Received: 29 March 2022 | Revised: 23 October 2022 | Accepted: 22 November 2022

IET Communications

DOI: 10.1049/cmu2.12548

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

A neuro-evolutionary approach for software defined wireless network traffic classification

Buddhadeb Pradhan[1] | **Mir Wajahat Hussain**[2] | **Gautam Srivastava**[3,4,8] | **Mrinal K. Debbarma**[5] | **Rabindra K. Barik**[6] | **Jerry Chun-Wei Lin**[7]

[1]Department of Computer Science and Engineering, University of Engineering and Management, Kolkata, West Bengal, India

[2]Department of Computer Science and Engineering, Alliance College of Engineering and Design, Alliance University, Anekal, Karnataka, India

[3]Department of Mathematics and Computer Science, Brandon University, Brandon, Manitoba, Canada

[4]Research Center for Interneural Computing, China Medical University, Taichung, Taiwan

[5]Department of Computer Science and Engineering, National Institute of Technology Agartala, Jirania, Agartala, Tripura, India

[6]School of Computer Applications, KIIT Deemed to be University, Patia, Bhubaneswar, Odisha, India

[7]Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway

[8]Department of Computer Science and Math, Lebanese American University, Beirut, Lebanon

Correspondence
Jerry Chun-Wei Lin, Western Norway University of Applied Sciences, Bergen, Norway.
Email: jerrylin@ieee.org

Abstract

Accurate network traffic classification is an essential and challenging issue for wireless network management and survivability. Existing network traffic classification algorithms, on the other hand, cannot meet the required specifications of real networks' in terms of user privacy control overhead, latency, and above all, classification speed. For wireless network traffic classification, machine learning-based and hybrid optimization techniques have been deployed. This paper takes a software-defined wireless network (SDWN) architecture for network traffic classification into account. Because the proposed scheme is perfectly contained within the network controller,the SDWN controller's higher processing capability, global visibility, and programmability can be used to achieve real-time, adaptive, and precise traffic classification. In this paper, a neuro-evolutionary approach is proposed in which the feed forward neural network (FFNN) is the base classifier and particle swarm optimization (PSO) is used to train the FFNN to accurately classify traffic while minimizing communication overhead between the controller and the SDWN switches. Simulation experiments were conducted by acquiring real-world internet datasets to test the efficacy of the proposed scheme. The results and the state-of-the-art comparisons show that the proposed approach has outperformed in terms of accuracy in wireless traffic classification.

1 | INTRODUCTION

Substantial growth in a myriad of applications has led to a vast rise in data with strict network needs. Conventional network devices operate with proprietary protocols, basically, a closed set of interfaces. They have inter-wined control with the forwarding/data plane, which makes quality of services (QoS) and deployment of any newer policy as per the application requirements very difficult [1].

Software-defined wireless networks (SDWN) have been regarded as the de facto standard for network paradigms owing to the removal of the control plane from the forwarding plane. This separation of planes facilitates innovation, removal of vendor lock-in, and flexibility in the network through specific policy enforcement as dictated by the application [2, 3]. The 'to' and 'from' communication is enabled by a well-known OpenFlow (OF) protocol [4, 5]. OF protocol amasses together the network statistics from the forwarding plane to a central point known as the control plane [6]. The control plane then dictates the network policy for each flow based on the global information/status received from all the devices operating in the network, which is depicted in Figure 1.

