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

Improving Fog Gateway with Novel Metaheuristic-Driven AI Technique for Lessening the Delay and Energy Measures

Pooja Grover[1], Swati Singh[2], Beemkumar Nagappan[3], Amandeep Gill[4], Abhiraj Malhotra[5]

Submitted: 05/02/2024 **Revised**: 13/03/2024 **Accepted**: 19/03/2024

Abstract: The increasing need for quick data transmission and energy efficiency at the edge of the network has led to the development of a technology known as fog computing. Fog gateways are crucial elements in the architecture, as they offer computational capacity and enhance accessibility to end users and Internet of Things (IoT) devices for data-absorbing tasks. Improving latency and improving energy efficiency at the Fog Gateway remains a significant challenge. This research proposes a framework utilizing metaheuristic-driven artificial intelligence (AI) techniques to address the problem. This work introduces an innovative snow ablation search-driven catboost (SAS-CB) approach for identifying computational demands. The information from the IoT-driven fog computing system is utilized to develop the proposed SAS-CB method. We utilized sensors to gather environmental information for this research. Further feature selection is carried out utilizing the snow ablation optimization (SAO) technique to decrease the misinterpretation rate of the CB technique. The proposed method is implemented on a Python platform and evaluated based on several metrics such as utilization of energy (9W), latency (20s), and accuracy (90.45%). The experimental evidence indicates that the suggested solution outperformed existing methods in enhancing the fog gateway with favorable energy and latency parameters.

Keywords:*Artificial Intelligence (AI),Energy Use, Fog Gateway, Internet of Things (Iot), Latency, Novel Snow Ablation Search Driven Catboost (SAS-CB)*

1. Introduction

Fog gateways possess revolutionized processing by maximizing efficiency and productivity by allocating processing resources nearest to the boundary of the networking.A fog gateway can quickly and efficiently minimize latency by putting hardware proximity to data sources. There is a significant requirement to improve fog gateway architecture since there is an increasing require for real-time processing and data communication [1].Distributing quantitative investigation and conservation, fog gateways bring processing capacity closer to consumers and IoT devices through cloud computing [2]. Businesses gain from fog computing because it slows down data preparation, improves dependability for time-sensitive activities, and decreases bandwidth consume [3]. An efficient fog gateway, which connects edge devices to fog systems, is crucial to the gateway's performance [4]. Energy economy is crucial for edge devices with restricted bandwidth in fog gateway configurations [5]. To fog gates and connection devices to last longer and have less effect on the surroundings, effective energy conservation is essential. Therefore, ecological and cost-effective processing constructions which is vital to design fog gateway solutions that are energy-efficient [6, 7].

Fog gateways can have their processing powers increased the data intake accelerated upwards and latency lowered by making use of modern hardware characteristics such as powerful central processing units, memory components and networking compatibility [8]. Fog gateway solutions can be made even more efficient by making the use of specialized processors including “Field-Programmable Gate Arrays (FPGAs) and Graphics Processing Units (GPUs)” [9, 10].

In this work, we present a novel snow ablation search-driven catboost (SAS-CB) method for determining the computing requirements.

There is a list of related works in Section 2. In Section 3, the methodology is presented. The findings are indicated in Section 4. Section 5 conclusion is presented.

[1]*Assistant Professor, Department of ISME, ATLAS SkillTech University, Mumbai, Maharashtra, India, Email Id- pooja.grover@atlasuniversity.edu.in, Orcid Id- 0009-0000-8753-6614*
[2]*Assistant Professor, Maharishi School of Engineering & Technology, Maharishi University of Information Technology, Lucknow, India - 226036, Email Id- swatisingh5444@gmail.com, Orcid Id- 0009-0007-3888-3258*
[3]*Professor, Department of Mechanical Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Karnataka - 562112, India, Email Id- n.beemkumar@jainuniversity.ac.in, Orcid Id- 0000-0003-3868-0382*
[4]*Dean R&D, Department of Electrical Engineering, Vivekananda Global University, Jaipur , Email ID- amandeep.gill@vgu.ac.in, Orcid Id- 0000-0002-3725-8449*
[5]*Centre of Research Impact and Outcome, Chitkara University, Rajpura-140417, Punjab, India, Email ID- abhiraj.malhotra.orp@chitkara.edu.in, Orcid Id- https://orcid.org/0009-0005-1871-4807*

