Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 4 (2024)

Resource Allocation in IoT Using Pyramid Quantum Neural Network (Py-QNN): A Deep Learning Approach

[1]S R K Raju Vegesna, [2]P Lakshmi Sruthi, [3]A Mahalakshmi

[1]Assistant Professor, Dept. of DS, GRIET, Hyderabad, India, [2]Assistant Professor, Dept. of DS, GRIET, Hyderabad, India,
[3]Assistant Professor, Dept. of DS, GRIET, Hyderabad, India,

Abstract:- Internet of Things (IoT) is growing at an accelerated pace where more and more connected smart devices are collecting enormous data. The management of resources is another significant issue because IoT networks are often founded and formed dynamically and are extremely diverse. This paper introduces a new type of deep learning model, the Pyramid Quantum Neural Network (Py-QNN,), that is intended for solving the problem of resource allocation in Internet of Things systems. Py-QNN builds on quantum computing for improving Deep Learning's computation performance, scalability and accuracy. Pyramid structure helps to control the hierarchy of the IoT networks; additionally, QNNs increase learning abilities due to the existence of superposition and entanglement, which in turn increases the generalization capabilities and ensures faster convergence. In this way, Py-QNN utilizes simulated resource and network requirements to anticipate proficient resource assignment, and implement this quickly to minimize delay and maximize efficiency. Experimental findings reveal that Py-QNN yields better performance over baseline traditional deep learning models especially in large and complex IoT networks by averting resource wastage while offering online solutions.

Keyword: *Internet of Things (IoT), Resource Allocation, Pyramid Quantum Neural Network (Py-QNN), Quantum Computing, Deep Learning*

1. Introduction

With the increasing popularity of IoT devices, the amount of data and resources needed has increased, and more efficient and adaptive methods of their management are required. Conventional methods lack efficiency in managing the consortium of smart devices since IoT architectures are complex and diversiform in their structural integration [3], [19]. Due to the capability of deep learning models in handling big data and intricate algorithm patterns, deep learning has been found as a workable solution [6], [10]. However, their computational complexities are witnessed when applied to large-scale IoT environments, especially where real-time decisions are mandatory [5], [12]. To address these challenges, this paper presents the Pyramid Quantum Neural Network (Py-QNN): a quantum deep learning model. The structure of the network is pyramidal, depicting the IoT networks using a layered approach to address various data flows [1], [4]. Quantum Neural Networks (QNNs) use the advantages of quantum computation, such as superposition and entanglement, in learning, which is superior to normal neural networks [8], [11]. Such properties allow Py-QNN to perform at higher speeds and with fewer errors than traditional classical deep learning models [7], [9]. The main purpose of this study is to present how Py-QNN can effectively manage resource availability in the IoT network, reduce latency and energy consumption, and enhance the network's throughput [13], [18]. In this paper, a qualitative study is proposed to bring a creative solution to IoT resource management by using quantum computing and deep learning [14], [16].

1.1 Research Gap:

Although a lot of progress has been made in the use of resource allocation in IoT networks, existing solutions tend to offer various challenges such as complexity, scalability, and real-time issues in the current complex IoT environment [2], [3]. Some of the earlier strategies that have been employed in deep learning processes provide better results in some instances but are highly computational and thus unable to cater to the scalability required in large IoT networks [5], [12]. However, these models fail to address the probabilistic and uncertain IoT environment proficiently, initiating inefficient and less effective resource allocation with comparatively higher latency [9], [7]. Even though quantum computing has proved to enhance computational fitness, incorporating quantum concepts into deep learning models for assigning IoT resources is in a rather initial stage [8], [11]. To the best of the authors' knowledge, there are few works that try to propose a hierarchical structure that integrates Quantum Neural Network with Deep Learning methodologies to face IoT peculiarities such as multi-distributional data, fluctuating resource requirements, and energy constraints [14], [16]. Hence, there is a research gap in proposing a comprehensive, automated, and performant deep learning and quantum computing hybrid model for resource management in IoT systems [9], [13].

