International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Deep Learning Algorithm Using Densenet to Enable Big Data Analytics in Large WiFi Systems

[1]Doradla Bharadwaja, [2]R. Gayathri, [3]D. Sugumar, [4]D. R. Denslin Brabin, [5]Faiz Akram, [6]Mohd. Javeed Mehdi

Submitted: 03/10/2023 **Revised**: 22/11/2023 **Accepted**: 03/12/2023

Abstract: The increasing mobile device and unceasing traffic demand enables the deployment of large-scale WiFi systems that offers indoor coverage and high-speed connectivity. The large-scale deployment of WiFi system is an on-going research in wireless system due to its challenging heterogeneous nature of access points. Such access points undergo rapid challenges due to traffic conditions and traffic consumptions with rapidly increasing input data. On other hand, massive connection with heavy traffic laden from the WiFi devices poses increased pressure on backhaul network and reduces the Quality of Service by the users. We have developed using DenseNets that reduces the backhaul traffic due to the WiFi access points. The study explores wide deployment of data cache from massive access points for serving the several thousand active users. The study reduces the backhaul traffic using deep learning model that conducts statistical analysis on the collected user records. Extensive simulations are conducted to study the efficacy of the model that includes the cumulative distribution function per access point traffic/entropy and Jaccard similarity, caching resource utility and cache gain ratio.

Keywords: *Deep Learning, Densenets, Big Data Analytics, Wifi Systems*

1. Introduction

There has been an increase in the creation of mobile data across wireless networks due to the increasing use of mobiles. IEEE 802.11 networks are used by a substantial number of end-users in a variety of settings [1] [2]. Large-scale wireless networks like this one provide valuable information about users, their networks, their usage, and their mobility patterns. [3]

The monitoring of a wireless LAN is more difficult than that of a wired LAN because of the inherent limitations of wireless monitoring. In the wired world, methods such as measuring metrics after data has travelled across the network are tried and proven, but they do not indicate the current network state [4].

An active measurement is required in order to accurately assess a network current condition; this necessitates changes in the parameters being examined. When using indirect measures, such as access point counters or spectral sensors for capturing frame data, the information accuracy is reduced in several ways. When it comes to gathering data from networks, this does not modify the network state, but it may be used to create context-aware apps that can help with monitoring at all levels. Passive scanning of beacons, together with information about the received signal power, allows the identification of neighbouring access points and the inference of their physical distance and radio coverage area.

Use of user positioning or geolocation information helps to discover usage trends and spot abnormalities, such as, for instance, it is also possible to design more efficient networks by studying the data produced by the networks themselves [4].

By analysing a vast detect problems or performance decreases at specific network locations, and optimise channel allocation [5]. An examination of the wireless network environment is a significant challenge because of factors [6]. Ability to extract information about users and networks are hampered by the absence of network intelligence mechanisms.

Real-time network analysis necessitates the use of large data streaming processing techniques, which may not be appropriate for other applications. Data quality and their statistical distribution are unknown, so this is a problem.

A huge data processing platform processes a limited dataset at a time in batch processing, which is the opposite of what is done in stream processing. As a result, batch processing necessitates a lot of storage space and has a noticeable impact on reaction time. Each data sample is processed at a shorter latency, and there

[1]*Assistant Professor, Department of Information Technology, Prasad V. Potluri Siddhartha Institute of Technology, Andhra Pradesh, India. Email ID: bharadwaja599@gmail.com*
[2]*Professor, Department of ECE, Rajalakshmi Engineering College, Tamil Nadu, India. Email ID: gayathri.r@rajalakshmi.edu.in*
[3]*Associate Professor, Department of ECE, Karunya Institute of Technology and Sciences (Deemed to be University), Tamil Nadu, India. Email ID: sugumar.ssd@gmail.com*
[4]*Professsor, Department of Computer Science and Engineering, DMI College of Engineering, Chennai. Email ID: denscse@gmail.com*
[5]*Assistant Professor, Faculty of Computing and Informatics, Jimma Institute of Technology, Jimma University, Ethiopia. Email ID: akram.faiz@ju.edu.et*
[6]*Assistant Professor, Department of ECE, Gokaraju Rangaraju Institute of Engineering and Technology, Telangana, India. Email ID: javeed954@grietcollege.com*

are no memory limits on the storage when using streaming data processing.