2. Related Works

Study [9] explored the role of abilities in facilitating “fog computing (FC)”.Their findings suggested a significant, but unexplored the capacity for intelligence in addressing the obstacles of FC. Research [10] introduced a new “intelligent multimedia data segregation (IMDS)” technique that utilized “machine learning (ML)” in the FC domain.They obtained a categorization accuracy of 92% in the model simulation, minimized latency by 95% contrasted to the previous system and enhanced thesuperior of services in e-healthcare.

Article [11] presented an analytics strategy for implementing a distributed network connection employing ML techniques in an FC environment.Additionally, their findings suggested that the training and forecasting durations for every system are appropriate use in environments where low latency is crucial. Author [12] investigated the advantages of fog computing through the introduction of an innovative generalized developing framework on the detector equipment by simulating the information streaming in the fog, rather than sending the entire raw sensor data to the cloud back-end.The investigation demonstrated that the suggested model reduce the amount of transmissions through the wireless link by 98%. Additionally, the fog node was effective in replicating the data streaming with 97% accuracy. The “secure computation offloading scheme in Fog-Cloud-IoT (SecOFFFCIoT)” environment presented in Paper [13]. The execution findings indicated that the interruption was minimal and insignificant energy consumption.

Study [14] emphasized the scheduling assignments for fog-based IoT systems to reduce long-term service latency and computational expenses while observing resource and timeline requirements. The evaluation findings demonstrated that the suggested algorithm surpassed some baseline techniques in service latency, computing cost, energy utilization and task completion. Additionally, it addressed the “Single Point of Failure (SPoF)” and load managing problems. Research [15] presented a comprehensive description of IoT, its expansion, FC and ML methods for enhancing security in IoT equipment and FC networks. The text explored ML methods for identifying irregularities and assaults, presented strategies for managing the increasing volume of data in IoT and examined security challenges related to FC.

3. Methodology

3.1 A simulation-based fog computing system

Developed a simulator (SR) for conducting FC or IoT-based investigations. The SR includes virtual elements including actuators, Fog equipment and sensors to simulate an FC environment. It enables resource administration strategies to manage, track and regulate all elements.

3.1.1 The Structure of Simulation-Based FC

The FC architecture filters and analyzes sensor statistics at the edge, reducing traffic on IoT networks. FC architectures disperse computation over the surroundings, enabling Flexibility Attributes. To create a simulated setting for FC, suggested an SR, "called iFogSim," in which the FC platform in IoT measures basic criteria including latency, energy consumption, cost, scalability and networking employ.

In iFogSim, FC characteristics are handled as components, with every layer pursuing specialized responsibilities and passing inputs to the higher component for evaluation. The major component of responsibility in this paradigm.

1. The IoT Sensors and Actuator function - The bottommost element that connects with actual life information collection and sends to the larger element utilizing sensors.

2. Fog equipment- Manage computing deployment requirements by processing sensor information, collecting outcomes and sending them to the clouds for preservation to the actuator for immediate responses.

3. IoT Data Flow- The element is responsible for transmitting the information sequence from the sensing to the deployment components or actuator. In ifogSim, the data flow is denoted as Tuple.

4. Observing component- The component is in charge of overseeing the accessibility of all iFogSim information, including fog equipment, actuators, sensors and power usage.

5. The information Administration- component is in charge of meeting the deployment's Quality of Service (QoS) needs by arranging the positioning and planning of the deployment components. A centralized information administration technique is employed, where every piece of equipment communicates resource statistics to a central information administrator.

