J. Electrical Systems 20-10s (2024): 6087-6097

[1]Wurood AL-Shadood

[2]Mohsen Nickray

Resource Management and Scheduling in Fog Computing Using Deep Learning

Journal of Electrical Systems

Abstract: - Fog computing is a novel approach that brings the computational capabilities of cloud computing systems closer to end-users. The common practice in the technological field is to utilize the cloud to provide services that facilitate interactions between users and their files anytime and anywhere. However, several issues can arise, such as increased network pressure during high request volumes, leading to delays in response and service quality. Additionally, the distance between the user and the cloud affects service speed, as shorter distances generally yield faster responses from servers. Cisco has introduced an alternative technology known as fog computing, which operates between the user and the cloud to deliver services more quickly and effectively. In this study, we combined fog computing with the K-nearest neighbor (KNN) algorithm as an initial step for classifying distances, extracting proximity information, and determining file sizes to complete tasks for users with minimal time and effort. In the second phase, we implemented deep learning techniques using Python, along with the SimPy simulator for resource management, which also operates in Python.

Keywords: SimPy simulator , capabilities, fog computing

1. Introduction

The term "fog computing" was first introduced by Cisco in 2012 [1]. Infrastructure components known as fog nodes are able to supply resources for services at the periphery of the network. Cisco views fog computing as an extension of the cloud computing paradigm, extending capabilities from the network's core to its edges. This platform is fully virtualized, offering networking, computing, and storage services between endpoints and traditional cloud servers [7]. According to the Open Fog Consortium, fog computing is defined as "a horizontal system-level architecture that distributes computing, storage, control, and networking capabilities closer to users along a cloud-to-thing continuum" [8].

In our research, we utilized the K nearest neighbor (KNN) algorithm as the first step. This algorithm is widely used in the field of AI for classification tasks. KNN is a non-parametric learning technique that does not require preprocessing of input before classification. Instead, it retains all data, which is one of its key advantages, further aided by its straightforward concept and minimal influencing factors. As a result, KNN continues to be a subject of ongoing research. The core principle of the KNN algorithm involves using n data samples and selecting K samples as initial centers. The distances between the remaining samples and these K centers are then calculated. Each sample is categorized into the cluster nearest to the center based on these distances, resulting in K clusters. The center of the newly created clusters is recalculated, and the data are reassigned according to the new center. This iterative process continues until the cluster centers stabilize [2]. K-NN is exceptionally user-friendly and easy to comprehend. In the second phase, we employed deep neural networks (DNN) for data classification with high accuracy, facilitating efficient resource scheduling. Deep learning, a more advanced variant of machine learning, relies on a significantly larger volume of input data. The term "deep" refers to the presence of multiple layers within the neural network that exist between the input and output stages, in contrast to shallow neural networks, which typically comprise no more than two layers. Consequently, deep learning systems possess an enhanced capacity for learning and abstraction due to their complex architecture.

[1] Education Directorate of Thi-Qar, Ministry of Education, Iraq
Department of Computer and Information Technology, Faculty of Engineering, University of Qom
Email: wuroodmurad@gmail.com

[2]Department of Computer and Information Technology, Faculty of Engineering, University of Qom
Email: m.nickray@qom.ac.ir