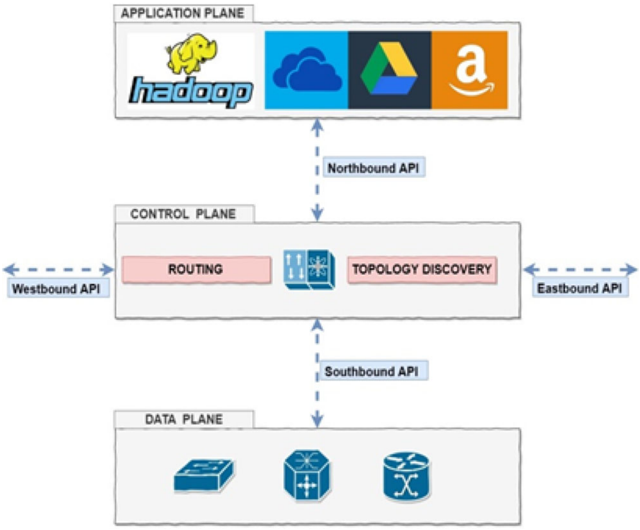

FIGURE 1 Software defined network planes overview

SDWN facilitates smooth big-data network handling through data streams emanating from increased IoT sensors to the number of Cloud data centers or inter-data center network traffic. The distributed information gathering from network devices enables the SDWN controller to make informed decisions. With emerging information and communication technologies (ICT) and the proliferation of next-generation networks, real-time IoT applications have a vast variance in the magnitude of data and frequency. Traffic classification might aid the controller in better network optimization, traffic engineering, and application-aware routing decisions. Works focused on traffic classification have emphasized handling application-centric nature while keeping the critical driver, namely QoS, unattempted for real-time IoT-based applications. Machine learning techniques predict effective decision-making with the help of historical and real-time data [4]. The statistical analysis of networks is easily done by every OF switch with the help of counters. Hence, these can be utilized by the controller to implement precise decision-making for networks. The demands of a myriad of applications and their conflicting resource requirements can be met by enabling application-aware networking. The key requirement to application-aware networking is network traffic classification. Network traffic classification in SDWN by the controller is pertinent to making informed decisions about network requirements of the application and overall network survivability. This kind of traffic classification would enable the way we segregate both small (mice) and large flows that affect data-center performance and thus QoS requirement is fulfilled considerably [7]. This segregation of both small and large flows is pertinent due to consuming considerable bandwidth, thus severely affecting the performance of the small flow because of having less delay. Furthermore, to satisfy the resource allocation requirements (QoS) for every application, traffic classification is pertinent for seamless operation in the network [4]. With a centralized view in SDWN, traffic classification by the controller will dictate application-specific rules to the forwarding plane which is essential in terms of both network efficiency as well as seamless network operation. Although traffic classification has been studied profoundly, it is still a pertinent research problem. FFNN-PSO is leveraged for fine-tuning overall performance. As per the literature surveyed, such SDWN traffic classification and the performance exhibited with ML and hybrid approaches are not found in the literature [8, 9]. Hence, the main contributions of this paper are as follows:

- A neuro-evolutionary approach has been implemented to improve the accuracy of traffic classification
- Machine learning algorithms are employed to test the suitability of the novel SDWN traffic classification problem.
- Stability analysis has been performed to validate the hybridization of FFNN and PSO

The remainder of the manuscript is organized as follows: Section 2 presents the state-of-the-art for traffic classification in networks. Section 3.1 provides a summary of the WSN and its application vis-a-vis SDWN. Section 4 discusses the overall solution methodology as well as problem formulation. Section 5 illustrates the details surrounding the implementation and conveys the results drawn. Finally, Section 6 concludes this manuscript.

2 | RELATED WORKS

Numerous attempts for network traffic classification have been taken care of, including QoS awareness, flow awareness, and application-aware. It was envisioned that application-aware classification should be focused on rather than QoS and flow-aware, whereas QoS helps distinguish the classes of a multitude of flows. The QoS-based traffic classification work is made up of semi-supervised learning, and deep packet inspection (DPI) for classification of traffic as was done by Wang et al. [10].

Flow-aware classification divides network traffic into elephant (huge) and mice (small) flows and Glick et al. [11] focussed on scheduling flow in a data center which is a hybrid architecture. ML algorithms find suitable applications at the network edge for classifying elephant flows. Peng et al. [12] classified elephant flows by employing a two-way strategy. Firstly, the identification of elephant flows was based on fields associated with the packet header. Secondly, a popular ML technique namely a decision tree was leveraged to classify elephant-type flows. Amaral et al. [13] utilized an OF-based SDN system for an enterprise network for traffic classification. When data collection was completed, then with the help of multiple classifiers, traffic flows are categorized into several applications. Li et al. [12] utilized a multi-classifier for traffic classification and employed both DPI and ML-based classifiers. For the start of every new-flow, an ML-based classifier was employed first and a check is given on reliability if it is above some predetermined threshold. Rossi et al. [14] utilized UDP traffic to enable traffic classification. To enable application-aware traffic classification, a scheme based on behavioural classification was proposed. The SVM algorithm is leveraged for classification of UDP traffic depending on the reception of packets and this traffic classification scheme shows a 90% accuracy. Qazi et al. [15] proposed Atlas, a traffic classification framework which enabled traffic classification for mobile applications. A crowd-sourcing approach is used to deal with ground-truth data and a decision tree was employed for traffic classification. The Atlas framework had a good accuracy of 94%. Shao et al. [16] then developed a neural-evolving model in which the model achieves good performance compared to past comparable works. Nakao et al. [17] utilized deep neural networks (NN) for identifying mobile-based applications. Experimental data was collected for traffic classification. For an eight layer NN model, five features were selected (type of the protocol, destination (address, port), size of the packet, and time to live (TTL)). Results for the mobile traffic classification scheme had an efficacy of around 93.5% and had categorized a total of 200 applications.

Existing works on network classification are limited by their application-centric nature, thus overlooking key criterion for real-time IoT applications, namely quality of service (QoS). In this paper, the focus is on augmenting SDN controllers' decision-making capacity and SDWN (software defined wireless networks) with machine learning (ML) algorithms to achieve real-time [18–20], QoS-aware network traffic classification. In short, the proposed framework jointly exploits optimization algorithms and semi-supervised ML for precise traffic classification while keeping communication overhead between controller and SDN switches minimal.

