International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Machine Learning and Prediction-Based Resource Management in IoT Considering Qos

Ravi C Bhaddurgatte, Vijaya Kumar B P, Kusuma S M

Abstract: ***Internet of Things (IoT) is one of the fast-growing technology paradigms used in every sectors, where in the Quality of Service (QoS) is a critical component in such systems and usage perspective with respect to ProSumers (producer and consumers). Most of the recent research works on QoS in IoT have used Machine Learning (ML) techniques as one of the computing methods for improved performance and solutions. The adoption of Machine Learning and its methodologies have become a common trend and need in every technologies and domain areas, such as open source frameworks, task specific algorithms and using AI and ML techniques. In this work we propose an ML based prediction model for resource optimization in the IoT environment for QoS provisioning. The proposed methodology is implemented by using a multi-layer neural network (MNN) for Long Short Term Memory (LSTM) learning in layered IoT environment. Here the model considers the resources like bandwidth and energy as QoS parameters and provides the required QoS by efficient utilization of the resources in the IoT environment. The performance of the proposed model is evaluated in a real field implementation by considering a civil construction project, where in the real data is collected by using video sensors and mobile devices as edge nodes. Performance of the prediction model is observed that there is an improved bandwidth and energy utilization in turn providing the required QoS in the IoT environment.***

Index Terms: ***Internet of Things [IoT], Machine Learning (ML), Quality of Service [QoS], Multi-layer Neural Network [MNN]***

I. INTRODUCTION

With the adaptation of newer technologies, increased Internet speed and large volume of IoT sensors, data, there is a need for proven techniques for analyzing and inferring from the large sensor data in the IoT domain. In this view, the decades old Artificial Intelligence (AI) has gained momentum with wide spread adaptations of ML techniques in many domains and systems [1, 2, 3, 9 and 1317]. ML techniques are becoming more essential and integral part of today's systems as the need for continuous and automated learning by experience is increasing due to dynamics of systems and services. The applications of AI/ML techniques for pattern and voice recognition, image processing need to be extended for automated learning from large data volume produced from IoT sensors considering QoS as key element in implementations. There are considerably minimal implementations of ML techniques applied in IoT for improving the QoS [4, 11, 15, 16, and 22]. The key contribution of this paper is to design and implement a prediction model using Multi-layer Neural Network (MNN) for QoS provisioning in IoT environment with respect to bandwidth and energy resources. Implementations over the realistic IoT environment is carried out using the proposed MNN model for improved QoS in terms of bandwidth and energy. The sections of the paper are organized as follows, section II discusses the literature review, findings and brief summary. The description of the proposed model is given in section III. Section IV explains the model architecture for an integrated IoT environment which uses a MNN algorithm for prediction and an algorithm for decision making. Implementation and results are discussed in section V, finally the conclusion and future directions provided in section VI.

Revised Manuscript Received on 30 July 2019.

***** Correspondence Author

Ravi C. Bhaddurgatte*, Research Scholar, Computer Science and Engineering Department, Jain University, Bangalore, India.

Dr. Vijaya Kumar B. P., ISE Department, M S Ramaiah Institute of Engineering and Technology, Bangalore, India.

Kusuma S. M., Telecommunication Engineering. Department, M S Ramaiah Institute of Engineering & Technology, Bangalore, India.

II. LITERATURE REVIEW

From the review of research and implementations of AI, ML and Deep Learning (DL) techniques in IoT and related areas, we find the adoption of ML in various systems and technology domains like wireless networks, Software Defined Networks (SDN), HetNets, Cloud Services, Data Centers, security etc [1, 5, 6, 7 and 8], also in many business domains, industry applications and block chain implementations [14, 20-22 and 25]. Deep Learning in specific has more implementations and there are various frameworks defined and implemented for DL [1, 8, 12 and 10]. Research works explored in which there are frameworks proposed and implemented using ML techniques in the domains of wireless networks, SDNs, Cloud Services etc., [1, 7 and 20]. Also, for research and implementation there are hardware devices used as edge based computing to run ML algorithms (like Deep Learning/CNN) in the IoT environment [1, 3, 10, and 8]. The authors in [2] review the architectures, accelerators and optimizations for varied software and hardware implementation of ML and review the open challenges to be addressed, detailed ML implementation review is done [17].