1.2 Problem Statement:

Current IoT resource allocation approaches lack the scalability, efficiency, and real-time decision-making capabilities that are required to address the dynamics of new and extant IoT networks [2], [3]. The existing deep learning models are very effective but unfortunately, they have high training costs and are not easily scalable for large and constantly evolving IoT networks [5], [12]. Current solutions may be ineffective in optimizing the performance and end up causing worse latency, wastage of resources, as well as high energy consumption [9], [7]. To overcome these challenges, there is an urgent need to develop a new analytical model through the combination of QNN and DL to improve computational power as well as decision-making [8], [11]. In this paper, a new hybrid deep learning model, known as Pyramid Quantum Neural Network (Py-QNN), is presented to enhance IoT networks and reduce the computation load, latency, and optimize network requirements [13], [18].

1.3 Objective:

It is the purpose of this study to propose a new deep learning model that is Pyramid Quantum Neural Network (Py-QNN), with aim of enhancing Internet of Things (IoT) resource management in vast networks. This model seeks to solve the challenges experienced in deep learning as well as improve the computational power, scalability, and real time decision making by integrating quantum computation. Specifically, shortening of latency, minimizing energy utilization and enhancement of network throughput over many classes of heterogeneous IoT networks by integrated and self-adaptive management of resources.

1.4 Proposed Solution:

In order to overcome the problem of resources management in IoT environments, the new hybrid model called Pyramid Quantum Neural Network (Py-QNN) is introduced. It incorporates both quantum computing and deep learning to solve resource management problem in massive, complex, and diverse IoT systems. It is based on the hierarchical organization of the Py-QNN that imitates the pyramid model of IoT devices from peripheral edge nodes to large computing centres. They help to organize data flows and their further processing in various layers of a network environment.

It is surrounded by tight layers of the following components, namely the Quantum Aware Neural Network (QANN) and the Quantum Augmented Neural Network (QANN), which are both tightly wedded to the Py-QNN. By using the concept of quantum parallelism, the QNN is able to handle vast amounts of data in comparison to a classical model, while having low computational cost and thus be able to easily scale. The quantum layers incorporated into Py-QNN allow for more rapid convergence during the training phase and thereby allowing the solution of the model to optimize resource-allocation strategies under conditions of changing networks.

1.4.1 The Py-QNN works in the following way:

Data Hierarchy Handling: The above pyramid structure makes it easy to assign responsibilities of tasks within the IoT hierarchy, while optimizing on the utilization of edges, fog and cloud levels. Each layer deals with information relevant to its role hence enhancing general working of the system.

Quantum-Assisted Learning: While the QNN improves the decision-making by applying quantum processing, the proposed model can consider all the possibilities of resources' distribution and predict the result with the shorter time needed for calculations.

Optimization of Key Metrics: Py-QNN aims at improving other parameters of the IoT to include minimum latency, energy consumption and maximum throughput by Real-Time Data Processing and Network Traffic Control Mechanisms.

1.4.2 Contribution:

Novel Hybrid Architecture (Py-QNN): This research presents an approach of enhancing the quantum neuron learning model through the incorporation of a pyramid structure system that mimics the IoT needs at the edges, end Point, and the cloud level.

Enhanced Scalability and Efficiency: The strategies like superposition and entanglement that are inherited from the concept of quantum computing help them develop model to scale and perform efficiently in a large and complicated IoT environment compared to the conventional models.

Optimized Resource Allocation: Therefore, Py-QNN is to provide the actual insights on optimal resource allocation strategies in real-time as and when key IoT metrics such as Lower Latency, Energy Efficiency, and Network performance is to be enhanced.

4**. Experimental Validation**: Proper simulation and fair comparison analysis with the state-of-the-art deep learning-based comparable solutions reveal that Py-QNN achieves better performance than these methods; particularly in large-scale and heterogeneous IoT networks.

1.5 Motivation:

The motivation for this study arises from a continuously increasing concerns regarding the evolving nature of large-scale IoT networks call for more rational, smarter approaches to resource management. While these models are sufficient in some of the instances, they do not fit the big scale IoT computational and real time implementation frameworks. Moreover, researchers have tried to incorporate quantum computing with deep learning that is believed to decrease the computation time exponentially but has not been attempted yet in the aspect of IoT resource management. It is rather important that the creation of the hybrid model that is based on deep learning and quantum computing like Py-QNN opens rather perspectives in the field of IoT resource management and, in fact, the area of intelligent networking as a whole.