3 | PRELIMINARIES

Here in this section, a brief description of SDWN and the basics of Feed Forward Neural Networks (FFNN) and Particle Swarm Optimization (PSO) are discussed.

3.1 | Software defined wireless networks

Software defined wireless networks (SDWN) [21] are ideally suited for low-rate personal area wireless networks with the utilization of minimal resources and shorter ranges. Sensor nodes in WSN are equipped with sensing, processing units and radio unit but with the proliferation of IoT sensors, WSN faces multiple challenges due to limited resources [22]. SDN [23, 24] can be perfectly used as a test case in WSN because of the centralization of the controller which is depicted in Figure 2. With the centralization at the controller, sensors can just become data/forwarding plane elements and can be relieved from all network control/management tasks like topology discovery, routing, etc.

3.2 | Feed forward neural networks

Feed forward neural networks (FFNN) are a pertinent data-driven computational technique that finds applications in various network traffic problems having a combination of topology information and network traffic. Basic details related to FFNN have been extracted from ref. [25].

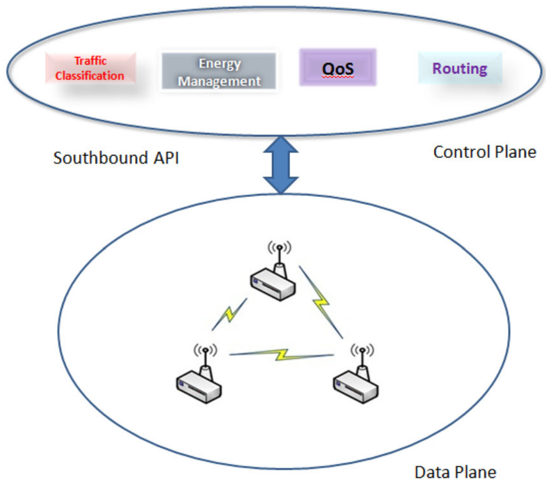

FIGURE 2 A SDWN architecture

3.3 | Basics of PSO

Particle Swarm Optimization (PSO) came into existence in the year 1995 investigated first by Kennedy and Eberhart [26]. It is regarded as a stochastic population-based meta-heuristic algorithm. Let us assume that the search space and the size of the swarm are D-dimensional and N (population size). The detailed mathematical notation for the same is extracted from ref. [27].

The ith particle position is represented as $x_{id} = (x_{i1}, x_{i2}, ..., x_{iD})$ where $x_{id} \in [lb_d, ub_d]$, $d \in [1, D]$ and ub_d and lb_d are the upper and lower bounds of the dth dimension of the search space. The velocity of the ith particle is represented as $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$. After each time step elapses, the velocity and position of the particle get changed as per the equations discussed below:

$$v_{ij}(t+1) = \omega \times v_{ij}(t) + c_1 \times r_1 \times \left(p^{lB}{}_{ij}(t) - x_{ij}(t) \right) + c_2 \times r_2 \times \left(p^{gB}{}_{j}(t) - x_{ij}(t) \right) \tag{1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{2}$$

where

- ω: inertia weight which balances the exploration and exploitation ability of PSO.
- r_1, r_2: two distinct random numbers, $r_1, r_2 \sim U(0, 1)$.
- c_1, c_2: acceleration-coefficients which pulls particle information in terms of best and global-best positions.
- t: current iteration.
- p_i^{lB}: best previous position found so far by the particle, called local best.
- p^{gB}: best position discovered so far by the whole swarm, called global best.
- $\omega \times v_{ij}(t)$: provides exploration ability for PSO.
- $c_1 \times r_1 \times (p^{lB}{}_{ij}(t) - x_{ij}(t))$: represents private thinking.

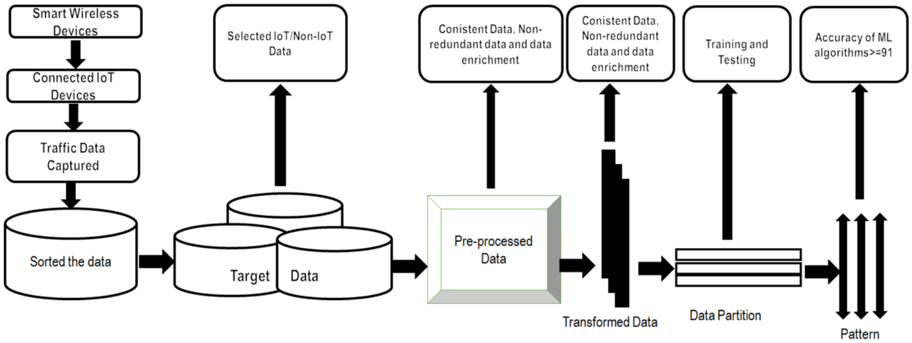

FIGURE 3 Flowchart of working procedure of the proposed approach

- $c_2 \times r_2 \times (p^{gB}{}_{j}(t) - x_{ij}(t))$: represents collaboration of particles.