There are new MNN based models like Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) are used as deep learning techniques [32, 33]. From the review we find that the majority of the implementations of ML techniques are addressing four major category of problems: (i) Prediction (ii) Adaptation (iii) classification and (iv) Routing which are summarized below.

Prediction models: Industrial machines and their maintenance requirements are predicted using ML technique (regression), data from the IoT sensors used for training [26]. Work on processing of complex events for prediction using an adaptive ML algorithm for pro-active and dynamic IoT applications in [21]. ML technique of neural network regression applied for data compression and prediction of time series data in industrial IoT are given in [14]. Authors in [3] presents framework for IoT devices for prediction and inference using deep learning considering QoS. Authors in [5] compare various ML algorithms for predicting the performance of Web Services for two QoS parameters throughput and response time. One of the ML learning implementation in [4] uses the offloading technique and optimize the performance of IoT applications using Deep Learning. More practical approach of bringing DL in IoT environment was verified and compared by in phases using library with basic building blocks for computing [10]. Roohollah et al. [6] implemented Deep Learning in IoT by using existing solutions to address the challenges of energy consumption and deployment using labeled data. Xuyu et al. defined Deep Learning framework [1] for radio frequency (RF) sensing in IoT and implementation using proposed framework. Location management in mobile networks is carried out in [18] using prediction based Neural Network model by considering mobile host movement patterns as time series data and connectivity management using neural networks in [31]. There are works for QoS in mobile networks using lineal programming resource reduction principle [34], and artificial intelligence based admission control of traffic for QoS in mobile networks [35].

Adaptability: Neural network combined with firefly is designed for effective self-adaptability and energy efficiency in boiler plants [20]. ML techniques also applied for link adaption in a wireless network to improve the energy efficiency/throughput by considering QoS constraints [27]. To tackle the issues of malware, QoS, traffic control ML techniques are adopted in [28], similarly ML technique is implemented for adaptive interference suppression in WSNs [29].

Classification & routing: Semi-supervised learning is implemented in Software Defined Networking [SDN] with a framework for traffic classification [7], internet multimedia for QoS along with other networking traffic classification based on data and control packets as training data sets [30]. Similarly the clustering algorithms like k-means, and decision tree learning algorithms are used for performance improvement with respect to QoS parameters. A detailed review of ML algorithms used for networking, classification and open research challenges is done by the authors in the paper [18]. Here we summarize the review into three parts. One, the design and development of methodologies and models supervised and un-supervised methods is across multiple domains and systems like – Cloud & Data Centers, HetNets, SDNs, WSNs, IoT, Industrial applications, e-commerce applications and more. Secondly, various ML/DL frameworks available (including open sources) and some of them are adopted in embedded devices / SoC's as edge nodes in IoT environments. Lastly, ML techniques are majorly implemented for solving problems broadly categorized into major groups of prediction, adaptation, classification and routing. Similarly, to extract the dynamic behaviors and variations of systems and users requirement in providing Quality of Services in IoT is a continued research which needs to be addressed. This is being addressed in this paper by designing a prediction model in IoT environment using a MNN based deep learning algorithm by considering the QoS parameters discussed in the subsequent sections.

III. PROPOSED MULTI-LAYER NEURAL NETWORK MODEL

The information gathered at edge nodes in IoT environment with users, systems and QoS requirements will vary with respect to time, location and application context and is chaotic in nature. For such scenario there is a requirement to build a learning and prediction model for the data gathered at the edge nodes in the context of unseen and varying scenarios for QoS. The data flow across different modules of the proposed model is shown in (*Fig. 1*) and algorithm steps are shown in the list (*Steps 1*.). The different modules and functions of the model are as described below. The model will have modules for '*context setting'* for setting hyper parameters and environment parameters, '*prediction module'* for learning and prediction of events. The '*training data*' and '*predicted data'* are stored in database for training and prediction. The prediction results are used by decision '*making module'* at the resource constrained perception layer which can make decision to optimize the resource requirements like bandwidth and energy for achieving QoS in the IoT system.