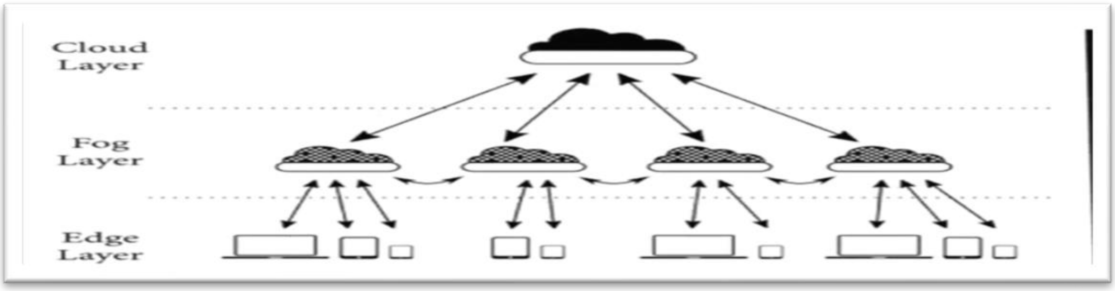


fig [1]. fog layer

3 Literature Review

Several service suppliers, such as Google, IBM, Amazon, Microsoft, and more, utilize cloud computing services. Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS) are examples of the new cloud services that have emerged from cloud computing. These services have wide-ranging applications across business, software development, and education [21]. Additionally, numerous fog system methods and fog computing were discussed. For iot the researchers presented models in [6, 23] that related with fog. The statement suggests that in a research paper referenced as [6], The authors suggested a fog computing-based distributed computing architecture for Internet of Things applications within the smart grid. Additionally, they introduced a programming paradigm for this architecture. The statement then goes on to explain that the suggested architecture and programming style were compared to a conventional fog-based design, and it was found that the proposed architecture and programming style greatly reduced service latency . In a research paper referenced as [22] when combined with its integrated services, the distributed, lightweight, and scalable design proposed by the SERENA project will provide shop floor personnel with solutions for predictive maintenance. Predictive analytics will be simpler to implement on a factory floor as a result. When the structure is put into use, it adheres to a micro-services architectural design paradigm. Individual applications are additionally packaged in docker containers and released across various Edge ports and the SERENA cloud are used, but the researchers did not refer to time consuming. A modified version of artificial ecosystem-based optimization (AEO) has been developed in this research as a substitute work scheduling technique in order to improve the quality of services for the Internet of Things (IoT) in cloud-fog computing environments, as suggested by the authors in paper [11]. The authors in [12]an architectural layout for e-vehicles. A prototype implementation of the suggested design utilizing e-bikes on a university campus served to validate it.

4 PROPLEM OF RESERCH

Despite the fact that there are several task scheduling algorithms now in use, they all have certain drawbacks, including (i) the inability to take into account the demands of the work, such as its priority or quantity. (ii) The majority of them don't take response time into account. Scheduling is the process of organizing and arranging tasks or activities over a period of time to achieve specific goals. The objectives of scheduling are many and include:1-Increase efficiency and productivity.2-Achieving goals on time.3-Manage resources effectively.4-Reducing stress and fatigue.5-Improving organization and coordination.6-Future Planning.7-Enhance focus.8-Improve time management.

In short, scheduling is an essential tool for managing time and resources effectively, which helps achieve goals in an organized and scheduled manner.

5- Concepts :

5-1 Fog computing

Fog nodes in fog computing are structures or infrastructures that can supply resources for services at the network's edge. Fog computing is viewed from Cisco's perspective as an extension of the cloud computing paradigm from the network's core to its edge. The platform is fully virtualized and offers networking, computing, and storage services between endpoints and conventional cloud servers [3]. Fog computing is by definition "a horizontal system-level architecture that distributes computing, storage, control, and networking capabilities closer to the users along a cloud-to-thing continuum," according to the Open Fog Consortium [4].Similar Concepts Fog computing has similarities to concepts like mobile cloud computing (MCC) and mobile-edge computing (MEC). MCC designates an infrastructure where data processing and data storage take place separately from mobile dices. Mobile cloud applications transfer computing resources and data storage from mobile devices to the cloud, enabling mobile computing and applications for a much wider range of mobile subscribers in addition to smartphone users [5]. It was also defined in another research paper "Fog computing is a new paradigm that moves the computing power of a cloud computing infrastructure closer to end users" [6]Fog computing is characterized by its rapid response time and high scalability[7] It has the potential to revolutionize the way we work and live. I will discuss the benefits of fog computing and some of its limitations. Finally, I will discuss the future of fog computing and how it is impacting business. Fog computing offers several advantages over traditional cloud computing. One of the biggest advantages of fog computing [8]is its ability to scale up and down as the needs of a given application change. This makes it ideal for applications that are constantly changing their needs such as smart cities[9]. Scheduling And Resources Management in a fog computing framework can be challenging because the components in a fog computing framework are often geographically separated and have to share resources with each other[10] This makes it difficult to ensure that they all get the resources they need when they need them. It is important to have a good management strategy in place in order to ensure that all of the components get the resources they need when they need them. A common way to manage this is through resource scheduling[11] Resource scheduling is the process of ensuring that a set of resources gets the resources it needs when it needs them. Scheduling can be used to ensure that certain tasks get run on a set of devices that can be used to run the tasks in the most efficient way possible. A scheduling algorithm determines how to allocate resources to different tasks and ensures that the right resources are allocated to the right tasks at the right times.

