method outperformed the K-means algorithm by accurately classify more than 90% of the traffic classes.

The authors in [12] investigated both SVM and K-means for the purpose of traffic classification in SDNs. The study showed that up to 95% overall accuracy can be achieved.

The authors in [13] presented vTC as a virtual network functions framework that capable of selecting the most suitable machine learning classifiers and the most effective flow features at run time. The experimental results of the study, which have been conducted on KDD [14], showed that the flow classification accuracy can be improved by up to 13%, where the reported overall accuracy was 95.6%.

Utilising machine learning techniques to classify the traffic flows towards developing an application-aware multipath routing method for SDNs was investigated in [15]. The authors presented AMPS framework to automatically classify the incoming traffic flows and apply QoS policy to each flow based on its requirements. A small experimental dataset was created based on 10 clients only where each of them used to run a different application type such as Skype, Facebook, YouTube and Dropbox. The experimental classification results showed that C4.5 DT algorithm obtained the highest accuracy with 98%, however, no precision, recall and f-measure were reported.

The authors in [16] presented a simple architecture that collects the network traffic flows through the standard OpenFlow protocol [17]. For the purpose of classifying the collected data, several machine learning algorithms such as random forest and stochastic gradient boosting were performed. The experimental evaluation, which was conducted over individual application classes, showed that the classification accuracy could reach up to 96%. The study lacks the exploration of the other performance metrics, i.e. precision, recall and f-measure.

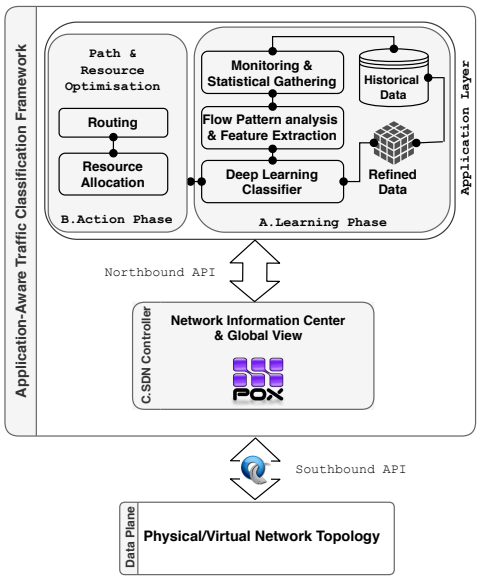

Fig. 1. The architecture of the proposed framework. Openflow is used on the southbound interface and the controller APIs, e.g. POX, are used on the northbound interface.

These works provide important insights into how future intelligent computer networks might function, however, the use of DL for SDNs is still in its infancy. There exist challenges that limit its real-world applicability. With a view to addressing this challenge a new DL model with less computation time and better accuracy is developed as a step towards building smart SDN system that are able to reconfigure the network automatically based on the learning model. In order to conduct a fair comparison, we compare tour work with the DL approach in [10] using the same experimental dataset and evaluation metrics.

III. The Proposed Framework

In this section we discuss our proposed framework and its components. Figure 1 illustrates the main components of the proposed framework where the application layer components represent the main contribution. In the subsections below, we discuss the components of this framework in detail.

A. Learning phase

This component highlights the main contribution of this paper. In this phase, the data collection, feature extraction and classification are made available for the next phase to perform some necessary actions. In the following we describe the modules of this component.

TABLE I
MOORE DATASET APPLICATION CLASSIFICATION.