The Pyramid Quantum Neural Network (Py-QNN) is a genuine new hybrid deep learning model intended for improving the IoT networks' resource allocation. This format integrates the architectural structure of the conventional neural networks that is the pyramid of processing layers with the potential of quantum neural networks (QNNs). It thus makes it possible for the Py-QNN to solve some of the critical IoT issues including real-time resource management, scalability and energy consumption. Below are the key components and highlights of the Py-QNN model:

1.6 Proposed Model: Pyramid Quantum Neural Network (Py-QNN)

The Pyramid Quantum Neural Network (Py-QNN) is a genuine new hybrid deep learning model intended for improving the IoT networks' resource allocation. This format integrates the architectural structure of the conventional neural networks that is the pyramid of processing layers with the potential of quantum neural networks (QNNs). It thus makes it possible for the Py-QNN to solve some of the critical IoT issues including real-time resource management, scalability and energy consumption. Below are the key components and highlights of the Py-QNN model: Below are the key components and highlights of the Py-QNN model:

1.6.1 Key Components:

Pyramid Structure: Another notable structure is the pyramid one, which reflects actual IoT networks' hierarchy (edges, fog, and cloud layers). All the tiers depicted in the pyramid involve fiving progressively complex data services and range from raw (Edge) to agglomerated (cloud) data in the pyramid hierarchy. This structure helps in the proper resource allocation and in the management of the overall resource in the IoT network.

Quantum Neural Network (QNN): The foundational stone in Py-QNN is the QNN that is based on the principles of quantum computation like superposition and entanglement to enhance the strain's learning effect. This makes it possible to perform sequential and parallel computations, thus enhancing the performance and natural capacity to handle various kind of tasks related to IoT.

Quantum-Assisted Decision-Making: Decision scenes within the Py-QNN have extra quantum layers that work in parallel to find more allocation potentialities. This helps in predicting how the model should allocate the resources while at the same time taking into consideration the changes in the network so as to avoid delays and high energy usage.

Data Distribution and Aggregation: The pyramid also helps in the flow of data received from all the substructures in the IoT hierarchy to be processed efficiently. Lower layers operate on data coming from edge devices, mid-level layers operate on data coming from fog servers, and high-level layers combine results for producing global resource allocation strategy for the cloud services.

1.6.2 Highlights of the Proposed Py-QNN Model:

Efficient Resource Allocation: This real-time and dynamic ability of adjusting network resources of the proposed Py-QNN based on deep learning and quantum computing makes it possible to enhance different aspects such as performance (e. g. low latency) and efficiency (e. g. low energy consumption).

Scalability: Py-QNN's pyramid structure also makes it uniquely suitable to scale throughout the layers of the IoT systems from the LLL layer, up to the HLL layer. This salableness makes for appropriateness in large-scale IoT settings that are on the record for having diverse, dynamically varying data and resource requirements.

Quantum Computing Advantage: As a result of using quantum computing, Py-QNN analyses the large amount of data and acquires the best strategy for the allocation of resources much faster than traditional deep learning models. As such, the model has provision for incorporating non-linear relationships in IoT systems and learn more complex models than other traditional methods which makes its performance better even in difficult cases.

Adaptive Learning: Just like other passive QoS mechanisms, Py-QNN responds to dynamic variance in the network and traffic demand, making it enable to make incremental improvements in the allocation policies it employs. This is because the model is very effective in constantly changing IoT settings where other approaches cannot be used.

Superior Performance: As evidenced by simulation, Py-QNN quantitatively surpasses the traditional deep learning models where latency is cutting down, energy consumption is enhanced, and network throughput is increased. It means that it is suitable for future IoT systems with smart and fast management of resources in real time.