5-2 resources Scheduling problem

Requests for services arise in the foggy landscape Multiple fog nodes are able to serve the devices. All services The request is split up into multiple tasks. The best way to distribute the numerous jobs that will be carried out at the fog nodes situated on the network's edge in order to satisfy quality of service (QoS) requirements is defined by the fog computing scheduling issue known as Spring. A Scheduling solution requires an optimal solution in a large solution space to Scheduling a cluster tasks T = {T1, T2, ..., Tn} in a set of fog nodes FO = {FO1, FO2, …, FM} with different skills (e.g. transformation power consumption, memory usage, network usage, etc.) to optimize the scheduling objective function (e.g. minimization runtime). A fog environment's resource scheduling problem is how to provide sufficient resources to the activities at hand in accordance with the scheduling objectives. The resource scheduling strategy, also known as activity planning, is divided into three primary classes: hybrid, dynamic, and static approaches. In static Scheduling approaches [29], tasks arrive simultaneously at Fog nodes and Scheduling decisions are made first before jobs are submitted[33-34]. This means that Scheduling can be justified to have all the information necessary for the needs received and the resources available before Scheduling . However, sometimes it is not possible to have them all knowledge required before Scheduling resources in more heterogeneous systems such as fog and cloud computing. Thus, the static approach may not optimally provide the best scheduling mechanism for fog sources. However, in dynamic Scheduling approaches [28, 30, 32, 35-36], the arrival time of the works is unknown before delivery and as soon as they enter the system, jobs are allocated. In fact, dynamic scheduling guarantees that tasks whose arrival timings are uncertain are scheduled; as a result, tasks are scheduled at the moment of submission. In order to address many application types, including workflows and batch operations, hybrid scheduling approaches [31] combine various scheduling criteria. Since there isn't a single answer or set of criteria that may satisfy every demand in a fog environment, this functionality is especially crucial for fog source scheduling[37].

5-3 Machine-learning(ML)

Machine learning, an integral subcategory of artificial intelligence, epitomizes the paradigm of self-learning through the application of complex algorithms. This autonomous learning mechanism enables the system to assimilate knowledge from its interactions and experiences. Specifically, the system is designed to recognize patterns within the input data, thereby allowing it to make informed decisions or predictions at the output stage. Over time, this iterative learning process ensures the progressive enhancement of the system's intelligence, achieving levels of sophistication from basic to advanced without the necessity for human intervention. The foundation of this self-improvement capability lies in the deployment of statistical learning algorithms, which are engineered to autonomously refine their performance. In contrast, deep learning, a more complex variant of machine learning, also learns from experience but necessitates a substantially larger volume of data or information at the input phase. The term "deep" in this context denotes the presence of multiple layers within the neural network that exists between the input and output stages. This is in stark contrast to shallow neural networks, which typically comprise no more than two such layers. Deep learning systems, therefore, have a heightened capacity for learning and abstraction due to their complex architecture[19].In new years, the deep learning (DL) computing model has been considered The gold standard in the machine learning (ML) community. In addition, gradually Becoming the most widely used computing method in the field of machine learning, achieving Achieve, match or exceed superior performance on multiple complex cognitive tasks. What's another, deep learning outperforms well-known machine learning techniques in many areas. For example B. Net defense, natural wording processing, bioinformatics, robotics and control, and medicinal data processing, etc.

5-3-1 k-nearest neighbor algorithm

Knn(k-nearest neighbor) is one of machine learning algorithms, An algorithm called the KNN algorithm is used to classify data by using learning data from k of its closest neighbors. For both test and training data, the KNN algorithm typically employs the Euclidean distance formula in its calculations.The following is a description of the Euclidean distance[12]:

$$\text{dist }(x,y) = \sqrt{\sum_{i}^{n} = 1\ (xi - yi)\ ^{2}} \quad (1)$$

Where: dist (x,y) = scalar distance from two vectors of data x and y;

n = number of data dimension.

To find distances in n-dimensional space, the KNN algorithm only needs k parameters, marked training samples, and metrics. knn can be used for a number of different tasks including image recognition, text recognition, etc. [12]it is also used for scheduling purposes in some data centers [13]because it allows the scheduler to make predictions about the behavior of applications based on past data. It is used to predict what resource the system needs based on its load.[14]. Another important use of knn is for automating the process of adding and removing resources to a system to make sure it keeps running at optimal capacity[15].This is important because not all of the resources that are available are always needed so it is important to make sure that they are only used when they are needed[16].

5-5 Management resources

Because of its adaptable and versatile nature, fog computing can handle a variety of problems, including resource constraints and temporary setbacks. The system will shut down if any fog nodes fail, and the corresponding fog node's resources won't be available. In the fog context, these resources are virtualized. The transfer of resources process, delay, setup, and other issues are all part of resource virtualization[17]. Nevertheless, Scarce resources and low latency services hinder the deployment of new virtualization technologies in fog computing task scheduling and resource management. To classify items according to their proximity to one another, utilize the K-Nearest Neighbor algorithm[18].