Traffic class	Representative application/s
WWW	HTTP and HTTPS
Mail	Pop2/3,smtp, and imap
FTP-control	FTP
FTP-pasv	FTP
FTP-data	FTP
Attack	Worm and Virus
P2P	Kazaa, BitTorrent and Gnutella
Database	Postgres, sqlnet, oracle and ingres
Multimedia	Voice and video streaming
Services	X11, dns, ident and ntp

1) Historical Data: This is designed to store almost all the data plane historical information of the underlay topology, e.g. events and incidents, network users and their associated application profiles. The refined data is a filtered dataset that can be used for special purposes. In our case, the refined data is specified for traffic classification where the stored information represents the network traffic flow features such as destination address, destination port, protocol type, packet size and the application class. Such information is selected in order to train the proposed DL model. At this stage of our work, we have utilised Moore dataset [18], which is a real-world traffic, to evaluate the proposed DL model. The dataset contains 10 separate sub-datasets of TCP traffic flows that were collected during different time periods. Each flow sample is comprised of 248 features such as flow size, duration time and the corresponding application class label. Due to the few samples of both interactive and game samples, we have removed them from the refined dataset. The experimental dataset have been categorised into 10 classes as shown in Table I. Finally, a random sample data of 10 samples were selected to get more accurate simulation results.

2) Monitoring and statistical gathering: This module is responsible for collecting information about the network traffic flow periodically which is then stored in the historical data component. The standard OpenFlow protocol is widely used to transfer the collected data from the data plane to the control plane.

3) Flow pattern analysis and feature extraction: This module is responsible for extracting flow features and flow labels in order to construct the refined database. Then, the training dataset is constructed randomly for learning purposes and the testing dataset is constructed for evaluation purposes.

4) DL classifier: We train the DL classifier using randomly selected samples of data, which we call the training set. We then test the classifier on the remaining data, the test dataset. The traffic flow feature set consists of D-dimensional vectors $\mathbf{f}_i \in \mathbb{R}^D$, where $i = 1, \dots N$. A label l_i is associated with each feature vector. There are K distinct classes of traffic. These different classes correspond to different applications. A score function, $f : \mathbb{R}^D \mapsto \mathbb{R}^K$ maps the traffic flow features to the classes using a linear mapping,

$$f(\mathbf{f}_i, \mathbf{W}, \mathbf{b}) = \mathbf{W}\mathbf{f}_i + \mathbf{b}. \tag{1}$$

The parameters that we fit to train our classifier are the weights $\mathbf{W} \in \mathbb{R}^{K \times D}$ and the bias vector $\mathbf{b} \in \mathbb{R}^{K \times 1}$ for a given set of training data features and labels. We optimize $\mathbf{W}$ and $\mathbf{b}$ such that the function $f : \mathbb{R}^D \mapsto \mathbb{R}^K$ produces scores that match with the true labels, l_i, over the whole training set. Once $\mathbf{W}$ and $\mathbf{b}$ have been learned we can then apply $f(\mathbf{f}_i, \mathbf{W}, b)$ to the test data and assign the traffic features in the test data set to labels.

One way to measure how consistent the predictions made using the training data are with the ground truth labels is by using a multi-class support vector machine as the loss function with a squared hinge loss function. A second approach is to generalize binary Logistic Regression to multiple classes, an approach typically classed softmax classification. We favour this second approach as it yields a more intuitive interpretation of the classifier, e.g. a set of normalized class probabilities as its output, and a probabilistic interpretation of the problem. The term $f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$ is called the softmax function. It takes a vector of real-valued inputs and maps it to a vector of values between zero and one that sum to one. The softmax classifier minimizes the cross-entropy between the estimated class probabilities and the true distribution (where all of the probability mass is at the correct class and this function is zero elsewhere). The probabilistic interpretation of the softmax function is that

$$P(l_i|\mathbf{f}_i; \mathbf{W}) = \frac{e^{f_{l_i}}}{\sum_j e^{f_j}} \tag{2}$$

is the normalized probability assigned to the correct label l_i given the feature set $\mathbf{f_i}$ and the weights $\mathbf{W}$, where we have used the bias trick to incorporate $\mathbf{b}$ into $\mathbf{W}$. A more traditional way of viewing this set-up is a Maximum Likelihood Estimation problem. If a regularization side-function is used this becomes Maximum A Posterior estimation, where the choice of a Gaussian is common. Our classifier's prediction is the class corresponding to the largest probability $l_\star = \max_i P(l_i|\mathbf{f}_i; \mathbf{W})$.