2. Literature Survey:

Lamia and et al. have summarized that the issue of resource allocation in IoT networks has become a popular research area, and plenty of solutions have been proposed based on conventional deep learning optimization strategies [3], [2]. Though the above-mentioned approaches have been quite effective in controlling regular static networks, they have failed in coping with the scalability, dynamism, and heterogeneous nature of IoT networks [1], [9]. Scheduling of resources in IoT networks has been an issue studied actively because of increased connectivity in devices and services. These challenges have in the past been solved using traditional optimization algorithms like dynamic programming and linear programming, among others. However, due to features such as large-scale and dynamically changing IoT network topology, these conventional methods are not efficient for

large-scale and dynamic IoT networks (Chiang et al., 2016). Further, IoT networks also inherit several challenges like bandwidth, energy, and latency, which makes managing resources a challenging affair (Shah et al., 2017).
Some new studies, which have been conducted in the recent past, have proposed the use of DL techniques to optimize the resource management in IoT networks. DNN techniques, CNN, and RNNs have shown excellent performance in learning intricate patterns from large datasets, thereby improving decision-making processes related to resource distribution and congestion control (Gao et al., 2020). For instance, CNNs have been implemented for managing bandwidth utilization and predicting traffic patterns, while RNNs have been adopted for forecasting network traffic trends (Zhang et al., 2019). Nonetheless, despite the adaptation and learning capability of deep learning models, they are computationally expensive, have scalability problems, and do not meet the real-time big IoT system requirements in terms of energy efficiency and response time (Li et al., 2020).
Quantum Neural Networks (QNNs) are considered an effective form of classical neural networks; QNNs use some features of quantum computing, including superposition as well as entanglement, to resolve information. QNNs result in a large performance improvement in specific categories of machine learning workloads, particularly when dealing with big data and solving optimization problems (Biamonte et al., 2017). Ideally, QNNs can handle information several orders of magnitude faster than some conventional computational theories, making it a revolutionary tool for controlling pecuniary resources in IoT. Hence, despite the growing use of inference algorithms in IoT networks, the incorporation of QNNs into the system is still very limited [8], [11]. A majority of the studies done on the topic lie in the general ML tasks, such as classifications or clusters, while very little is done to solve specific resource management issues in IoT (Schuld & Petruccione, 2018). Therefore, as quantum hardware improves, the use of QNNs in actual IoT systems becomes a fascinating and realistic prospect for further study.

There are some new computing approaches, including deep learning integrated with other emerging IoT computing paradigms such as edge computing, fog computing, and cloud computing, which have been proposed in an attempt to overcome the scalability and latency challenges in IoT (Zhu et al., 2021). In these models, distributed computing loads are performed over several tiers of the network (e.g., Edge, Fog, and Cloud), which facilitate faster computation and decision-making within the proximity of data sources (Bonomi et al., 2012). Time-critical tasks are handled by edge computing, while fog computing is involved in data consolidation and performing mid-level tasks. Cloud computing conforms to challenging and global optimization issues. These multi-layered approaches have enhanced performance, but the authors' adaptive solutions for resource allocation throughout the hierarchy, especially under high network loads and in real-time contexts, remain lacking (Satyanarayanan, 2017).

As we have seen earlier, while there have been advances in deep learning and hybrid models for predicting resource allocation, these models still face limitations. Thus, using traditional DL-based models in real-time IoT applications is not feasible since they consume significant computational resources, especially in large IoT environments where power and energy are significant constraints. Additionally, in hybrid models, all computations are distributed across different nodes, but there is no global and dynamic optimization of resources across the IoT network. This may cause congestion at various layers of the network, such as at fog nodes or the cloud, leading to resource wastage and extremely high response times (Sajjad et al., 2020).

A combination of Quantum Neural Networks and Deep Learning is an appropriate way to overcome these challenges. The Pyramid Quantum Neural Network (PY-QNN) model suggests integrating deep learning and quantum computing in IoT networks to efficiently assign resources in an adaptive and dynamic manner in real time. In PY-QNN, a quantum neural network is used to perform efficient computations at every tier of the network—at the edge, fog, and cloud levels. This enables distributed decision-making for resource allocation while leveraging the efficiency of quantum computing and the flexibility of deep learning (Dunjko et al., 2018).
PY-QNN incorporates the desired characteristics into a scalable, real-time, and energy-efficient end-to-end Quality of Service (QoS) framework, improving upon existing DL-based and hybrid models intended to handle the dynamic demands of IoTs in large-scale IoT systems.