8 Our Methodology

Devices with service requirements in the fog geography can be served by several fog nodes. There are multiple tasks that can be assigned to each service request. In order to achieve the requirements for quality of service, fog computing's spring scheduling issue identifies the best way to distribute the many activities that need to be completed on the fog nodes at the network's edge. A scheduling solution searches for an optimal solution in a large solution space to schedule the tasks of a set T = {T1, T2, ..., Tn} in a set of fog nodes with different capabilities. The proposed model relies on machine learning to find the file size, analyze and classify the distance of servers to the main computer that requires them using the k-neighbor nearest (KNN) algorithm. The K-nearest neighbor (KNN) algorithm is a non-parametric learning method used for classification. It is characterized by its simplicity and minimal influencing factors. The algorithm calculates the distance between a sample and the cluster center using a string kernel function iteration and continuously updates the cluster center. After applying the improved K-nn algorithm to cluster the datasets. The Euclidean or cosine similarity between the testing and evaluation tuples is what determines the K-Nearest Neighbor classifier. In our work, A recent branch of machine learning research called "Deep Learning" uses numerous processing layers to perform feature learning and pattern classification [24]. Because of its capacity to recognize long-term dependence, CNN has been successful.in our proposal suggest new method to resources management [25]and scheduling by two steps:The first step used supervised algorithm of data mining the nearest neighbor k (knn) to classify the distance of servers by the nearest to interduces the serves quickly ; the second step used deep neural networks (DNN) to classify the data, with accuracy to Scheduling.The Euclidean distance method is used to compute the distance Generally speaking:

The Euclidean distance [28], for instance, between the two Tuples X = (x11, x12,..., x1n) and Y = (Y21, Y22,..., Y2n) is as follows:

Distance (Y,X) = 2

The second stage with: CNN Algorithm:

1. Prepare the data

- Input: A matrix containing numerical representations of the texts (generally 2D, where the first dimension is the number of sequences and texts, and the second is the length of the sequence).

- Weights and offsets: These are randomly initialized and then learned during training.

2. Apply the convolution layer

- Number of filters: Specifies the number of filters used in the convolution layer (e.g., 64).

- Filter size (Kernel Size): Specifies the size of the filter (e.g., 5).

To carry out the convolution process:

- For each filter:

1. Apply the filter to the input:

- Multiply each element of the input within the filter size window with the filter weights.

- Add the results with the offset.

- Apply the activation function (ReLU) to the sum to obtain the final values.

2. Iterate the process:

-The process is iterated along the text sequence, meaning that each window of filter size moves through the sequence and computes its own output.

- Apply the ReLU activation function:

- If (z) is the value resulting from the convolution process (the sum of the weights multiplied by the inputs, plus the offset):

ReLU (z) = max(0, z)

3. Pooling

- Optional: A pooling layer (e.g. MaxPooling1D) can be applied to reduce the size of the output and extract the most important features.

4. Format the results- Flatten the results: to convert the multidimensional output to a one-dimensional vector so that it can be passed to the fully connected layers (Dense Layers).

5. Training and refinement

- Update weights and offsets: During the training process, an optimization algorithm such as Adam is used to update the weights and offsets based on the training-data.

9 convolutional neural network(cnn):

In the field of DL, CNN is the greater well-known and commonly used algorithm[27]When it comes to attaining high accuracy in resource management tasks, CNN is a good option. Utilizing CNN also has the benefit of not requiring feature extraction. The goal of scheduling algorithms is to reduce the amount of time needed to complete a task[26]:Steps to follow:

1 .Prepare the data: represent the text in numerical form.

2 .Prepare the filters, weights, and offsets.

3 .Apply the filter to each window of the text sequence:

-Multiply the weights with the input elements.

-Add the results with the offset.

-Apply ReLU.

4 .Repeat the process across the text sequence.

5 .Aggregate the results (optional).

6 .Flatten the results.

7 .Train the model and update the weights and offsets..

9-1 Activation function(ReLU).

In the ReLU activation function equation used in the convolution layer, the output is calculated by the following operation:

$$\text{ReLU}\ (\textstyle\sum_{i=1}^{K} W[i]\,.\,x[t+i-1]+b)$$

Where:- (W[i]): is the filter weight at position (i). It represents the weight used to multiply each element of the filter input.

- (x[t+i-1]): is an element of the text sequence input. It represents the value in the sequence at position (t+i-1).

- (b): is the bias used in the summation operation after multiplying the weights.