The limitless number of samples and the demand for minimal delay while processing fresh samples necessitate that classic machine learning methods be modified for the streaming data. Streaming data leads to learning problems in deep learning algorithms. It is important to keep an eye out for changes in the statistical data distribution in deep learning applications. In this work, we have developed using DenseNets that reduces the backhaul traffic due to the WiFi access points.

The main contribution of the paper is given below:

- The study explores wide deployment of data cache from massive access points for serving the several thousand active users.
- The study reduces the backhaul traffic using deep learning model that conducts statistical analysis on the collected user records.
- Extensive simulations are performed.

2. Related Works

User mobility characterization applications are classified by the author of [6], attempting to forecast the person future location and mobility patterns, only the number of people going through a given location is taken into account. User mobility patterns are identified by doing a spatial-time analysis, which aims to discover the groups of users with similar characteristics and their future paths [7]. In [8] claim that in order to maximise the QoS of mobile users, next-generation communications will employ. Because of this, the authors advise that users should be able to easily move between different access points or base stations on their mobile devices. Using an MDP, the proposal selects the best networks to enhance QoS, they utilise an algorithm that uses reinforcement learning to determine a better option.

The Q-learning technique is in the article [9] to propose a method each user sets a bit in a bit vector, and the state set for Q-learning spectrum sensing contains all of the possible combinations of bits in that vector. The action set for Q-learning is indexed from 0 to 1, with 0 denoting an available state for secondary users and 1 denoting an unavailable state. When the action is in line with the channel occupation, the rewards are positive; otherwise, they are negative.

In addition, the Q-Learning method is used to implement bandwidth control for cloud providers. For one, they regularly adjust network infrastructure parameters to match service level agreements with clients, while for another, they maximise network occupancy and, consequently, income for the cloud provider.

Wireless networks have recently been studied using DL to learn more about their structure and behaviour. In order to investigate bi-modal data, created an algorithm based on DL. Devices' indoor location fingerprints are generated by their algorithm. Bi-modal data is used to train DL to develop feature-based fingerprints offline. In the deep autoencoder network, each point has its own unique fingerprint.

One solution uses an Auto Encoder, while the other uses Convolutional Neural Networks (CNNs) for indoor localization. The authors conclude that the Auto Encoder technique has a lower level of error. It employ DL to detect wireless channel activity more accurately and robustly. The key concepts are selecting high-quality Wi-Fi channels and switching between them to construct an extended channel. Using a recursive neural network model, the authors first look for trends in the usage of specific channels.

3. Proposed Method

Numerous computational deep learning models have been developed in this context to address a variety of workloads. Static large data can be processed with batch data processing techniques. In terms of batch data processing, Hadoop is the most commonly used tool. Analyzing network data necessitates the use of systems that can analyse data in real time.

3.1. DenseNet

It was set to 4 in this case because that what the ResNet-18 model suggested. Each dense block will have 128 more channels as a result of this study convolutional layer channel count of 32. As a result, the height and width are reduced by half, as well as the number of channels.

For functions, the study recalls the Taylor expansion. In relation to x = 0, the expression for the point is given below:

$$f(x)=f(0)+f'(0)_x+f''(0)_2!x^2+\ldots.$$

The main point is that a function can be decomposed

$$f(x)=x+g(x).$$

A basic linear term is created by ResNet, and a nonlinear term is created by ResNet. Is there a way for the study to catch information beyond the two terms? DenseNet was one such solution.

Differences are primarily based on the fact that outputs are concatenated rather than appended in DenseNet. As a result, after applying a series of more complex functions, the research performs a mapping to its values. DenseNet is the name given to this algorithm because the dependence block between variables gets quite thick.

The final link in this chain is tightly linked to all the links that came before it.

S`

Dense Layers:

It uses a modified activation, batch normalisation and convolution structure similar to ResNet. Convolution blocks with the same number of output channels make into a dense block. A convolution block input and output are combined in a forward propagation investigation, though.

Transition Layers:

Adding too many transition layers will result in an overly complicated model, since each thick block increases the number of channels. Transition layers are utilised to keep the model simple.

Training:

The study will reduce the input 224×96 because it is utilising a deeper network here.

3.2. Data Analytics and Classification

RSSI readings from several office Wi-Fi networks were used to generate the dataset. One-to-one computing is being implemented in Uruguay by this organisation. So, one of its most important duties is to ensure that all educational institutions in the country have access to Wi-Fi internet access.