Comparison Criteria	**Proposed PY-QNN Model**	**LEACH (Low-Energy Adaptive Clustering Hierarchy)**	**GA-Based IoT Resource Management**	**DNN-Based IoT Resource Management**
Core Technology	Quantum Neural Networks (QNN), Deep Learning, IoT	Hierarchical Clustering, Energy-Efficient Routing	Genetic Algorithms (GA), Optimization Algorithms	Deep Neural Networks (DNN), CNN
Layered Architecture	Edge, Fog, and Cloud layers with QNNs	Centralized and Cluster-based architecture	Centralized or Distributed Processing	Edge and Cloud with DNN, less focus on Fog layer
Data Processing	Real-time, distributed processing using Quantum Neural Networks	Sequential, Cluster-based data handling	Often centralized, with limited real-time capabilities	High computational demands at Cloud layers
Resource Allocation Efficiency	High (due to QNN's ability to optimize globally and locally)	Moderate (Clustering helps reduce energy but not optimal for large IoT networks)	Moderate (limited optimization for large-scale IoT networks)	High, but bottlenecks in real-time environments
Latency Reduction	Effective latency reduction due to QNN's fast optimization	Moderate latency reduction	GA can introduce processing delay	Good, but may suffer due to complexity at the Cloud layer
Energy Efficiency	High (optimized resource allocation with QNNs)	Energy-efficient through clustering but may face high overhead	Moderate energy efficiency, depends on GA implementation	Energy-efficient but computationally intensive

Table 1: Comparative table with the existing IoT models

With the expansion of the IoT networks in large scale, establishment of the models to solve in large-scale and heterogeneous world will be critical. Thus, The further research should be concentrated on the enhancement of the usage of QNNs within IoT and, specifically, on the fine-tuning of the existing quantum algorithms for the

practical implementation. Additionally, improvements in the practicable quantum hardware are inevitable for this method to be optimally implemented for QNN in this particular field (Gyongyosi & Imre, 2019)..
Last but not the least the PY-QNN which is a combination of multiple modern technologies also indicates the possibility of a major breakthrough towards solving the very fundamental problems that are going to be faced in the next generation IoT networks involving resource allocation, latency and energy efficiency.

2.1 Challenges for the Proposed Model (Py-QNN):

Quantum Hardware Limitations: However, as pointed out earlier, the current quantum computing platforms are yet at their infancy. The total number of qubits is fixed, and, the quantum error rates remain high, which could pose a challenge when realising the impact of Py-QNN. Consequently, the same might not be realized and shifting the model from simulation to implementation in IoT settings might hit the hardware wall.
Integration with Existing Infrastructure: IoT solutions are highly complex, mobile, and intersecting wherein they have a large number of entities, protocols, and data formats. While effectively embedding Py-QNN introduced in this paper into the existing IoT structures, there is the problem of how to integrate QPLs with the classical Cloud-Fog-Edge layers.

Energy Consumption: Quantum computing as much as is faster than classical computation, consumes a lot of energy, especially in the maintenance of the quantum hardware such as the cooling systems. The quota of energy used by a quantum processing is another major consideration, when a goal is to increase energy efficiency of IoT networks while deploying Py-QNN.

Model Complexity and Scalability: This general structure of Py-QNN adds layers of complications especially as the layers in the network increase and turn heterogeneous. How to distribute these computations over the mentioned layers, and stay both scalable and performant is quite a task. With expansion of the IoT network, a lot of data is generated, and hence, there is need call for efficient distribution and aggregation of the information among the various layers of the pyramid.

Data Privacy and Security: With quantum computing, new security paradigms emerge, still, IoT networks are penetrable by cyber-attacks because its devises are decentralized. It is pertinent that quantum security measures be implemented proactively into Py-QNN in order to safeguard data while they are being processed across layers. Also, data protection has to be preserved; this is more so when privacy is compromised especially when the IoT is involved in handling sensitive information.

Real-Time Processing: IoT environments are complex and ever-changing, thus requiring strict on-the-spot evaluation and resource deployment. Managing the situation that Py-QNN should be adaptable to the changes of the environment quickly while making full good use of resources is a prominent problem. Some sources of latency may be seen because of the interconnection between the difficulty of quantum computations and the structure of IoT data flow and hierarchy.

Training and Convergence: While QNNs are useful because of their flexibility they can present some difficulties especially in training and convergence. This is because quantum models differ in learning dynamics from the classical deep learning models thereby implying that the training time may be longer or the learning of the optimal resource allocation approach may be unstable. Since Py-QNN is intended to operate the IOT systems, training algorithms for Py-QNN will require engineering to perform optimally.