4 | MATERIALS AND METHODS

In this section, proposed approach details, benchmark dataset description, experimental setup, and finally performance metrics are provided.

4.1 | Proposed approach

The software-defined network traffic classification problem and methodology adopted for this study are described in this section. Traffic classification methodologies have been briefly described, ranging from data collection to optimized network traffic. Figure 3 describes the working procedure for this traffic classification problem leveraging the neuro-evolutionary approach. In the consequent subsections, detailed discussions are addressed.

4.1.1 | Queries are to be addressed

While collecting the internet traffic dataset, the below-mentioned queries are to be considered:

- Will there be any problem if testing and training datasets are collected from different networks?
- Will it is beneficial if testing and training datasets are obtained from identical network requirements?
- What if network attributes are changed and will they affect network traffic?

4.2 | Implementation of algorithm

To address the above queries, dataset collection and ML algorithm implementations are done tactfully. The primary issue is to fit the proposed problem with ML algorithms. Numerous ML algorithms exist, but it is hard to decide which algorithm will be the best fit for specific SDWN applications. To test the efficacy of the proposed traffic classification problem, two popular ML algorithms, namely Logistic Regression (LR) and Naïve Bayes (NB) have been identified to fit the suitability of the problem. The detailed explanation of these two algorithms is extracted from ref. [27]. In this paper, the traffic classification problem is investigated with other classifiers, and it is envisioned that a neuro-evolutionary approach is found to be best suited for the problem. An activity diagram as shown in Figure 3 represents the working procedure of ML-based traffic classification for SDWN. The importance of using a neuro-evolutionary approach for this problem is to test the suitability of two different dimension-based problems. In this regard, a stability analysis for this proposed approach is employed.

4.3 | Complexity analysis of PSO

The number of computations required for a complete run of the PSO algorithm is the sum of computations required to calculate the cost of a candidate solution (based on the current position of the particles) and computations required to update each particle's position (1) and velocity (2). Both of these are directly proportional to the number of iterations.

The computational complexity of evaluating the cost function depends on the particular cost function under consideration. These computations need to be performed at every iteration for all PSO variants and cannot be reduced.

For the second set of computations (i.e. the ones required for the update equations), the standard PSO algorithm requires 5DN multiplications per iteration. This shows that the cost associated with the update equations makes a significant contribution to the total computational cost of the PSO algorithm.

4.4 | Stability analysis of FFNN-PSO

PSO stability analysis is carried out for the problem mentioned above. Here, FFNN gets trained with PSO & the trained FFNN with PSO is leveraged for network traffic classification. Further, the PSO stability analysis concludes that the proposed neuro-evolutionary approach is also stable to implement for the traffic classification problem.

Here, the stability-analysis derivation is presented. In this derivation, $pbest(t_l)$ and $gbest(t_g)$ are kept fixed for an iteration [28, 29]. The detailed parameter information can be found from ref. [27]. The proposed algorithm discussion can be minimized further without any loss of generality to one-dimensional based analysis which is as follows:

$$g_{s+1} = \omega * g_s + \gamma_1 * (s_l - n_s) + \gamma_2 * (t_g - v_s) \tag{3}$$

Let

$$\gamma = \gamma_1 + \gamma_2, p = \frac{\gamma_1 s_l + \gamma_2 s_g}{\gamma_1 + \gamma_2} \tag{4}$$

Then, Equation (3) can be facilitated as:

$$g_{s+1} = g_s * \omega + (t - v_s) * \gamma \tag{5}$$

$$d(q_{s+1}) = d(q_s) * (1 - d(\gamma)) + d(\gamma) * t \tag{6}$$

$$d(q_{s+1}) = d(q_s) * (1 - \tau_\gamma) + \tau_\gamma * t \tag{7}$$

Here, τ_γ is the normal value of γ and it is random [0,1] one. So, $d(\gamma) = \frac{1}{2}$.

The generic formulation for recurrence relation is as follows:

$$d(q_s) = \frac{1}{2^s} * (q_0 - \gamma) + t \tag{8}$$

where the initial position is denoted as q_0.