Most Wi-Fi networks are found in public primary and secondary offices, and it is crucial to keep this in mind. These educational institutions are housed in a wide range of buildings, from century-old structures with multiple levels. Each building is normally covered by 5 or 6 APs on average, but about 20% of the buildings required more than 10 APs, as shown in the graph.

The algorithm is used to manage radio resources by the WLCs. Radio transmissions of NDP packets use a single radio chain with the utmost power and lowest data rate permitted for the channel/band. NDP packets are sent every 180 s by default on all channels. In the 2.4 GHz range, the AP goes off-channel roughly every 16 seconds to deliver an NDP packet; in the 5 GHz band, it goes off-channel every eight seconds. NDP data and RSSI for all received packets are forwarded to the WLC. The WLC takes five measurements each neighbour and averages them over a period of 15 minutes. Because it is busy and has less available spectrum than the 5 GHz band, the study will focus our attention on 2.4 GHz measurements. As a result, in order to build the conflict blocks, the study treat each AP as a node. As a result, each time stamp corresponds to a single directed block for each office.

This study relies on conflict blocks generated by Wi-Fi network data, which are defined in terms of their definitions, so the study will use them to examine the results. Using the Python package, the study were able to achieve this for each block by calculating the collection of selected features. The functions were used to compute the features in various preprocessing phases. All edges having an RSSI value lower than-80 dBm, which is considered the Channel Assessment criterion for APs, are first discarded from consideration. Edges between all of the nearby APs that do not impact one another are thus erased in this manner. The following features are then calculated:

Number of edges

Links with RSSI values greater than or equal to 80 dBi are counted as having this value. For each pair of APs that can see each other above the threshold, the symmetry should hold.

In-degree centrality

There was yet another preprocessing step required for the remaining features because the edge weights had to seem like a distance measure in those cases. In order to get the distance between APs, i.e., the adjacency matrix entry, the study use Eq.(1) to convert the RSSI values (dBm). This distance is then used in Eq. (1) to get the distance between APs.

$$w_{ij}=0.1(10-RSSI_{ij}) \qquad (1)$$

Each Wi-Fi network AP conflict block can be interpreted in many ways, and each feature picked has a possible interpretation. All nodes have an influence on their neighbors, which is reflected in all centrality measurements, regardless of the disparities between them. This means that in a Wi-Fi network, an AP with a high centrality value is more likely to be located in an area with higher levels of interference from its nearby APs. When it comes time to deal with key concerns, such as channel allocation and transmission power regulation, it important to know how many channels each AP needs to avoid interference.

As a result, most of the characteristics do not have a high connection with each other, preventing the feature vector from being overly saturated with redundant data. The final step in data exploration and feature analysis is PCA decomposition, which digs deeper into the significance of each characteristic found in our sample set.

4. Results and Discussions

In this section, the study validate the proposed DenseNet based on several network metrics that include cumulative distribution function per access point traffic/entropy and Jaccard similarity, caching resource utility and cache gain ratio. The model is simulated in python simulator with eclipse IDE on a high end computing engine. The proposed model is compared with conventional deep learning models including Deep Auto Encoders (DAE) and Residual Neural Network (ResNet).

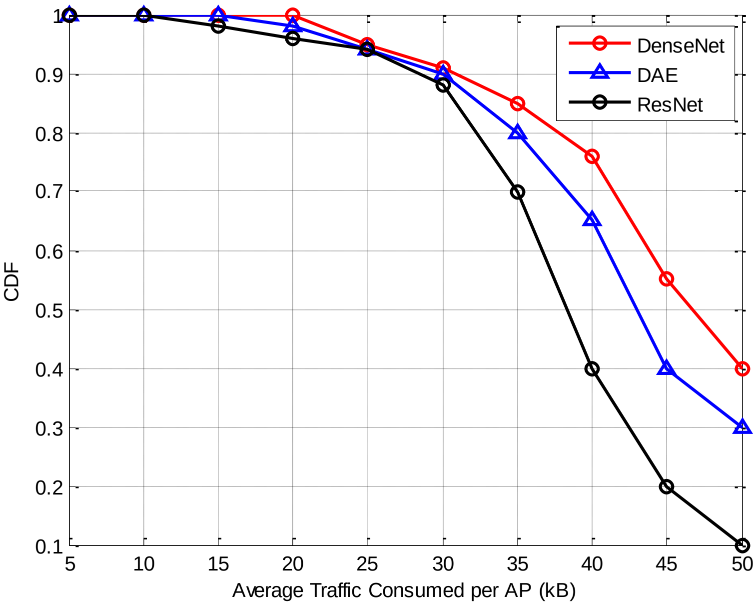

Fig 1: CDF vs. Consumed Traffic

For a successful cache deployment, it is important to know where the most traffic is consumed and the consumed traffic. The study collects all user association records from WiFi to capture the attribute. Each user association records are logged for a week on the platform, which can collect raw management data from all APs in real time.