The architecture of the proposed learning method, which we call Deep-SDN, includes 12 deep sequential layers in addition to the two standard layers, i.e. input and output. Moreover, the architecture has 3 repetitive blocks where each block includes 4 layers as follow:

(i) A batch normalisation [19] is used to accelerate the process of training as well as to achieve more stable distribution of activation values throughout training.

(ii) A regular densely connected neural-network layer with a number of units, N, is formed as follow:

$$N = t/(\psi + \ell) \times \alpha \tag{3}$$

Where, t represents the number of tensors, ψ represents the application types, ℓ represents the length of tensor and α ranges over the interval [1,10).

(iii) The Rectifier Linear Unit (ReLU) [20], which is used as an activation function for the connected hidden layer. This function is formed as follow:

TABLE II
THE ARCHITECTURE OF DEEP-SDN THAT SHOWS THE DIFFERENT TYPES OF HIDDEN LAYER IN ADDITION TO THE INPUT AND OUTPUT LAYER.

Layer (type)	Output Shape	Param #
Input	(None, 248)	61752
Batch normalisation_1	(None, 248)	992
Activation_1 (Activation)	(None, 248)	0
Dropout_1 (Dropout)	(None, 248)	0
Dense_2 (Dense)	(None, 100)	24900
Batch normalisation_2	(None, 100)	400
Activation_2 (Activation)	(None, 100)	0
Dropout_2 (Dropout)	(None, 100)	0
Dense_3 (Dense)	(None, 50)	5050
Batch normalisation_3	(None, 50)	200
Activation_3 (Activation)	(None, 50)	0
Dropout_3 (Dropout)	(None, 50)	0
Output	(None, 10)	561
Total params: 93,855 Trainable params: 93,059 Non-trainable params: 796		

$$f(x) = max(0, x), \text{ s.t. } f(x) = \{x \in \mathbb{R} | x \geqslant 0\} \quad (4)$$

where x represents as an input value to a neuron. The function $f(x)$ receives thresholding values at 0. The output is 0 when the value of x is less than 0, while, the output is a linear function when $x \geqslant 0$. The ReLU activation function is illustrated in Figure 2.

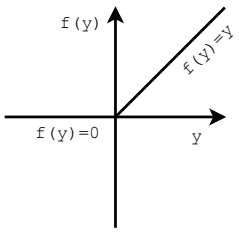

Fig. 2. ReLU function where $f(x)$ is either 0 when $x < 0$, or a linear when $x \geqslant 0$.

(iv) Towards better regularisation and avoiding the over fitting, a dropout layer with 20% dropping rate is adopted [21]. As an illustrative example, Figure 3 (a) shows multi-layer neural network without dropout, while, in Figure 3 (b) a multi-layer neural network with dropout is demonstrated. The model schema architecture of Deep-SDN is given in Table II.

Subsequently, the three architectural blocks are connected to dense Softmax layer as a final step before the classification.

B. Action phase

This phase is responsible for exploiting and adopting the learned information, which is subsequently utilised in high-level processes such as load balancing, routing, resource allocation, with the aim of improving network performance. These high-level processes are out of scope of the current submission, but warrant investigation in future work.

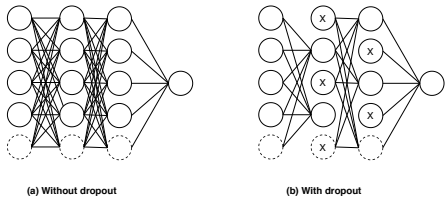

Fig. 3. Deep neural network example that shows the network with and without dropout.