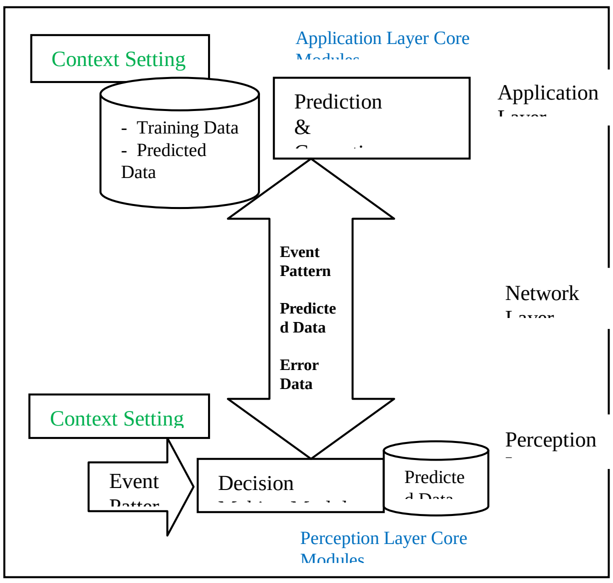


Fig1. Data flow for Prediction and Decision Making in IoT

Before we proceed on detailing the model architecture, algorithm and implementation we define some of the relevant definitions here.

Event Pattern (E_P) is the set of event patterns till time T_n, here n is the time intervals as day/ hour/ minute/ seconds/ milliseconds recorded at an IoT edge node. The events data at regular intervals of time t_1, t_2, ...t_n is represented as event pattern, E_P. The example edge nodes can be a CCTV camera, mobile device, sensor or any of the IoT edge nodes. Consider the set of event pattern, $E_P = \{E_1, E_2, ...E_n\}$ be the set of event patterns recorded in an IoT environment. Let the event pattern E_i represent the pattern at time interval t_i and here we represent the pattern E_i as a pair of values like (O_i, L_i). Here for any event instance E_i the example values are like E_i = {*Object Type, Geo Location*} at time T_i = 8:45 AM.

- O_i represents the *object type* at i^{th} time interval. The event type will depend on the context of IoT environment. For example if we consider the IoT application where we want to track the inward and outward movement of objects, the objects is represented as $O_i \in$ {*Truck, Car, People, and Two Wheeler etc.*}.

L_i, is the *geo location* with set of longitude and latitude values at the i^{th} time interval. The geo location may be represented by longitude and latitude or by the mapped geo location or area based on the context of IoT environment.

Parameters vs hyper parameters, here the parameters are basically the values which are learnt during training (*object type* and *location*) and used for training. Whereas hyper parameters in our definition are defined and set beforehand (for example '*season*' or '*# of phases*', '*project phase*' or '*project type*' or 'project size' , 'duration' etc.) and not learnt.

Context of IoT environment: The model is for generic event patterns prediction in an IoT application environment. The example contexts would be like: projects execution, objects movement in a factory environment, gated entry and exit environments where the movements patterns to be predicted based on the defined hyper parameters of the context and applicable IoT environment.

Learning Data, is basically the set of training data used for training the model under consideration. The sample training data is represented as set of event patterns as shown in table (*Table 1*.) below with set of input pattern values and desired output.

Table 1. *Sample Input Events data pattern*

Event Sub pattern	**1**	**2**	**3**	**4**	**5**
Input Event1	*e1*	*e2*	*e3*	*e4*	*e5*
Input Event2	*e5*	*e3*	*e1*	*e2*	*e4*
Input Event3	*e2*	*e5*	*e5*	*e3*	*e3*
Input Event4	*e2*	*e4*	*e2*	*e5*	*e1*
Input Event5	*e1*	*e5*	*e1*	*e3*	*e4*
Desired Output Event	***e2***	***e5***	***e1***	***e3***	***e4***