Lemma 1. *The sequence $d(q_s)$ is convergent and converges to t.*

Proof: $d(q_s)$ converging to t suggests that the value of $s'(\nu)$ exists for $\exists\, \nu > 0$, such that if $s > s'(\nu)$

$$|d(q_s) - t| < \nu \tag{9}$$

Considering Equation (8)

$$\frac{1}{2^s}|(q_0 - Z)| < \nu \tag{10}$$

Therefore it can be concluded that

$$2^s > \frac{|(q_0 - Z)|}{\nu} \tag{11}$$

So,

$$s > \log(\frac{|(q_0 - Z)|}{\nu}) \tag{12}$$

4.5 | Stability analysis for second-order

The mathematical formulation for the random variable's v variance is expressed through second-order stability analysis which is mentioned below:

$$N(q) = d^2(q) - d(q^2) \tag{13}$$

In this way, it is necessary to compute $d({q_s}^2)$ for calculating $N(q)$. q^2_{s+1} will be computed as per the following;

$$q^2_{s+1} = [q_s * (1 - \gamma) + p * \gamma]^2 \tag{14}$$

$$q^2_{s+1} = (\gamma^2 + 1 - \gamma * 2) * q^2_s + 2 * p * q_t * \gamma(1 - \gamma) + t^2 * \gamma^2 \tag{15}$$

So the desired value of q^2_{s+1} is

$$\begin{aligned} d(q^2_{s+1}) = d[&(1 - 2 * \gamma + \gamma^2) * q^2_s \\ &+ 2 * \gamma(1 - \gamma) * t * q_s + \gamma^2 * t^2] \end{aligned} \tag{16}$$

Deriving from Equation (16)

$$\begin{aligned} d(y^2_{s+1}) = &(1 - 2 * \tau_\gamma + d(\gamma^2)) * d({q_s}^2) \\ &+ 2 * t(\tau_\gamma - d(\beta^2)) * d(q_s) \\ &+ t^2 * d(\gamma^2) \end{aligned} \tag{17}$$

As is already mentioned, γ, a random number, distributed uniformly, varies between 0 to 1, So,

$$\begin{cases} d(\gamma^2) = \dfrac{1}{3} \\ N(\gamma^2) = \dfrac{1}{12} \end{cases} \tag{18}$$

Substituting Equation (18) in Equation (17)

$$d(q^2_{s+1}) = \frac{1}{3} * d(q^2_s) + \frac{1}{3} * t * d(q_s) + \frac{1}{3} * t^2 \tag{19}$$

The value of $d^2(q_{s+1})$ is calculated as follows:

$$d(q^2_{s+1}) = \frac{1}{4} * d^2(q_s) + \frac{1}{2} * Z * d(q_s) + \frac{1}{4} * t^2 \tag{20}$$

Now $N(q_{s+1})$ can be obtained by substituting Equations (19) and (20) in Equation (10) as:

$$N(q_{s+1}) = \frac{1}{4} * N(q_s) + \frac{1}{12} * d(q_s - t^2) \tag{21}$$

Hence,

$$d(q_s - t^2) = \frac{1}{3} * d(q_s - t^2) \tag{22}$$

Recurrence relation can be mathematically expressed as;

$$N(q_s) = \frac{1}{4^s} * n(q_0) + d(q_0 - s)^2 * \left(\frac{1}{3^s} - \frac{1}{4^s}\right) \tag{23}$$

Lemma 2. *The sequence $n(q_t)$ is convergent and converges at 0.*

Proof: The ultimate limit of $v(y_t)$ approaches following $v(y_0) = 0$ when $t \to \infty$

$$\lim_{t\to\infty} v(y_t) = \lim_{t\to\infty} (E(y_0 - p)^2) * \left(\frac{1}{3^t} - \frac{1}{4^t}\right)$$
$$= E(y_0 - p)^2 \lim_{t\to\infty} \left(\frac{1}{3^t} - \frac{1}{4^t}\right)$$
$$= 0 \tag{24}$$

Hence, it is evident that our proposed neuro-evolutionary approach meets the criteria to qualify for the first and second-order stability analysis.

4.6 | Dataset description

We have used one dataset for our publicly available approach. The brief description of those considered datasets is as follows: The used dataset has 21K rows and covers ten local workstation IPs for a tenure of 90 days. Each row consists of four columns:

- date: dd-mm-yyyy (from 01-07-2006 through 30-09-2006)
- l_ipn: local IP (signifies as an integer number between 0-9)
- r_asn: remote ASN (identifies the remote ISP as integer number)
- f: flows (connections counting for a particular day)

In this dataset, 20,804 samples are there and four attributes are present. The dataset source can be found in Kaggle[1].

4.7 | Performance metrics

The accuracy parameter is used to evaluate the classification model. It signifies relationships and patterns between variables in a dataset based on the input, or training, data. In the case of binary classification, accuracy can also be calculated in terms of positives and negatives:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{25}$$

where TP = true positives, TN = true negatives, FP= false positives, and FN = false negatives.

4.8 | System configuration and experimental environment

In this paper, system setup, and software development setup is as follows:

- System configuration: Windows 10 64 bit OS, processor AMD A6-9225 Radeon R4 with RAM 4 Gb.