6. Deployment approach- It also referred as the "Programming Approach." iFogSim implements an SR deployment utilizing the deployment framework. The deployment Framework is based on a "Distributed Data Flow (DDF) Model," in which transferring components are represented as components or edges and data flow occurs through instructed edges connecting these components or vertices.

iFogSim SR allows the creation of IoT networks using pre-defined categories including Physical Structure, Sensor, Actuator and Fog equipment within the software or using the “Graphical User Interface (GUI)” that offers a drop-down functionality for drawing a structure. iFogSim is a Java-depend SR that is a modified version of the CloudSim(CS) SR. iFogSim is built on the CS principle includes all CS modules and additional capabilities designed for the FC conditions.

3.2 Snow ablation search (SAS)

The concept of SAS is also known as SAO is derived from the melted and sublimated characteristics of snow. Subsequently, the mathematical structure of this technique is presented. The SAO was modeled after the sublimating and melted qualities of snow. Subsequent sections will discuss the SAO technique dual-population system, exploring level, exploiting level and initiation level.

3.2.1 Initiation level

The iterative procedure in SAO starts with an constructed swarm. The swarm is usually depicted as a matrix with M rows and Dim columns, where M is the dimension of the swarm and Dim is the number of dimensions in the solutions space, as indicated in Equations. KandVrepresent the bottom and upper boundaries of the solutions space, accordingly. In equation (1) θ represents a randomized produced amount in the range of 0 to 1.

$$Y = K + \theta \times (V - K) = \begin{bmatrix} Y_{1,1} & Y_{1,2} & \cdots & Y_{1,Dim-1} & Y_{1,Dim} \\ Y_{2,1} & Y_{2,2} & \cdots & Y_{2,Dim-1} & Y_{2,Dim} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_{M-1,1} & Y_{M-1,2} & \cdots & Y_{M-1,Dim-1} & Y_{M-1,Dim} \\ Y_{M,1} & Y_{M,2} & \cdots & Y_{M,Dim-1} & Y_{M,Dim} \end{bmatrix}_{M\times Dim} \quad (1)$$

3.2.2 Exploring level

The component offers an in-depth explanation of SAO's exploration methodology. Due to the unpredictable motion, the search mechanisms present a highly decentralized characteristic, when the snow or aqueous solution, transforms into vapor. The technique utilizes the Brownian principle represent the circumstance. Brownian principle is a stochastic technique commonly employed to model phenomena including animal feeding behavior and irregular particle movements. The step size of a typical Brownian principle is dictated by a probabilistic density function derived from a normal distribution with a mean of zero and a deviation of one. Here is the comparable mathematical description as shown in equation (2).

$$e_{BM}(w; 0,1) = \frac{1}{\sqrt{2\pi}} \times \exp\left(-\frac{w^2}{2}\right) \quad (2)$$

Below the equation used to calculate coordinates during the exploring method.

$$Y_j(s+1) = Elite(s) + BM_j(s) \otimes \left(\theta_1 \times \left(H(s) - Y_j(s)\right) + (1-\theta_1) \times \left(\bar{Y}(s) - Y_j(s)\right)\right) (3)$$

The character⊗ represents entry-wise multiples, θ_1 represents an amount systematically chosen from the range[[0,1], $Y_j(s)$denotes the j_{th}individual at the t_{th}iteration and $BM_j(s)$represents a vector containing random amounts following Gaussian transportation, symbolizing Brownian principle. $\bar{Y}(s)$is the centroid location(CL) of the whole swarm, $Elite\,(s)$ is an independently selected participant from a category of numerous $Elite$ in the swarm, and $\bar{Y}(s)$denotes the current optimum solution.

$$Y(s) = \frac{1}{M} \sum_{j=1}^{M} Y_j(s) \quad (4)$$

$$Elite(s) \in [H(s), Y_{second}(s), Y_{third}(s), Y_d(s)] \quad (5)$$

$Y_{third}(s)$and$Y_{second}(s)$represent the 3rd and 2nd-best people in the present group. $Y_d(s)$represents the CL of individuals with fitness levels in the top 50%. The executives in the investigation are individuals whose fitness levels ranked in the highest 50% for easiness. Additionally, $Y_d(s)$is calculated using the mathematical procedure in Equation.

$$Y_d(s) = \frac{1}{M_1} \sum_{j=1}^{M_1} Y_j(s) \quad (6)$$

$Y_j(s)$represents the j_{th}top executive and M_1 is the number of executives, which is half dimension of the total swarm. The $Elite\,(s)$ is independently selected from a collection that comprises the CL of executives as the present finest solution, the 2nd- finest individuals and the 3rd- finest individuals for every variation.