C. SDN controller

The SDN controller represents the network's brain. It is responsible for routing updates, constructing the forwarding rules, optimising the network's performance and classifying the network traffic based on the learned traffic information. There are different types of SDN controllers such as POX, NOX, OpenDayLight with different specifications and characteristics. We will employ the POX controller as it facilitates fast prototyping [22]. The standard OpenFlow protocol is used as a southbound API for establishing the communication between the data and control planes, whereas the set of POX APIs is used on the northbound interface for developing various network control applications.

IV. PERFORMANCE EVALUATION AND SIMULATION RESULTS

We evaluate our approach using Moore's dataset under two headings: (1) the efficiency and (2) the applicability of the approach. The efficiency is concerned with gauging the overall performance of the proposed DL approach while the applicability attempts to gauge to what extent the proposed model can be employed in a real-world SDN environment.

A. Efficiency measurements

To measure the efficiency of the proposed model, we consider the following commonly used performance evaluation metrics: accuracy, precision, recall and f-measure. Then we compare the performance of the proposed model, Deep-SDN, against the DL model proposed in [10]. This part of evaluation is divided into two parts. The first experiment was conducted on the 10 datasets, where each dataset represents a period of time captured within a day, where the purpose of this experiment is to obtain the overall efficiency measure. While, the second experiment was conducted on the combined datasets, i.e. 10, and it has been designed in order to measure the performance over each individual application type. Figure 4 shows the first experimental evaluation results. With 96% overall accuracy, it can be clearly seen that Deep-SDN outperformed DL by achieving higher performance with respect to all measurements metrics, i.e. accuracy, precision, recall and f-measure. Figure 6 shows the second experimental evaluation results. Again, Deep-SDN showed its ability to obtain higher performance compared to DL. Such efficiency improvement comes as an advantage of using both batch normalisation and dropout layers in the architecture of Deep-SDN. These

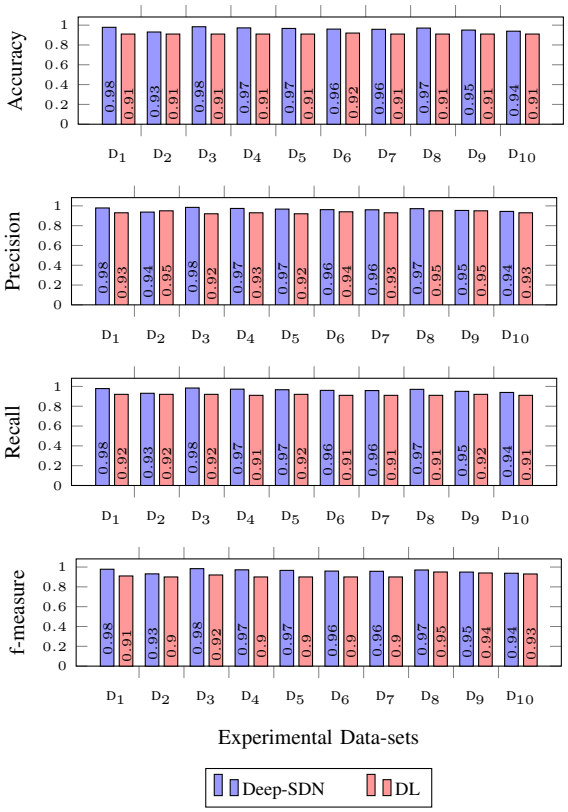

Fig. 4. Performance measurements over 10 different Moore datasets. The performance of Deep-SDN over the four metrics is better than the performance of DL.

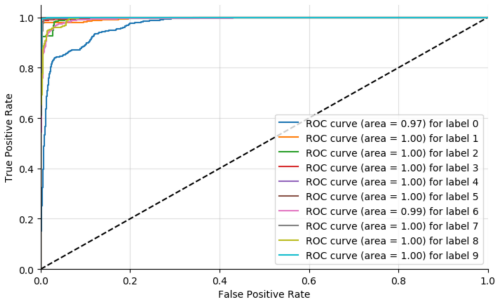

Fig. 5. Receiver operating characteristic (ROC) analysis for the application types/classes of the 10 experimental datasets.

two layers have brought more robustness to the proposed model, which results in enhancing the performance efficiency. Moreover, we include the receiver operating characteristic (ROC) curve in Figure 5, which shows the true and false positive rates of each application class that obtained by the Deep-SDN.