A. *Multilayer Neural Network (MNN) based Deep Learning Model for Prediction*

Here the model considers context based IoT environment that adopts MNN based deep learning method for prediction. Deep learning normally handles any complex scenarios given the considerable data set with proper neural network architecture, either for prediction, adaptation, classification and routing. Since the model is generic, can be used for any of the context to predict the events and activities. There are different deep neural network models: (a) feed-forward model in which information moves in only one direction (b) forward-propagation model with back-propagation for adjusting the weights for improving the results (c) Recurrent Neural Network (RNN) model which maintains the internal memory and information is looped back and used along with recent put for making predictions more accurate for temporal and sequence based applications. Unlike feed-forward neural networks, RNN can map one input to many and many inputs to one, many inputs to many outputs. Selection of particular neural network model / algorithm needs a clear understanding of the complexity of data type, application context and data set. From the understanding of the chosen data set we propose to implement prediction algorithm using Long Term Short Memory (LSTM) neural network model which uses internal memory. LSTM is an improved version of RNN under deep learning that best suits for temporal and sequential data. Some key terminologies listed below in *List 1*.

List 1. *Some of key terminologies / concepts of LSTM Model*

- LSTM consists of single input layer, one or more hidden layer and an output layer.
- Each of the LSTM layer are recurrent and will have memory blocks for storing information which are called cells.
- The property of selective remembering or forgetting is implemented by LSTM architecture using three gates: input gate, forget gate and output gates.
- Input gate serves by providing additional information to the '*cell state'* which will be needed by LSTM.
- The forget gate works to remove the information from the '*cell state'* which is not required by the LSTM.
- The output gate works to filter the unwanted information from '*cell state'* and pass it to the next cell.
- Known issues of RNN vanishing gradients, solved in LSTM to achieve more accuracy and relatedly shorter training duration.

B. *Learning / Training Steps*

Steps1. High-level Steps: Define LSTM model, Training & Prediction

BEGIN

1. *Dataset Preparation: Clean the data and convert column representation to row representation for sequential representation.*
2. *Define LSTM model with proper initial values and layers as:*
   - *Container to hold the layers (mostly sequential) and # of Memory unit values.*
   - *First LSTM layer for inputs with data rows, lag values and columns of the data set*
   - *Dropout layer after each LSTM layer to avoid over fit.*
   - *Dense output layer by defining number of outputs and activation function*
   - *Dropout layer after each LSTM layer to avoid over fit.*
3. *Compile the model by providing optimization algorithm to train the network and loss function for the model.*
4. *Train and Fit the model providing values of: batch_size, number_of_epochs and validation_split (like 80:20).*
5. *Evaluate the model for a separate data set, collect and verify the required metrics*
6. *Make Predictions of events as required by the data set.*
7. *Record outputs, graphs as required.*

END

In general there is no rule on number of hidden layers and memory cells for LSTM, but could depend on the context and application domain. General practice is to start the model with one hidden layer and 2 or 3 memory cells and test the model, then based on the performance the model one can add additional layers and memory cells as required for improving the model for the complex prediction problems. The steps of defining the model and prediction are listed below [*Steps 1*].

The learning model is designed to take set of features (training data) for each parameter of varied objects. The parameters could identify different objects (for example trucks, cars and people) depending on the context. For the input event pattern E_n at time T_n, with values (O_i, L_i) the MNN model predicts the event pattern E_{n+1} at time instance T_{n+1}. The values of E_{n+1} will determine the *object type* and *geo location* of the event pattern E_{n+1}.

IV. THE ARCHITECTURE WITH MNN MODEL FOR PREDICTION

There are specific models, architectures and frameworks for addressing QoS in IoT, but IoT as a domain is very dynamic having diversified applications and usage contexts. Here we present a model which can have adaptable functional modules like: prediction, computation, decision making which may fit across layers based on the context of IoT systems. The *Table 2* lists some of the core functional modules of IoT systems across layers [23 Section II] and "to be published" [24] and adaptable modules based on the context. The level of processing, analysis and application functions across layers depends on the context, resource constraints and need of IoT applications. The model here is implemented with a three layer IoT architecture [*Fig 2.*] in which the MNN model for learning and prediction adopted at the application layer. The input data and the training data set would depend on the defined hyper parameters and parameters of specific IoT application or context.