3.2.3 Exploiting level

The segment discusses the exploitation characteristics of SAO. When the snow melts and turns into an aqueous solution, search mechanisms should utilize exceptional solutions around the existing optimum solution instead of expanding with a disorganized aspect in the solutions region. The degree-day approach is a utilized model for snowmelt that illustrates the melted processes. This method is expressed as follows.

$$N = DDF \times (S - S_1) \quad (7)$$

Nrepresents the rate which snow melted a crucial parameter for simulating melted patterns through the exploiting stage. $S$is the mean temperature recorded each day. The baseline temperature, denoted as S_1 is typically established at 0.

$$N = DDF \times S \quad (8)$$

The DDF, ranging from 0.35 to 0.6, signifies the degree-day component. Below the equation that modifies theDDF variable in each iteration.

$$DDF = 0.35 + 0.25 \times \frac{e^{\frac{s}{s_{max}}} - 1}{e - 1} \quad (9)$$

The terminating circumstance is indicated byt_{max}. The melted ratio in SAO is calculated using the following equation.

$$N = \left(0.35 + 0.25 \times \frac{e^{\frac{s}{s_{max}}-1}}{e-1}\right) \times S(t), S(t) = e^{\frac{s}{s_{max}}} \quad (10)$$

The location updated formula appears after the SAO exploiting step.

$$Y_j(s+1) = N \times H(s) + BM_j(s) \otimes \left(\theta_2 \times \left(H(s) - Y_j(s)\right) + (1-\theta_2) \times \left(Y(s) - Y_j(s)\right)\right) (11)$$

Where $\theta_2$is an independently determined integer from the range of -1 to 1 and $N$is the rate at which snow melts. The trait facilitates interpersonal communication. Participants in the stage are inclined to explore prospective areas due to the parameters $-\theta_2 \times \left(H(s) - Y_j(s)\right)$and $(1-\theta_2) \times \left(\bar{Y}(s) - Y_j(s)\right)$are influenced by the CL of the swarm and the present optimum searched period.

3.3 CatBoost (CB)

The CB classification is an additional method in ML technique known for its efficiency in forecasting categorized features. CB is a gradient-boosting method that utilizes binary decision trees as basis predictors. We are given a dataset $C = \{(W_i, z_i)\}_{i=1,\ldots\ldots,n} where W_i = (w_i^1, w_i^2, \ldots \ldots, w_i^m)$is vectors of m characteristics and $z_i \in \mathbb{R}$ is the tackle characteristic that can be binary or represented as a numerical value (0 or 1). The samples (W_i, z_i) are independently and similarly dispersed to uncertain distributions $o(.,.)$. The purpose of the learning assignment is to train the functional $G: \mathbb{R}^m \to \mathbb{R}$that reduces the predicted loss as specified in equation (12).

$$\mathcal{L}(G) \coloneqq \mathbb{E}\left(z, G(W)\right) \quad (12)$$

Where $K(.,.)$is a differentiable loss functional and (W, z) is a test data point independently selected from the training dataset C.

The gradient boosting approach produces a succession of estimates $G^s: \mathbb{R}^m \to \mathbb{R}, s + 0,1 \ldots$ in a greedy manner, continuously. $G^s$is derived from the preceding assumption G^{s-1}, G^s using an additive procedure, where $G^S = G^{s-1}\alpha H^s$. Here, α is the step size and $G^s\mathbb{R}^m \to \mathbb{R}$is a base predictor determined from a set of variables H to minimize the expected loss defined in equation (13).

$$h^s = argmin_{h\in H}\mathcal{L}(G^{s-1} + h) = argmin_{h\in H}\mathbb{E}K\left(z, G^{s-1}(W) + h(W)\right) \quad (13)$$

The minimizing challenge is solved using the Newton technique, which involves a 2nd-order assumption of $\mathcal{L}\left(G^{s-1} + h^1\right)$at G^{s-1} or by performing a downward gradient stage. Each of these algorithms involves gradient descent.