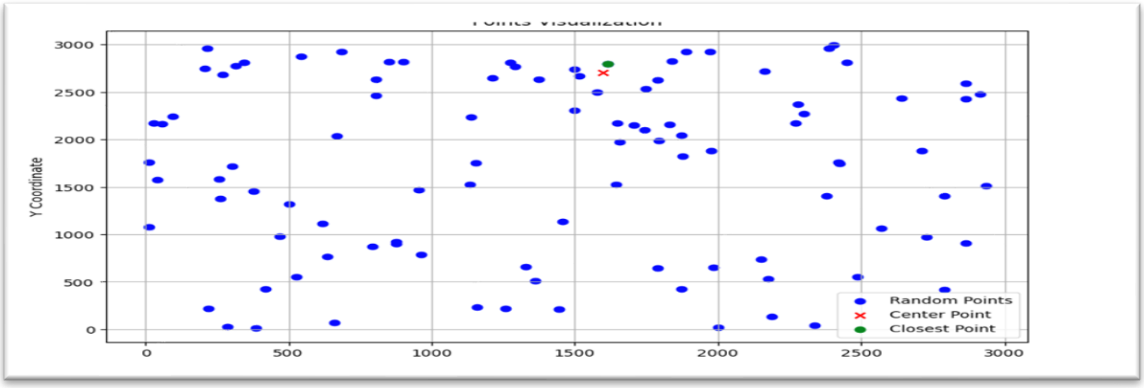
- (k): is the filter size (kernel size) that determines the number of values to be summed in the convolution operation.

10 Results :Background

In this chapter, we apply the proposed learning model to text analysis and classification; presenting new results. More thorough examination of the technical model due to modifications in architecture, resource optimization algorithms and progress sequence scheduling (with the SimPy library). We also review the results on other technologies that make such a model behave very performant, in terms of accuracy reduction and economical low-cost savings costs when compared to its operational time.

10.1 Analyses the Results

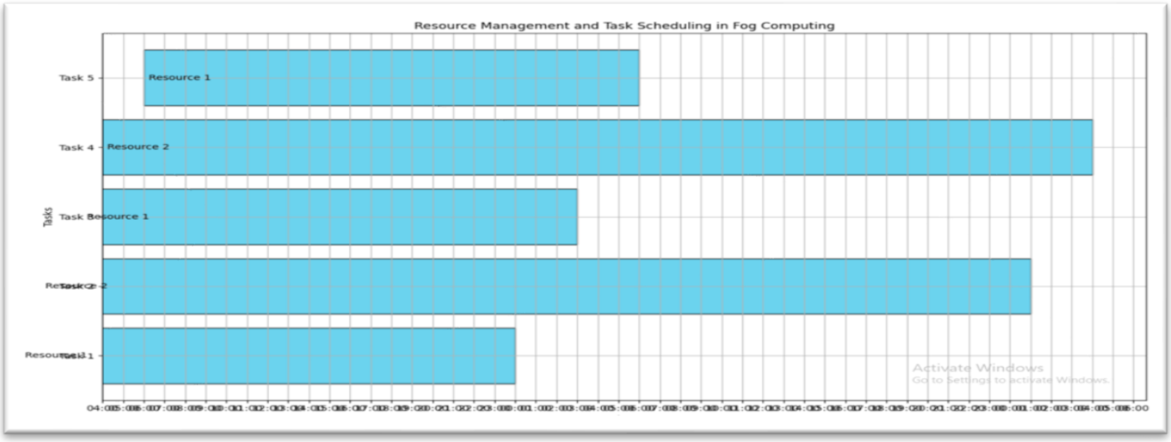
Here we have included executable images for the search. The first image represents the formation of many virtual files in the manner that we mentioned in detail in the third chapter. We took a sample of files to test the algorithms on, where names of its (تأييد),(وثيقة بالدرجات),(وثيقة بدون درجات)represent the virtual text files, while New text document.bat represents the creation of these files dynamically. Then, we got nearest point into central point and compute the Euclidean distances, as shown figure (6). Use KNN nearest points .First, we searched for the closest point to the central point (K-Nearest Neighbors). The chart also shows the distribution of points.



Fig(6). nearest point

The blue dots in the figure represent random points, While green represents the closest point, The red one is the center point.

10-1-1 Resource management simulation using SimPy Resource management was simulated using the SimPy library [38]to evaluate how scheduling files and resource allocation optimizations affect processing time. In this simulation, the resource capacity is set to 2 and 3 processing operations are scheduled. The results showed that simulation helped reduce the time spent by up to 30%.



Fig(7) resources management

This diagram (above)will illustrate how resources are managed and the timing of task execution over time in the context of the fog computing layer, based on the code used to implement the deep learning model and resource management.