As seen in Figure 1, the findings of the CCDF in two distinct months are very similar since the two curves are so close together. It suggests that if the first deployment method is well-planned, the caching gain can last for a long time because the demonstrated consistency of traffic consumption every day is demonstrated to be consistent. Second, the amount of AP traffic consumed is spread out over a wide range of traffic volumes.

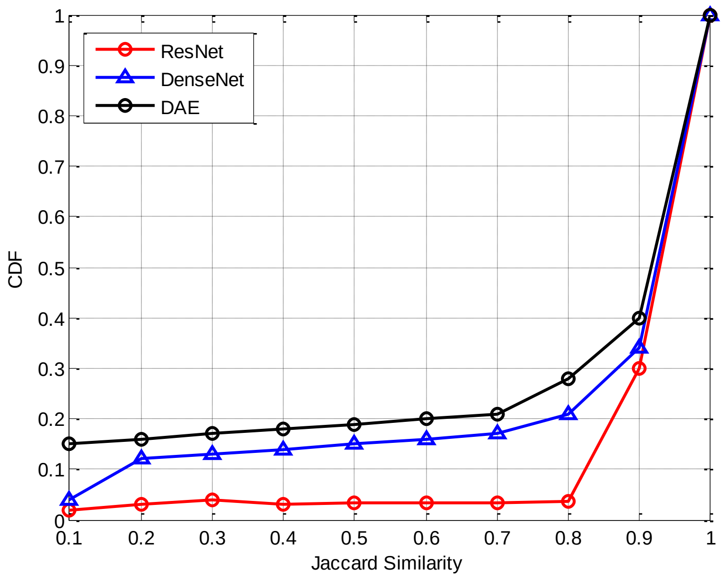

Fig 2: CDF vs. Jaccard similarity score

Figure 2 shows a strong Jaccard similarity score between popularity ranking and traffic consumption. According to the findings, AP popularity is positively connected with its traffic consumption. Since more traffic is generated at APs, caching services can be provided to a greater number of users if the cache deployment method is intended to cache this traffic as well.

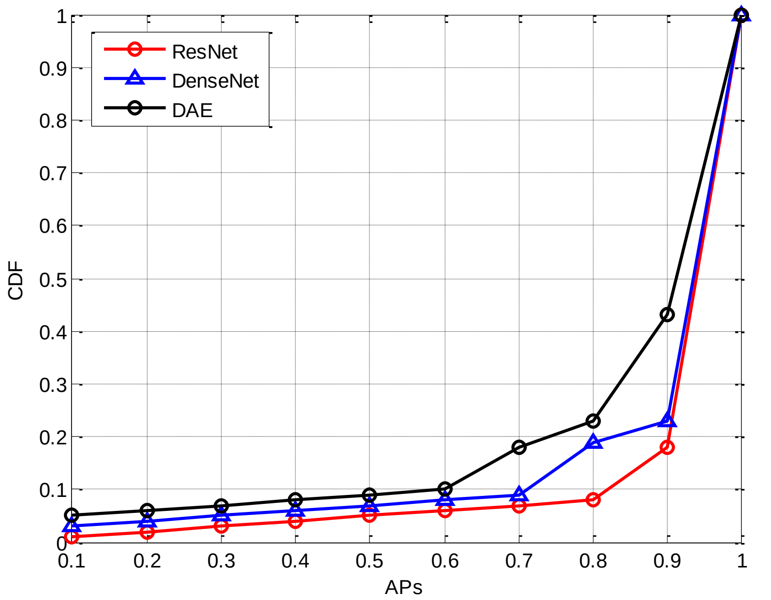

Fig 3: CDF vs. APs

The CDF of APs in buildings and edge nodes is shown in Figure 3 of the study. Edge nodes are used instead of APs for caching so that all AP-associated clients can access and benefit from the same cache resource. It has the potential to dramatically increase the utilisation of cache resources. A specialised cloud centre is not necessary because the edge node is tiny enough to only have a few APs, for example. Edge node caching is considered easier to create and maintain than a data centre in every building.