3.4 Snow Ablation Search-Driven CatBoost (SAS-CB)

A hybrid solution called SAS-CB is suggested to improve fog gateway technologies by targeting the reduction of delay and optimization of energy consumption. SAS-CB combines SAS and CB. SAS emulates the actual snow ablation procedure as a search algorithm to enhance fog gateway efficiency. The system adapts fog node settings in response to environmental factors, workload requirements, and energy limitations. SAS-CB uses statistical analysis to anticipate workload changes and proactively optimizes fog node arrangements in real time to improve efficiency and energy savings. This adaptive method guarantees that fog gateways function in different circumstances, minimizing delays and preserving energy resources. Algorithm 1 shows the Snow Ablation Search-Driven CatBoost (SAS-CB).

Algorithm1: SAS-CB

$importnumpyasnp$

$importpandasaspd$

$fromcatboostimportCatBoostRegressor$

$fromsklearn.model\_selectionimporttrain\_test\_split$

$fromsklearn.metricsimportmean\_squared\_error$

$defsas\_cb(X\_train, y\_train, X\_test)$:

$X\_train, X\_val, y\_train, y\_val = train\_test\_split(X\_train, y\_train, test\_size = 0.2, random\_state = 42)$

$model = CatBoostRegressor(iterations = 1000, depth = 6, learning\_rate = 0.1, loss\_function = 'RMSE', logging\_level = 'Silent')$

$model.fit(X\_train, y\_train, eval\_set = (X\_val, y\_val), early\_stopping\_rounds = 50, verbose = False)$

$y\_pred = model.predict(X\_test)$

$returny\_pred$

$if \_name\_ == "\_main\_"$:

$data = pd.read\_csv('data.csv')$

$X = data.drop(columns = ['target\_column'])$

$y = data['target\_column']$

$predictions = sas\_cb(X\_train, y\_train, X\_test)$

$mse = mean\_squared\_error(y\_test, predictions)$

$print(f'MeanSquaredError: \{mse\}')$

4. Result

The collection includes 11,000 samples of temperatures, moisture, gases, and image statistics. The collection is divided into 2 sections for training and testing. Seventy percent (7700) of the sample is used for training, whereas thirty percent (3300) is used for testing. Two methods are used for pre-processing acquired statistics, computing the median of the entire day's data for values that are inappropriate and computing the mean of data in intervals and adding to the subsequent intervals. [16].

We implemented our approach in Python (v 1.8) and the system configuration includes Pytorch 1.14, compatible with Python 1.6, on a Windows 11 OS. The proposed method is evaluated in terms of accuracy, latency and utilization of energy compared with the existing approaches, which are "multiscale convolutional long short-term memory model (MCLSTM) [17], binary convolutional neural network with numerous skip connections (BNSC-Net) [18], Bidirectional Long Short-Term Memory (BiLSTM) [19].

Accuracy matrices to enhancements in reducing delay and

saving energy in fog computing settings are evaluated by measuring metrics such as transmission latency and consumption of energy. A comparison of accuracy is presented in Fig 1. Our suggested SAS-CB approach performed (90.45%), in contrast to (85%), (86.10%), and (87.90%) of the current techniques such as MCLSTM, BNSC-Net and BiLSTM. The outcomes demonstrate that the suggested strategy outperforms existing methods.