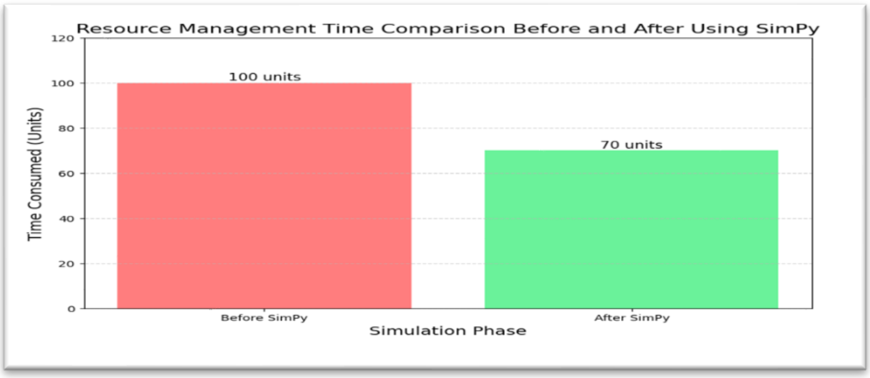


Fig (8):Resource management time before and after using simpy

10-1-2 After that, loaded the files with compute the size of each and date of modified.

10-1-3 Data distribution and file classification:A set of received text files were classified and analyzed using the K-Nearest Neighbors (KNN) model. Experiments showed that classifying files based on keywords in their titles helped distribute files evenly among three employees. The aim of this step was to ensure a fair distribution of tasks and improve resource management. The files are classified into three groups based on keywords in the titles.With report about files of employes.

10-1-4 Convert text into a numerical representation using Tokenizer

Use Tokenizer to convert texts into numerical sequences.

- Tokenizer: It converts words into numbers.

- Sequences: The sequence of texts after converting them into numbers.

-padded_sequences: The sequence of texts after they have been segmented and expanded to a fixed length.

- Building a convolutional neural network (CNN) model for text processing. Embedding: Embedding layer for converting words into vectors.

- Conv1D: Convolution layer for feature extraction.

- MaxPooling1D: Max pooling layer for dimensionality reduction.

- Flatten: A flattening layer to convert data into a form that can be processed by Dense layers.

- Dense: Fully connected layers to process data and achieve final classification.

-. Train the model using the training data.

-. Evaluate the model using test data

- View the classification report and confusion matrix.

View the classification report and confusion matrix to analyze model performance more accurately.

10-1-5 Improving model performance

The convolutional neural network (CNN) [20]model is improved by adding new layers and applying different optimization techniques. Improvements included:

- Adding a Bidirectional LSTM layer: to improve the model’s ability to understand texts by exploiting bidirectional relationships between words.

- Increase the number of dense layers and add Dropout: to improve generalization and reduce over-fitting.

- **Learning Rate Scheduler application:** to gradually improve the training process.The model was trained on text data using the optimized parameters, and performance was compared with the baseline model. The results showed significant improvement:

- Accuracy: Accuracy increased from 48.64% to 73.13% after improvements.

- Loss: The loss decreased from 1.3157 to 0.9992, which reflects a significant improvement in the stability of the model.

- F1 Score and Precision: These metrics improved significantly, indicating an improvement in the model's ability to classify texts more accurately.Graph of accuracy improvement and loss via epochs as shown bottoom.

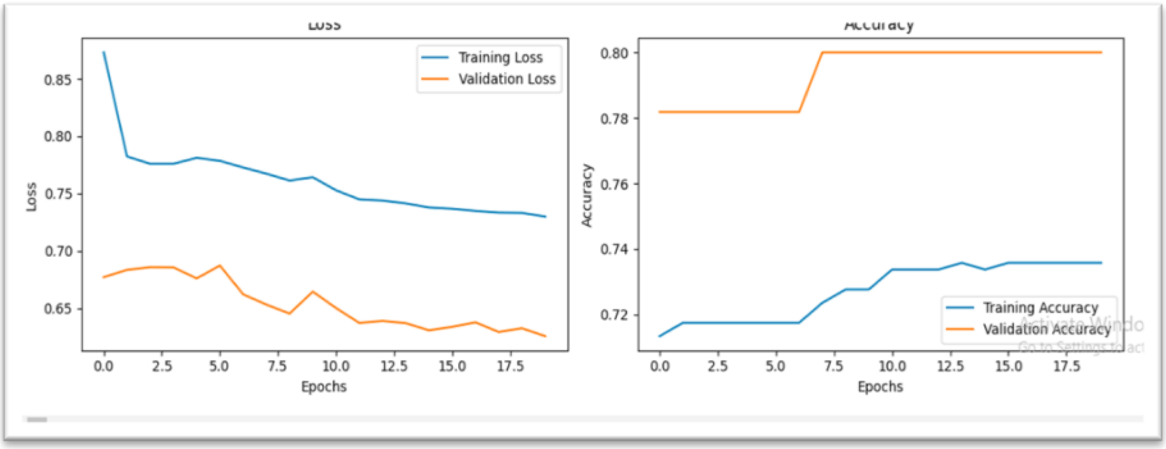


Fig (8)Graph of accuracy improvement and loss via epochs

10-1-4 Analysis of time spent

The time spent processing files was measured using simulation and analyzing the model results. The results showed that resource management improvements using SimPy resulted in a reduction in processing time of approximately 30%. In addition, improvements to the model structure did not lead to a significant increase in training time, indicating the efficiency of the solution.