[1] https://www.kaggle.com/datasets/crawford/computer-network-traffic

TABLE 1 Network traffic distribution related to different applications

Types of applications	Traffic numbers	Traffic class
Skype, FTPS, SFTP	554	File transfer
Firefox, Chrome	795	Web browsing
Whatsapp, Skype, Facebook, Hangouts	1045	Chat
YouTube	673	Streaming

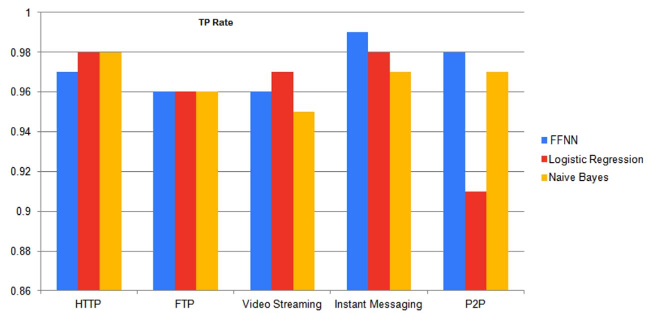

FIGURE 4 TP percentage of different ML algorithms based on QoS parameters

- Software development: The neuro-evolutionary approach for traffic classification problems is implemented through Python 3.7 with the help of Jupyter Notebook.

5 | RESULTS, ANALYSIS, AND DISCUSSION

A real network traffic dataset is used to test the ensemble-based network traffic classification. The Kaggle dataset was used as the training and testing datasets. The following is the list of network flows studied for network traffic classification which is as follows: VoIP, P2P, streaming, file transfer, web browsing, streaming, and chat. Table 1 summarizes the traffic types and network applications that relate to them. We divided network traffic into several classes based on QoS needs rather than identifying specific applications.

The patterns used by the ML classifier for traffic identification include the inter-arrival time and length of the packet. Skype has gained huge prominence because it represents P2P based application and it belongs to VoIP. Figure 4 shows the traffic pattern (TP) percentage among the multiple protocols leveraged for this study which includes FTP, Video Streaming, HTTP, P2P and instant messaging.

Table 1 describes the traffic data distribution in SDWN for the protocols used in this study. From Figure 5, it is shown that the proposed neuro-evolutionary has suggested less traffic accuracy than its counterparts due to carrying most traffic. On the other hand, the accuracy of instant-messaging was the highest compared to others due to carrying less traffic load. The suitable proportion traffic percentage has been taken for each traffic distribution. The accuracy comparison percentage for the three different approaches has been presented. Figure 5

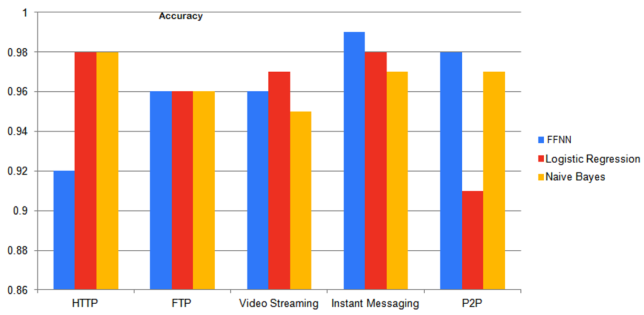

FIGURE 5 Accuracy comparison among the different ML algorithms

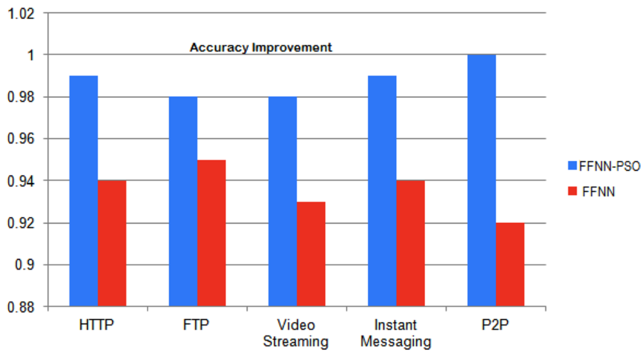

FIGURE 6 Accuracy improvement using optimization algorithm

describes the net accuracy comparison for the above-mentioned approaches. As already said, using neuro-evolutionary approach results in a change in traffic classification accuracy. This is illustrated in Figure 6, which describes that inclusion of a hybrid approach, more accuracy improvement is noted with a solitary optimization algorithm. It can be effectively stated that the network traffic classification is substantially improved with the neuro-evolutionary process.

First, the optimization process is carried out by loading a feed forward neural network with the initialized parameters. The settings for the former include the comprehensive attribute set and their reduction into a small enough to be collected and manipulated in real-time. The one-time collection step will not ensure that the data pre-processing classification process is not overwhelmed. The count of the iterations is kept to be constant afterward to ensure the optimization of the nodes and activation function. Lastly, a stability analysis is executed on the training iterations count. The step-by-step optimization method cannot always converge to either a global maximum or a local minimum. From our experimental analysis, a higher classification accuracy of about (96%) is achieved which can be verified in Table 2.