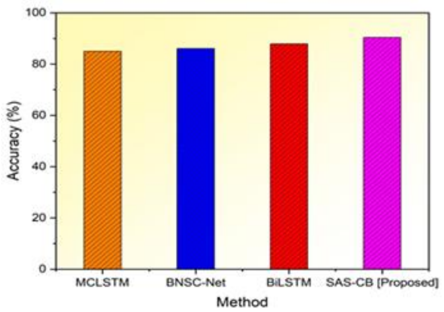

Fig .1. Result of accuracy

Latency managing fog gateways reduces delays and energy consumption, improving platform effectiveness and productivity in interpreting data at the edge of the network. A latency comparison is shown in Fig 2. While the existing methods, MCLSTM, BNSC-Net and BiLSTM, achieved (100s), (85s) and (50s) respectively, our proposed SAS-CB methodology achieved (20s). The findings demonstrate that our suggested approach is lower than the existing methods.

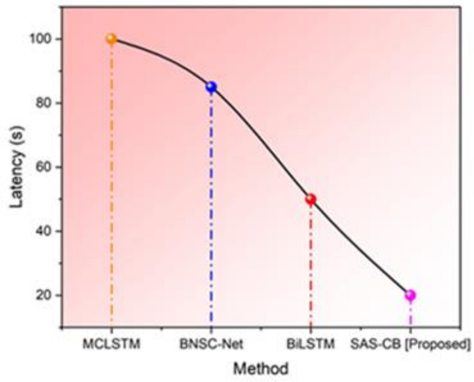

Fig .2. Output of latency

Energy matrices through proper utilization can allocate resources, prioritize tasks and implement energy-saving measures to enhance performance while minimizing energy consumption. An energy comparison is shown in Fig 3. While the existing methods, MCLSTM, BNSC-Net, and BiLSTM achieved (10W), (15W) and (17W) respectively, our proposed SAS-CB methodology achieved (9W). The findings demonstrate that our suggested approach lower than the existing methods for improving fog gateway for lessening the delay and energy measures. Table 1 shows the overall result comparison.

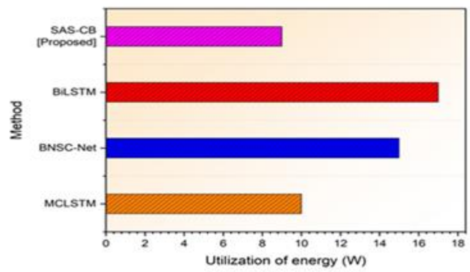

Fig.3. Output of utilization of energy

Table 1.overall result comparison

Methods	Accuracy%	Latency(s)	Utilization of energy(W)
MCLSTM	85%	100s	10W
BNSC-Net	86.10%	85s	15W
BiLSTM	87.90%	50s	17W
SAS-CB [Proposed]	90.45%	20s	9W

5. Conclusion

The increasing need for fast and energy-saving computing at the edge of the network has led to the emergence of fog computing as a possible approach. Fog gateways were essential for delivering computing resources and enabling data analysis for end users and IoT devices. The research presents a new framework utilizing metaheuristic-driven artificial intelligence tools to tackle the issue. The SAS-CB architecture utilizes IoT-driven fog computing statistics to assess computing needs. The framework improves feature selection by utilizing sensors to gather environmental data. The SAS-CB technique greatly improves fog gateways in terms of utilization of energy (9W), latency (20s), and accuracy (90.45%). The proposed method, executed on the Python platform and assessed using diverse criteria, exhibits better performance in comparison to current techniques. The experimental results confirm that it improves fog gateway operations, highlighting its ability to tackle the changing issues of edge computing in current network conditions. The constraints might be sensitivity to parameter decisions, reliance on the integrity and availability of accessible data and difficulties in managing complicated or noisy collections. The future potential of this technology includes improved optimization through enhanced techniques, combination with a big data analysis, use in other domains including climate simulation and its adaption for real-time decision-making in snow-related sectors and investigations.

References

[1] O'donovan, P., Gallagher, C., Bruton, K. and O'Sullivan, D.T., (2018). A fog computing industrial cyber-physical system for embedded low-latency machine learning Industry 4.0 applications. Manufacturing letters, 15, pp.139-142.