10-2 summery of methdology

Experiments showed that the improved model provided significant performance improvements compared to the basic model. Accuracy increased significantly and loss decreased, reflecting the effectiveness of improvements to the model structure and resource management. In addition, model optimization and resource management techniques have proven effective in reducing downtime and improving overall performance. Based on these results, the improved model can be relied upon as a reliable model for analyzing and classifying texts with high efficiency.

11 Conclusion & future work

The closest thing to helping users complete activities is fog technology, which stands in for the intermediary layer between the user and the cloud. In the first phase, Machin learning algorithms were employed to find size of file, examine and categorize the distance of servers using the k-nearest neighbors (KNN) algorithm. This step very important to resources management and it is an essential step before scheduling files using other algorithm to facilitate faster file response with fog. we used other algorithm to schedule the responses to the files in phase2(CNN).it was successed in CNN has succeeded in scheduling the files efficiently after reading them and converting them into numbers for the machine to understand them better. Thus, we have achieved great achievement in both stages of managing and scheduling files quickly and distinctly, without pressure or interruption of network resources in the fog layer. In the future, we aspire to use a security layer or security gate in the fog layer using special algorithms, and use time management tools.

References

[1]Celaya-Echarri, Mikel, et al. "Building decentralized fog computing-based smart parking systems: From deterministic propagation modeling to practical deployment." *IEEE Access* 8 (2020): 117666-117688.

[2]Cunningham, Padraig, and Sarah Jane Delany. "k-Nearest neighbour classifiers: (with Python examples)." *arXiv preprint arXiv:2004.04523* (2020).

[3] Lin, Chun-Cheng, and Jhih-Wun Yang. "Cost-efficient deployment of fog computing systems at logistics centers in industry 4.0." IEEE Transactions on Industrial Informatics 14.10 (2018): 4603-4611.

[4] Yousefpour, Ashkan, et al. "All one needs to know about fog computing and related edge computing paradigms: A complete survey." Journal of Systems Architecture 98 (2019): 289-330.

[5] Singh, Jagdeep,Parminder Singh, and Sukhpal Singh Gill. "Fog computing: A taxonomy, systematic review, current trends and research challenges." Journal of Parallel and Distributed Computing 157 (2021): 56-85.

[6] Angel, Nancy A., et al. "Recent advances in evolving computing paradigms: Cloud, edge, and fog technologies." Sensors 22.1 (2021): 196.