B. Applicability measurements

Given that the controller is responsible for dealing with the newly arrived traffic flow to a SDN network, therefore, as a control application, it is necessary for the classification module to be able to process the arrived traffic in real time fashion. Thus, measuring the required time to classify a number of

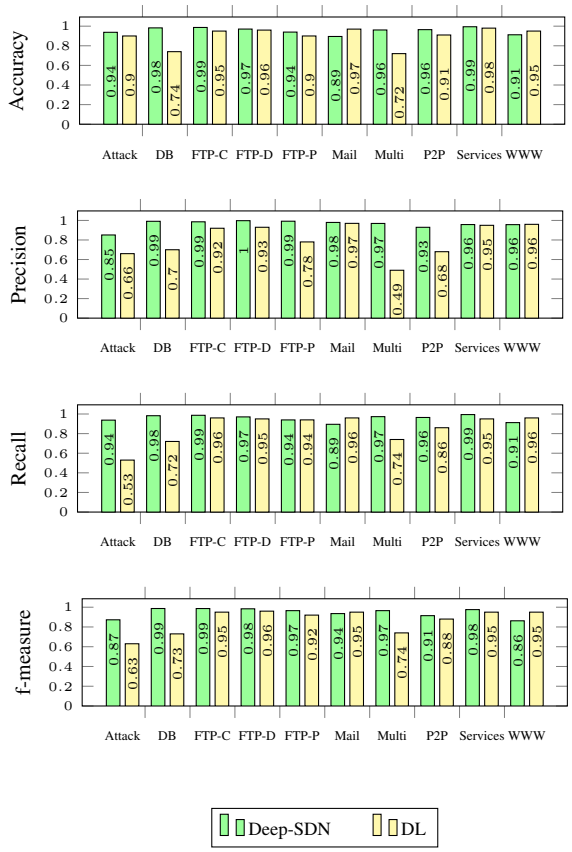

Fig. 6. Performance measurements over the individual applications of Moore datasets. Deep-SDN performs better than DL across the different type of applications.

TABLE III
TIME REQUIRED TO CLASSIFY DIFFERENT TRAFFIC FLOW REQUESTS AT ONCE.

Requests	Time (ms)
100	1.938
1000	2.38
10000	6.371
100000	11.341

flows, i.e. requests, is necessary to provide an indication about the speed of the classification process. To do so, random traffic flows with different sizes, i.e. 100–100000, were selected from the experimental dataset. Then, we report the classification latency of each traffic as shown in Table III. According to Table III, one can observe is that the processing time is directly proportional to the number of requests, also, a large number of requests can be classified within a reasonable time where it took only 11.3ms to classify 100000 requests.

V. CONCLUSION

This paper demonstrated the promise of using DL in the problem of network traffic classification. A new application-aware classification framework for SDNs is introduced. The proposed framework consists of two main parts, namely learning and action. In this ongoing study, we have implemented

the first part, i.e. learning, through implementing a new DL model, called Deep-SDN. The proposed model is able to identify the network traffic application types with high accuracy and high speed, which makes it applicable for online traffic identification. The performance of Deep-SDN was tested and evaluated through extensive simulation experiments on a real-world traffic dataset. Four metrics were used to examine our model, these were: accuracy, precision, recall and f-measure. The experimental findings have shown the effectiveness of Deep-SDN in identifying the traffic application types. The overall performance of Deep-SDN is compared against the proposed model in [10]. It has been found that Deep-SDN exhibits better performance by reporting 96% overall accuracy.