Table 2: Typical Functional modules across the layers in IoT Systems

Note: The functional modules listed in the table in italics are adoptable across layers.

Application Core Layer Functions	Network Layer Core Functions	Perception Layer Core Functions	Context Based Functions Can be adapted *'Any Layer'*
➢ **Data Reception** ➢ *Processing* ➢ *Analysis* ➢ *Application* ➢ …..	➢ **Routing** ➢ **Transport** ➢ *Processing* ➢ *Analysis* ➢ *Application* ➢ …..	➢ **Data Gathering** ➢ **Data Sending** ➢ *Processing* ➢ *Analysis* ➢ *Application* ➢ …..	➢ **Adaptability** ➢ **Decision Making** ➢ *Processing* ➢ *Analysis* ➢ *Application* ➢ …..

The model is a novel approach which is more generic and can automatically learn for any new set of parameters with a new set of features at any stage based on the context defined. The predicted '*event pattern'* is stored at the edge nodes (or can be retrieved from cloud) in the local storage. As per the limitations of IoT edge nodes, because of the limited computing power at the edge, computationally simple '*decision making'* algorithm is implemented at the edge node. The decision making algorithm [*Algorithm 1.]* can mark the event as '*error'* when the event occurred at the instance not predicted by the prediction model. The model feeds the input events to the training data set for future experience and learning.Based on the training data set (historical data) the model predicts the event at time t_{n+1}. The prediction of the event E_{n+1} by MNN at time interval t_{n+1} is based on the input event sub pattern {E_{n-k+1}, E_{n-k+2}...E_n}. The predicted output event would determine the next output event E_{n+1} as {O_{n+1}, L_{n+1}}.

Algorithm 1. Decision Making and marking the prediction error

BEGIN

1. *For each Event*
2. *Use the prediction result to compare with 'input event'*
3. *If event is as per prediction make decision as 'Error' or 'No Error'*

   THEN

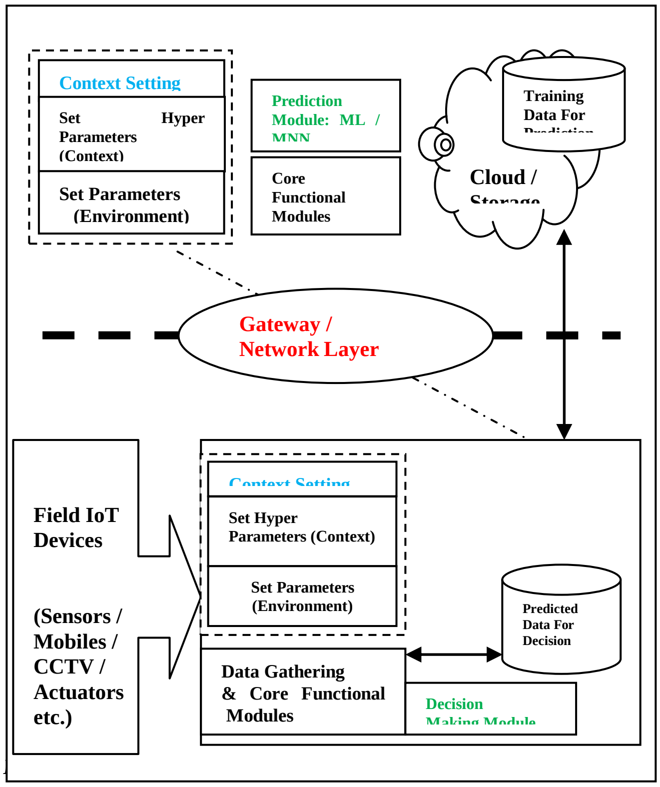
   a. *Event processed with allocated bandwidth.*
   b. *Add input event to 'Training Set'*

   ELSE

   a. *Event processed with allocated bandwidth.*
   b. *Add input event to 'Training Set'*
   c. *Mark 'error'*

END

The model loads the training data of the corresponding context based on the hyper parameters and IoT specific parameters defined [*Fig 2.*]. Prediction based QoS optimization in implemented for optimizing the QoS by optimizing the parameters under consideration: *bandwidth* and *energy*. The model adopts the low computing '*decision making'* algorithm at edge nodes for comparing the occurred '*event pattern'* with the predicted event(s) to allocate the resources in an optimized way for achieving the QoS in the IoT environment.