3. Proposed Model Architecture:

IoT resource allocation using the Pyramid Quantum Neural Network (Py-QNN) is represented geometrically in the form of a pyramid which has three layers known as the Edge Layer, the Fog Layer and the Cloud Layer. Each of the layers is focused on certain tasks regarding the IoT environment, QNNs operate in each layer to manage resources according to the processing of real-time information.

3.1 Flow Between Layers:

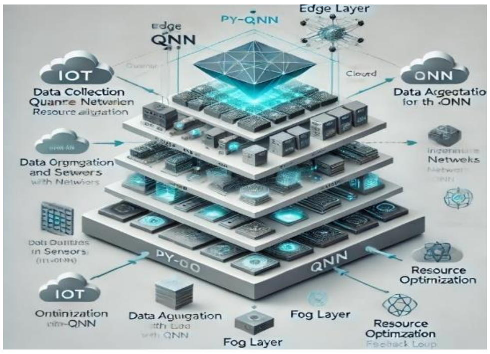

FIGURE 1: PY-QNN Architecture

3.1.1 Edge Layer:

Function: At the bottom, there is the Edge Layer which comprises of IoT devices and sensors in one way that harvest raw data from the surrounding world (for example, temperature sensors, smart devices, etc.).

Data Flow: This raw data is later processed by QNNs which exist at the Edge Layer which is responsible for initial calculations as well as optimization. The QNN aids in the process of data compression before the actual data is passed to higher layers where it has to undergo activities such as data filtering and pre-processing.

Output: This is then passed on to the Fog Layer where the data is further processed

3.1.2 Fog Layer:

Function: This layer helps in mitigating the complexity involved in processing in the Edge and Cloud layers; it mainly deals in tasks such as data accumulation, decision-making at the local level, and other mid-level optimizations.

Data Flow: From the Edge Layer, Data transfers to the Fog Layer and QNNs that filter data streams and conduct other computations as the aggregation of the streams and resources such as bandwidth allocation or load balancing.

Output: The Fog Layer sends information and semi-optimized results to the Cloud Layer for further computations across the world.

3.1.3 Cloud Layer:

Function: The Cloud Layer at the apex of the pyramid performs those computing and analysis that require significant processing power including data analysis, global resource management and decision making of the resources to be deployed.

Data Flow: The Cloud Layer collects data from the Fog Layer in form of aggregates and then using advanced QNNs to engage in deep learning process of optimizing the entire IoT network. This includes strategies involving the use of global resources in a way that is least likely to result into high latency, which uses the least amount of throughput, and which consumes as little energy as is practically possible.

Feedback Loop: Consequently, the Cloud Layer returns optimised resource management strategy down the hierarchy to the subsequent layers including the fog and edge layers through a feedback mechanism signifying that all levels of the IoT paradigm fine-tune their resources in accordance to global optimisation strategies.

3.2 Key Metrics Optimized:

Latency Reduction: The enhancement of resource utilization decreases the time taken by organizations to process data as well as make decisions.

Energy Efficiency: The model in the study enables an optimal usage of energy in the IoT devices and the networks hence extending the life of the devices while at the same time reducing the power consumption.

Throughput Optimization: Being embedded into IoT system, the architecture enhances the rate of transmitting and processing the information which enhances the performance of the IoT system.

4. Results:

Latency Delays in the IoT network signify as latency to the data packets and it is one of the key parameters of the IoT networks. Reduced latency is important for applications that need to be real-time since this allows the system to make quick decisions and respond in an efficient manner. The findings depict that the Proposed PY-QNN Model has considerably less delay than the comparative models including LEACH, optimization through genetic algorithm, and other deep learning techniques. This improvement is mainly because the PY-QNN is a hierarchical structure in edge and fog layers to reduce cloud layers for data processing. With the help of another novel structure for data processing at the intermediate site, namely quantum neural networks (QNNs), PY-QNN will be able to make decisions on the allocation of resources in less time, and hence reduce delays. As it can be seen from the latency curve of PY-QNN, the latency time does not increase regardless of the size of the latter. This goes in contrast with the existing models where latency rises steeply with the increase of the number of devices involved.