As part of our future work, we intend to implement the remaining part of our framework, i.e. action, in order to use the useful information that obtained from the learning phase in different network aspects such as resource allocation and routing. Also, we are planning to extend our proposed Deep-SDN to be able to predict the network traffic by considering some approaches like [23], [24].

REFERENCES

[1] J. Naughton, "The evolution of the internet: from military experiment to general purpose technology," *Journal of Cyber Policy*, vol. 1, no. 1, pp. 5–28, 2016.

[2] "Cisco annual internet report (2018–2023) white paper," 2020, . Accessed on 14^{th} April 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[3] N. Al Khater and R. E. Overill, "Network traffic classification techniques and challenges," in *2015 Tenth International Conference on Digital Information Management (ICDIM)*. IEEE, 2015, pp. 43–48.

[4] Y. Xue, D. Wang, and L. Zhang, "Traffic classification: Issues and challenges," in *2013 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2013, pp. 545–549.

[5] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393–430, 2018.

[6] A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, E. Alarcón, M. Solé, V. Muntés-Mulero, D. Meyer, S. Barkai, M. J. Hibbett *et al.*, "Knowledge-defined networking," *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, pp. 2–10, 2017.

[7] T. Koponen, S. Shenker, H. Balakrishnan, N. Feamster, I. Ganichev, A. Ghodsi, P. B. Godfrey, N. McKeown, G. Parulkar, B. Raghavan *et al.*, "Architecting for innovation," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 3, pp. 24–36, 2011.

[8] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[9] J. Suárez-Varela and P. Barlet-Ros, "Sbar: Sdn flow-based monitoring and application recognition," in *Proceedings of the Symposium on SDN Research*, 2018, pp. 1–2.

[10] C. Zhang, X. Wang, F. Li, Q. He, and M. Huang, "Deep learning–based network application classification for sdn," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 5, p. e3302, 2018.

[11] P. Wang, S.-C. Lin, and M. Luo, "A framework for qos-aware traffic classification using semi-supervised machine learning in sdns," in *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 2016, pp. 760–765.

[12] Z. Fan and R. Liu, "Investigation of machine learning based network traffic classification," in *2017 International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2017, pp. 1–6.

[13] L. He, C. Xu, and Y. Luo, "Vtc: Machine learning based traffic classification as a virtual network function," in *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, 2016, pp. 53–56.

[14] S. D. Bay, D. Kibler, M. J. Pazzani, and P. Smyth, "The uci kdd archive of large data sets for data mining research and experimentation," *ACM SIGKDD explorations newsletter*, vol. 2, no. 2, pp. 81–85, 2000.

[15] S. T. V. Pasca, S. S. P. Kodali, and K. Kataoka, "Amps: Application aware multipath flow routing using machine learning in sdn," in *2017 Twenty-third National Conference on Communications (NCC)*. IEEE, 2017, pp. 1–6.

[16] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, "Machine learning in software defined networks: Data collection and traffic classification," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. IEEE, 2016, pp. 1–5.

[17] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[18] A. Moore, D. Zuev, and M. Crogan, "Discriminators for use in flow-based classification," Intel Research, Cambridge., Tech. Rep., 2005.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[20] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.

[21] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Advances in neural information processing systems*, 2017, pp. 3581–3590.

[22] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, and R. Smeliansky, "Advanced study of sdn/openflow controllers," in *Proceedings of the 9th central & eastern european software engineering conference in russia*. ACM, 2013, p. 1.

[23] B. Pfülb, C. Hardegen, A. Gepperth, and S. Rieger, "A study of deep learning for network traffic data forecasting," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 497–512.

[24] C. Hardegen, B. Pfülb, S. Rieger, A. Gepperth, and S. Reißmann, "Flow-based throughput prediction using deep learning and real-world network traffic," in *2019 15th International Conference on Network and Service Management (CNSM)*. IEEE, 2019, pp. 1–9.