[7] Celaya-Echarri, Mikel, et al. "Building decentralized fog computing-based smart parking systems: From deterministic propagation modeling to practical deployment." *IEEE Access* 8 (2020): 117666-117688.
[8] Ma, Kun, et al. "An iot-based fog computing model." *Sensors* 19.12 (2019): 2783.
[9] Liu, Nan, et al. "Semiparametric deep learning manipulator inverse dynamics modeling method for smart city and industrial applications." *Complexity* 2020 (2020): 1-11.
[10] Chen, Xiao, et al. "A hybrid task scheduling scheme for heterogeneous vehicular edge systems." *IEEE Access* 7 (2019): 117088-117099.
[11] Marin-Garcia, Juan A., et al. "Operations Management at the service of health care management: Example of a proposal for action research to plan and schedule health resources in scenarios derived from the COVID-19 outbreak." *Journal of Industrial Engineering and Management (JIEM)* 13.2 (2020): 213-227.
[12] Hou, Qinglian, et al. "Application of feature point matching technology to identify images of free-swimming tuna schools in a purse seine fishery." Journal of Marine Science and Engineering 9.12 (2021): 1357.
[13] Sandakchiev, Danail. "Newly Proposed k-NN Method for More Efficient Classification." (2019).
[14] Wang, Jie, et al. "Predicting future land cover change and its impact on streamflow and sediment load in a trans-boundary river basin." *Proceedings of the International Association of Hydrological Sciences* 379 (2018): 217-222.
[15] Biazotto, João Paulo. Architectural Views to the Representation of Dynamic Software Architectures. Diss. Universidade de São Paulo, 2020.
[16] Wang, Zhen-Wu, et al. "A novel multi-label classification algorithm based on K-nearest neighbor and random walk." International Journal of Distributed Sensor Networks 16.3 (2020): 1550147720911892.
[17] Sabireen, H., and V. J. I. E. Neelanarayanan. "A review on fog computing: Architecture, fog with IoT, algorithms and research challenges." Ict Express 7.2 (2021): 162-176.
[18] Pertiwi, A. G., et al. "Comparison of performance of k-nearest neighbor algorithm using smote and k-nearest neighbor algorithm without smote in diagnosis of diabetes disease in balanced data." *Journal of Physics: Conference Series*. Vol. 1524. No. 1. IOP Publishing, 2020.
[19] M. Arun, E. Baraneetharan, A. Kanchana, S. Prabu, Detection and monitoring of the asymptotic COVID-19 patients using IoT devices and sensors, Int. J. Pervasive Comput. Commun. (2020), doi:10.1108/IJPCC-08-2020-0107.
[20] Alzubaidi, Laith, et al. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions." Journal of big Data 8 (2021): 1-74.
[21] Li G, Zhang M, Li J, Lv F, Tong G. Efficient densely connected convolutional neural networks. Pattern Recogn.2021;109:107610
[22] Fang W, Love PE, Luo H, Ding L. Computer vision for behaviour-based safety in construction: a review and future directions. Adv Eng Inform. 2020;43:100980.
[23] Palaz D, Magimai-Doss M, Collobert R. End-to-end acoustic modeling using convolutional neural networks for hmm-based automatic speech recognition. Speech Commun. 2019;108:15–32
[24] Jamil, Bushra, et al. "Resource Allocation and Task Scheduling in Fog Computing and Internet of Everything Environments: A Taxonomy, Review, and Future Directions." *ACM Computing Surveys (CSUR)* (2022).
[25] Muhammad Rizwan Anawar, Shangguang Wang, Muhammad Azam Zia, Ahmer Khan Jadoon, Umair Akram, Salman Raza, "Fog Computing: An Overview of Big IoT Data Analytics", Wireless Communications and Mobile Computing, vol. 2018, Article ID 7157192, 22 pages, 2018. https://doi.org/10.1155/2018/7157192.
[26] Effective scheduling algorithm for load balancing in fog environment using CNN and MPSO.
[27] Zhou DX.Theory of deep convolutional neural networks: down sampling.Neural Netw. 2020;124:31927
[28] Souza, V.B., Masip-Bruin, X., Marín-Tordera, E., SànchezLópez, S., Garcia, J., Ren, G.J., Jukan, A., Juan Ferrer, A.:Towards a proper service placement in combined fog-to cloud (F2C) architectures. Futur. Gener. Comput. Syst. 87,1–15 (2018)
[29] Lin, C.-C., Yang, J.-W.: Cost-efficient deployment of fog computing Systems at Logistics Centers in industry 4.0.IEEE Trans. Ind. Inf. 14(10), 4603–4611 (2018).
[30] Gupta, H., et al.: iFogSim: a toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. Software: Practice and Experience (SPE). 47(9), 1275–1296 (2017).
[31] Mahmud, R. and R. Buyya, Modelling and simulation of fog and edge computing environments using iFogSim toolkit. Fog and Edge Computing: Principles and Paradigms. p. 1–35 (2019)
[32] Ghobaei-Arani, M., Khorsand, R., Ramezanpour, M.: An autonomous resource provisioning framework for massively multiplayer online games in cloud environment. J. Netw. Comput. Appl. 142(18), 76–97 (2019)
[33] Sun, Y., Lin, F., Xu, H.: Multi-objective optimization of resource scheduling in fog computing using an improved NSGA-II. Wirel. Pers. Commun. 102(2), 1369–1385 (2018) on. IEEE (2015)
[34] De Benedetti, M., et al.: JarvSis: a distributed scheduler for IoT applications. Clust. Comput. 20(2), 1775–1790 (2017)
[35] Zeng, D., Gu, L., Guo, S., Cheng, Z., Yu, S.: Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system.IEEE Trans. Comput.65(12), 3702–3712 (2016)

[36] Sun, Y., Zhang, N.: A resource-sharing model based on a repeated game in fog computing. Saudi journal of biological sciences (SJBS). 24(3), 687–694 (2017).

[37] Ghobaei-Arani, Mostafa, Alireza Souri, and Ali A. Rahmanian. "Resource management approaches in fog computing: a comprehensive review." *Journal of Grid Computing* 18.1 (2020): 1-42.

[38] Zinoviev, Dmitry. "Discrete Event Simulation: It's Easy with SimPy!." *arXiv preprint arXiv:2405.01562* (2024).