[2] Angel, N.A., Ravindran, D., Vincent, P.D.R., Srinivasan, K. and Hu, Y.C., (2021). Recent advances in evolving computing paradigms: Cloud, edge, and fog technologies. Sensors, 22(1), p.196.

[3] Das, S. and Guria, P., (2022, January). Adaptation of machine learning in fog computing: An analytical approach. In 2022 International Conference for Advancement in Technology (ICONAT) (pp. 1-11). IEEE.

[4] FarajiMehmandar, M., Jabbehdari, S. and Haj SeyyedJavadi, H., (2020). A dynamic fog service provisioning approach for IoT applications. International Journal of Communication Systems, 33(14), p.e4541.

[5] Alzubi, O.A., Alzubi, J.A., Alazab, M., Alrabea, A., Awajan, A. and Qiqieh, I., (2022). Optimized machine learning-based intrusion detection system for fog and edge computing environment. Electronics, 11(19), p.3007.

[6] Etemadi, M., Ghobaei-Arani, M. and Shahidinejad, A., (2021). A cost-efficient auto-scaling mechanism for IoT applications in fog computing environment: a deep learning-based approach. Cluster Computing, 24(4), pp.3277-3292.

[7] Ullah, A., Yasin, S. and Alam, T., (2023). Latency aware smart health care system using edge and fog computing. Multimedia Tools and Applications, pp.1-27.

[8] O'Donovan, P., Gallagher, C., Leahy, K. and O'Sullivan, D.T., (2019). A comparison of fog and cloud computing cyber-physical interfaces for Industry 4.0 real-time embedded machine learning engineering applications. Computers in industry, 110, pp.12-35.

[9] La, Q.D., Ngo, M.V., Dinh, T.Q., Quek, T.Q. and Shin, H., (2019). Enabling intelligence in fog computing to achieve energy and latency reduction. Digital Communications and Networks, 5(1), pp.3-9.

[10] Kishor, A., Chakraborty, C. and Jeberson, W., 2021. A novel fog computing approach for minimization of latency in healthcare using machine learning.

[11] Patman, J., Alfarhood, M., Islam, S., Lemus, M., Calyam, P. and Palaniappan, K., (2018, April). Predictive analytics for fog computing using machine learning and GENI. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 790-795). IEEE.

[12] Lavassani, M., Forsström, S., Jennehag, U. and Zhang, T., (2018). Combining fog computing with sensor mote machine learning for industrial IoT. Sensors, 18(5), p.1532.

[13] Alli, A.A. and Alam, M.M., (2019). SecOFF-FCIoT: Machine learning based secure offloading in Fog-Cloud of things for smart city applications. Internet of Things, 7, p.100070.

[14] Gazori, P., Rahbari, D. and Nickray, M., (2020). Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach. Future Generation Computer Systems, 110, pp.1098-1115.

[15] Moh, M. and Raju, R., (2018, July). Machine learning techniques for security of Internet of Things (IoT) and fog computing systems. In 2018 International Conference on High Performance Computing & Simulation (HPCS) (pp. 709-715). IEEE.

[16] Suryadevara, N.K., (2021). Energy and latency reductions at the fog gateway using a machine learning classifier. Sustainable Computing: Informatics and Systems, 31, p.100582.

[17] Qiao, H., Wang, T. and Wang, P., (2020). A tool wear monitoring and prediction system based on multiscale deep learning models and fog computing. The International Journal of Advanced

Manufacturing Technology, 108, pp.2367-2384.

[18] Wu, L., Lin, X., Chen, Z., Huang, J., Liu, H. and Yang, Y., (2021). An efficient binary convolutional neural network with numerous skip connections for fog computing. IEEE Internet of Things Journal, 8(14), pp.11357-11367.

[19] Bharathi, P.D., Narayanan, V.A. and Sivakumar, P.B., (2022). Fog computing enabled air quality monitoring and prediction leveraging deep learning in IoT. Journal of Intelligent & Fuzzy Systems, 43(5), pp.5621-5642. 10.3233/JIFS-212713 v