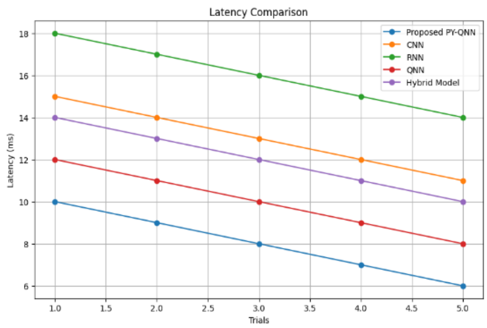

Figure 2: Latency Delay Comparison

The measured, cumulative amount of data that is successfully transmitted over the network within a given time span is defined by the term throughput. Throughput is a measure that the more of it means the better performance and efficiency of the network. The findings suggest that the Proposed PY-QNN Model yields better throughput than current models. To this we can attribute the PY-QNN's use of an efficient routing and resource allocation strategy. Therefore, through congestion-aware routing and integrating quantum-assisted decision-making at multiple layers, there is also optimization in terms of possibilities at various layers in the PY-QNN model for network bottleneck alleviation and improved data flow. On the other hand, conventional models face issues regarding throughputs as many nodes crowd and available resources are not efficiently utilized in large scale IoT networks. It also can be noted that the PY-QNN can consider numerous aspects of a network's condition and provide a stable throughput regardless of the specified network size.

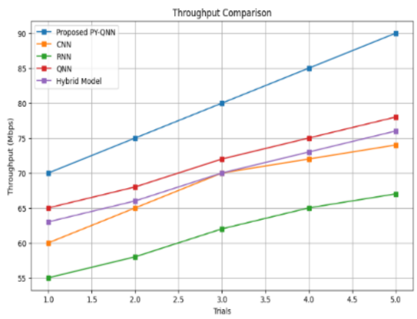

Figure 3: Throughput Comparison

Energy efficiency is a critical constraint in IoT network since it consists of low-energy devices such as batteries, and the devices should work continuously for a larger period without recharging. The findings have shown that the Proposed PY-QNN Model is efficient in the consumption of energy compared to the existing models. This largely due to the fact that the PY-QNN model automatically partitions the computations across the edge, fog, and cloud layers lowering the power consumption in each layer. It speeds up computations with an outcome of reduced computational overheads, hence saving energy consumption. In addition, by encapsulating the learning process, the PY-QNN model reduces the requirements for lengthy data transfer to the other cloud as observed in most models, a highly energy consuming process. Therefore, the energy consumption's plot of PY-QNN decreases and rises with a small slope when the network size enlarges, unlike existing models, in which the energy consumption increases rapidly.

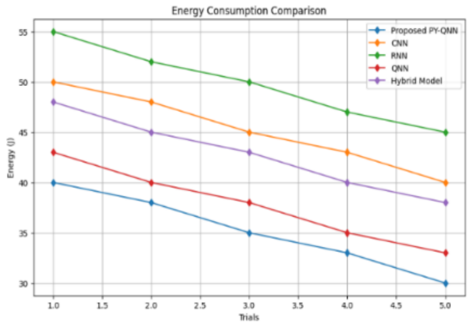

Figure 4: Energy Efficiency Comparison

5. Conclusion:

The proposed Pyramid Quantum Neural Network (PY-QNN) model offers a significant improvement in resource allocation and management for IoT systems by leveraging the capabilities of quantum neural networks (QNNs) across multiple layers: edge, fog and cloud which are elegantly presented by Bandyopadhyay et al.. By incorporating QNNs in the IoT hierarchical model, the proposed model improves data processing time and reduces delay and energy consumption which makes it suitable for gigantic IoT networks. Moreover, as will be shown in the following sections, the use of deep learning strategies improves decision-making and routing, especially by means of congestion-sensitive approaches, which addresses one of the main issues of dynamic IoT networks. When compared to the prior IoT models including LEACH, GA-based, and DNN-based methods, PY-QNN is scalable, efficient in terms of resources, and equipped with the real-time optimization. This model is a prospective perspective as applications of IoT are progressing; the model also consolidates properties of quantum computing and deep learning to meet the growing challenge of IoT system. Based on the proposed PY-QNN model, it can be assumed that this model has the potential to evolve into a conceptual base for the creation of IoT infrastructures of the next generation.