5.1 | Comparison with the state-of-the-art

The proposed wireless traffic classification method was compared with four deep learning (DL) algorithms. These DL algorithms are: Long Short Term Memory (LSTM) [31], Deep Belief Networks (DBN) [30], Convolutional Neural Networks (CNN) [33], and Recurrent Neural Networks (RNN) [32]. Experimental results as described in Table 2 show that the proposed FFNN-PSO based scheme has improvements in accuracy classification when directly compared to other approaches.

TABLE 2 Proposed wireless traffic network classification comparison

Methodology & ref. no utilized	Precision	Recall	Accuracy
Deep belief network (DBN), [30]	95	96	96
Long short-term memory (LSTM), [31]	92	93	92
Recurrent neural network (RNN), [32]	94	93	94
Convolutional neural network (CNN), [33]	94	95	95
Proposed traffic classification (FFNN-PSO)	96	96	96

6 | CONCLUSION

This paper uses ML methods to classify traffic in SDN networks to make informed decisions regarding their quality of service (QoS) requirements and their underlying applications. Datasets were normalized for classification purposes, and efficacy was increased through testing and training of datasets acquired from open sources. Three ML classifiers, including Logistic Regression (LR), Naïve Bayes (NB), and Feed Forward Neural Networks (FFNN), were considered with a hybrid FFNN-PSO. ML algorithms were used to improve the accuracy of traffic classification. Furthermore, using FFNN-PSO facilitates accuracy improvements of traffic classification using the same classifiers. Because there is no processing overhead with the proposed strategy, it appears to be promising. The future works will deal with detecting flows of a new application, including implementations on Windows, iOS, and Linux. Underwater wireless sensor network research is also undergoing to address the traffic classification problem.

AUTHOR CONTRIBUTIONS

Buddhadeb Pradhan: Conceptualization, methodology, writing - original draft. Mir Wajahat Hussain: Formal analysis, investigation, writing - original draft. Gautam Srivastava: Validation, writing - review and editing. Mrinal K. Debbarma: Methodology, validation. Rabindra K. Barik: Investigation, writing - review and editing. Jerry Chun-Wei Lin: Conceptualization, supervision, writing - review and editing.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

FUNDING INFORMATION

This research was partially funded by Western Norway University of Applied Sciences, Bergen, Norway.

DATA AVAILABILITY STATEMENT

Data is available upon request from the corresponding author.