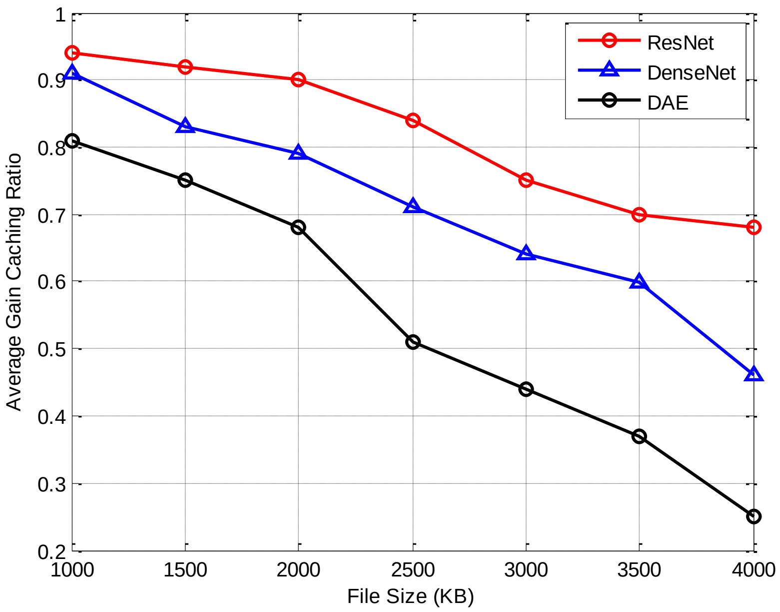

Fig 4: Average Caching Gain Ratio

The caching gain ratio plot is shown in Figure 4, and it reveals the following three key findings: When two curves are close together, the suggested DenseNet is able to achieve an almost ideal level of performance.

5. Conclusions

In this paper, DenseNet reduces the backhaul traffic in the WiFi access points and it explores wide data cache deployment from massive access points for serving the several thousand active users. The study reduces the backhaul traffic using deep learning model that conducts statistical analysis on the collected user records. The cumulative distribution function per access point traffic/entropy and Jaccard similarity, caching resource utility and cache gain ratio.

Pattern recognition training in the future will use labelled patterns to train algorithms. Aside from the computational costs and time involved in labelling. To further complicate things, labels are affected by the speed and volume of data. One of the most pressing training-related issues is the issue of overfitting, which remains an open question.

References

[1] Yang, H. H., Xu, C., Wang, X., Feng, D., & Quek, T. Q. (2021). Understanding age of information in large-scale wireless networks. *IEEE Transactions on Wireless Communications*, *20*(5), 3196-3210.

[2] Lv, Z., Lou, R., Li, J., Singh, A. K., & Song, H. (2021). Big data analytics for 6G-enabled massive internet of things. *IEEE Internet of Things Journal*, *8*(7), 5350-5359.

[3] Cheng, S., Ma, L., Lu, H., Lei, X., & Shi, Y. (2021). Evolutionary computation for solving search-based data analytics problems. *Artificial Intelligence Review*, *54*(2), 1321-1348.

[4] Shapsough, S., Takrouri, M., Dhaouadi, R., & Zualkernan, I. A. (2021). Using IoT and smart monitoring devices to optimize the efficiency of large-scale distributed solar farms. *Wireless Networks*, *27*(6), 4313-4329.

[5] Covert, M. W., Gillies, T. E., Kudo, T., & Agmon, E. (2021). A forecast for large-scale, predictive biology: Lessons from meteorology. *Cell Systems*, *12*(6), 488-496.

[6] Midoglu, C., Kousias, K., Alay, Ö., Lutu, A., Argyriou, A., Riegler, M., & Griwodz, C. (2021). Large scale speedtest experimentation in Mobile Broadband Networks. *Computer Networks*, *184*, 107629.

[7] Fan, C., Yan, D., Xiao, F., Li, A., An, J., & Kang, X. (2021, February). Advanced data analytics for enhancing building performances: From data-driven to big data-driven approaches. In *Building Simulation* (Vol. 14, No. 1, pp. 3-24). Tsinghua University Press.

[8] Asadianfam, S., Shamsi, M., & Kenari, A. R. (2021). TVD-MRDL: traffic violation detection system using MapReduce-based deep learning for large-scale data. *Multimedia Tools and Applications*, *80*(2), 2489-2516.

[9] Yin, L., Lin, N., & Zhao, Z. (2021). Mining daily activity chains from large-scale mobile phone location data. *Cities*, *109*, 103013.