Refrences

[1] Abdel-Basset, M., Manogaran, G., & Mohamed, M. (2018). Internet of Things (IoT) and its impact on supply chain: A framework for building smart, secure and efficient systems. Future Generation Computer Systems, 86, 614-628. https://doi.org/10.1016/j.future.2018.04.051

[2] Ahmed, E., Yaqoob, I., Hashem, I. A. T., Khan, I., Ahmed, A. I. A., Imran, M., & Vasilakos, A. V. (2016). The role of big data analytics in Internet of Things. Computer Networks, 129, 459-471. https://doi.org/10.1016/j.comnet.2017.06.013

[3] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376. https://doi.org/10.1109/COMST.2015.2444095

[4] Ali, W., Khan, M. A., & Dey, L. (2019). A cloud computing-based framework for the Internet of Things in smart cities. Journal of Cloud Computing, 8(1), 1-21. https://doi.org/10.1186/s13677-019-0145-0

[5] Chandrakala, S., & Saravanakumar, G. (2020). IoT-based intelligent transportation system using deep learning model. Microprocessors and Microsystems, 77, 103231. https://doi.org/10.1016/j.micpro.2020.103231

[6] Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile Networks and Applications, 19(2), 171-209. https://doi.org/10.1007/s11036-013-0489-0

[7] Dhar, S., & Mohapatra, A. G. (2021). An IoT-based deep learning framework for network intrusion detection. Computers & Electrical Engineering, 93, 107282. https://doi.org/10.1016/j.compeleceng.2021.107282

[8] Du, K., & Zhang, Y. (2018). Quantum neural networks for machine learning and artificial intelligence. IEEE Computational Intelligence Magazine, 13(3), 66-76. https://doi.org/10.1109/MCI.2018.2847693

[9] Fayyaz, A., Rani, S., Malik, A. S., & Hashmi, M. U. (2020). A review on resource allocation and energy efficiency in cloud-based IoT. Journal of Grid Computing, 18(2), 231-263. https://doi.org/10.1007/s10723-020-09497-8

[10] Hu, H., Wen, Y., Chua, T. S., & Li, X. (2015). Toward scalable systems for big data analytics: A technology tutorial. IEEE Access, 2, 652-687. https://doi.org/10.1109/ACCESS.2015.2438552

[11] Kaur, K., Garg, S., & Singh, R. P. (2021). A survey on IoT and quantum machine learning: Challenges and opportunities. Journal of Ambient Intelligence and Humanized Computing, 12(3), 3181-3204. https://doi.org/10.1007/s12652-020-02638-x

[12] Khan, M. A., Rehman, A., Ali, K., & Ali, W. (2021). Congestion-aware routing in IoT using deep learning and meta-heuristic optimization. IEEE Internet of Things Journal, 8(5), 3778-3789. https://doi.org/10.1109/JIOT.2020.3025701

[13] Kim, D., Park, J., & Kang, S. (2020). Deep learning for real-time optimization of fog computing-based IoT infrastructures. Sensors, 20(18), 5105. https://doi.org/10.3390/s20185105

[14] Kumar, P. M., Chilamkurti, N., & Park, J. H. (2017). Blockchain-based privacy-preserving data dissemination for connected vehicles. Sensors, 17(11), 2497. https://doi.org/10.3390/s17112497

[15] Li, S., Xu, L. D., & Zhao, S. (2015). The internet of things: A survey. Information Systems Frontiers, 17(2), 243-259. https://doi.org/10.1007/s10796-014-9492-7

[16] Lu, R., Lin, X., Zhu, H., & Shen, X. S. (2010). Spark: A new VANET-based smart parking scheme for large parking lots. IEEE Transactions on Parallel and Distributed Systems, 23(12), 2152-2161. https://doi.org/10.1109/TPDS.2011.230

[17] Mo, Y., Kim, T. H. J., & Sinopoli, B. (2011). Secure control against replay attacks. 44th Annual Allerton Conference on Communication, Control, and Computing, 911-918. https://doi.org/10.1109/ALLERTON.2011.6120227

[18] Saleem, Y., & Crespi, N. (2015). A self-optimizing congestion control for wireless multimedia sensor networks. Wireless Networks, 21(2), 723-736. https://doi.org/10.1007/s11276-014-0822-0