ORCID

Gautam Srivastava https://orcid.org/0000-0001-9851-4103

REFERENCES

1. Sinha, Y., Haribabu, K., et al.: A survey: hybrid SDN. J. Network Comput. Appl. 100, 35–55 (2017)
2. Hussain, M., Reddy, K., Roy, D.: Resource aware execution of speculated tasks in Hadoop with SDN. Int. J. Adv. Sci. Technol. 28(13), 72–84 (2019)
3. Hussain, M.W., Reddy, K.H.K., Rodrigues, J.J., Roy, D.S.: An indirect controller-legacy switch forwarding scheme for link discovery in hybrid SDN. IEEE Syst. J. 15(2), 3142–3149 (2020)
4. Kreutz, D., Ramos, F.M., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. Proc. IEEE 103(1), 14–76 (2014)
5. Hussain, M.W., Pradhan, B., Gao, X.Z., Reddy, K.H.K., Roy, D.S.: Clonal selection algorithm for energy minimization in software defined networks. Appl. Soft Comput. 96, 106617 (2020)
6. Hussain, M.W., Moulik, S., Roy, D.S.: A broadcast based link discovery scheme for minimizing messages in software defined networks. In: 2021 IEEE Globecom Workshops (GC Wkshps), pp. 1–6. IEEE, Piscataway, NJ (2021)
7. Xie, J., Yu, F.R., Huang, T., Xie, R., Liu, J., Wang, C., et al.: A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. IEEE Commun. Surv. Tut. 21(1), 393–430 (2018)
8. Sharma, D.K., Brahmachari, S., Singhal, K., Gupta, D.: Data driven predictive maintenance applications for industrial systems with temporal convolutional networks. Comput. Ind. Eng. 169, 108213 (2022)
9. Alqaralleh, B.A., Aldhaban, F., Alqaralleh, E.A., Kumar, A., Gupta, D., Joshi, G.P., et al.: Swarm intelligence with adaptive neuro-fuzzy inference system-based routing protocol for clustered wireless sensor networks. Comput. Intell. Neurosci. 2022, 7940895 (2022)
10. Wang, P., Lin, S.C., Luo, M.: A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs. In: 2016 IEEE International conference on services computing (SCC), pp. 760–765. IEEE, Piscataway, NJ (2016)
11. Glick, M., Rastegarfar, H.: Scheduling and control in hybrid data centers. In: 2017 IEEE Photonics Society Summer Topical Meeting Series (SUM), pp. 115–116. IEEE, Piscataway, NJ (2017)
12. Xiao, P., Qu, W., Qi, H., Xu, Y., Li, Z.: An efficient elephant flow detection with cost-sensitive in SDN. In: 2015 1st International Conference on Industrial Networks and Intelligent Systems (INISCom), pp. 24–28. IEEE, Piscataway, NJ (2015)
13. Amaral, P., Dinis, J., Pinto, P., Bernardo, L., Tavares, J., Mamede, H.S.: Machine learning in software defined networks: data collection and traffic classification. In: 2016 IEEE 24th International conference on network protocols (ICNP), pp. 1–5. IEEE, Piscataway, NJ (2016)
14. Rossi, D., Valenti, S.: Fine-grained traffic classification with netflow data. In: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, pp. 479–483. ACM Press, New York (2010)
15. Qazi, Z.A., Lee, J., Jin, T., Bellala, G., Arndt, M., Noubir, G.: Application-awareness in SDN. In: Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, pp. 487–488. ACM Press, New York (2013)
16. Shao, Y., Lin, J.C.W., Srivastava, G., Guo, D., Zhang, H., Yi, H., et al.: Multiobjective neural evolutionary algorithm for combinatorial optimization problems. IEEE Trans. Neural Networks Learn. Syst. (2021)
17. Nakao, A., Du, P.: Toward in-network deep machine learning for identifying mobile applications and enabling application specific network slicing. IEICE Trans. Commun. E101.B(7), 1536–1543 (2018)
18. Shabaan, M., Arshid, K., Yaqub, M., Jinchao, F., Zia, M.S., Bojja, G.R., et al.: Survey: smartphone-based assessment of cardiovascular diseases using ECG and PPG analysis. BMC Med. Inf. Decis. Making 20(1), 1–16 (2020)
19. Ambati, L.S., El.Gayar, O.: Human activity recognition: a comparison of machine learning approaches. J. Midwest Assoc. Inf. Syst. 2021(1), 4 (2021)
20. Ambati, L.S., El Gayar, O.: A comparative study of machine learning approaches for human activity recognition. Paper presented at the Fifteenth Midwest Association for Information Systems, 28–29 May 2020
21. Kobo, H.I., Abu Mahfouz, A.M., Hancke, G.P.: A survey on software-defined wireless sensor networks: challenges and design requirements. IEEE Access 5, 1872–1899 (2017)
22. Haque, I.T., Abu Ghazaleh, N.: Wireless software defined networking: a survey and taxonomy. IEEE Commun. Surv. Tut. 18(4), 2713–2737 (2016)
23. Ahmed, U., Lin, J.C.W., Srivastava, G.: Feedback learning in software-defined mobile network for resource aware load balancing under fault tolerance conditions. IEEE Instrum. Meas. Mag. 25, 32–37 (2022)
24. Cheng, C.-F., Srivastava, G., Lin, J.C.-W., Lin, Y.-C.: Fault-tolerance mechanisms for software-defined internet of vehicles. IEEE Trans. Intell. Transp. Syst. 22, 3859–3868 (2021)
25. Pradhan, B., Nandi, A., Hui, N.B., Roy, D.S., Rodrigues, J.J.: A novel hybrid neural network-based multirobot path planning with motion coordination. IEEE Trans. Veh. Technol. 69(2), 1319–1327 (2019)
26. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, pp. 1942–1948. IEEE, Piscataway, NJ (1995)
27. Pradhan, B., Srivastava, G., Roy, D.S., Reddy, K.H.K., Lin, J.C.W.: Traffic classification in underwater networks using sdn and data-driven hybrid metaheuristics. ACM Trans. Sens. Networks 18(3), 1–15 (2021)
28. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. 6(1), 58–73 (2002)
29. Liu, Z.G., Ji, X.H., Liu, Y.X.: Hybrid non-parametric particle swarm optimization and its stability analysis. Expert Syst. Appl. 92, 256–275 (2018)
30. Shao, H., Tang, L., Dong, L., Chen, L., Jiang, X., Wang, W.: A research of network applications classification based on deep learning. In: International Conference on Machine Learning and Intelligent Communications, pp. 13–21. Springer, Cham (2018)
31. Ramakrishnan, N., Soni, T.: Network traffic prediction using recurrent neural networks. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 187–193. IEEE, Piscataway, NJ (2018)
32. Wang, X., Chen, S., Su, J.: App-Net: a hybrid neural network for encrypted mobile traffic classification. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 424–429. IEEE, Piscataway, NJ (2020)
33. Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 43–48. IEEE, Piscataway, NJ (2017)

How to cite this article: Pradhan, B., Hussain, M.W., Srivastava, G., Debbarma, M.K., Barik, R.K., Lin, J.C.-W.: A neuro-evolutionary approach for software defined wireless network traffic classification. IET Commun. 1–8 (2022). https://doi.org/10.1049/cmu2.12548