As discussed earlier the event pattern of the context would depend on the parameters like – time, location and season. For illustration we have considered data of object movements (*object type: labor and material*) into a civil construction project site. The hyper parameters for the context are: (i) *project type*, (ii) *project phase*, (iii) *duration* and (iv) *project size*. We have taken the production data of projects of similar type, duration and size as training and test data. Here the time interval t_n, can be considered either as daily, hourly or weekly as per the span of the project execution which has to be modulated as per the project type.

International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

For example for a construction project, we have taken the live data (object movements) of multiple projects of a client for training the prediction model. The high level steps of the overall approach is given in *Algorithm 2* below.

Algorithm 2. **High Level Steps of the Overall Approach**

BEGIN

1. *Acquisition of historical data /* Using Android APP */*
2. *Set Context parameters /* Time, project phase, duration*/*
3. *Repeat steps 4 to 6 for Varied values /* data conditions, epochs, windows */*
4. *Prediction (phase, duration) /* LSTM */*
5. *Make decision /* Decision Algorithm */*
6. *Measure the utilization /* Bandwidth, energy */*

END

V. IMPLEMENTATION AND RESULTS

Implementations are carried out to evaluate the performance of the prediction model using MNN based deep learning algorithm LSTM for different scenarios. The real time data collected from the construction project sites using a separate Android mobile APP at different instances of event occurrences. The data consists of project name /id, date, object type, images and location.

A. *Prediction and Decision Making for QoS Provisioning*

The prediction model is implemented by using Python (*Keras*) for Deep Learning / LSTM. The real time data of two large construction companies consisting of 50 projects is collected, cleaned, pre-processed and used for learning. Here a stage-wise data analysis is carried out for varied time lines and of different project durations for predictions. We have implemented Long Short Term Memory (LSTM) MNN model as a deep learning to analyze the sequential data collected from the above projects. Here the model uses about 80% of data for training and 20% of data for validation for evaluating the performance with respect to prediction. The results indicate that when the data is raw the '*absolute mean error*' is high and when we use more cleaned data the '*absolute mean error*' gets reduced and as the epochs increase, the predictions become more stable and the '*train*' and '*valid*' has become more stable. The graphs below show the results with '*absolute mean error*', '*Train*' and '*Valid*' for varied iterations (epochs) and window sizes. The graphs generated using matplotlib in graphs (*Fig 1.* to *Fig 7.*) show the stability of the model as the valid data passed thru and reduction in error for '*Train*' and '*Valid*'.

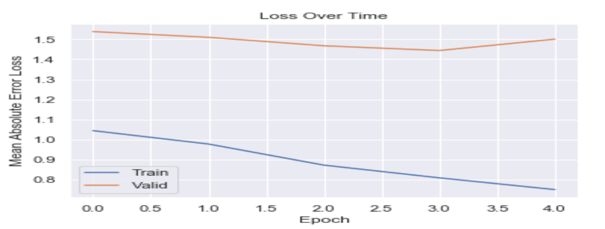


Fig 3. : *Raw Data; Window: 5 & Epoch: 5*

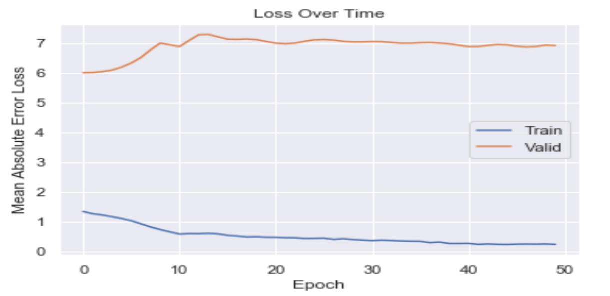


Fig 4. : *Raw Data; Window: 5; Epoch: 50*

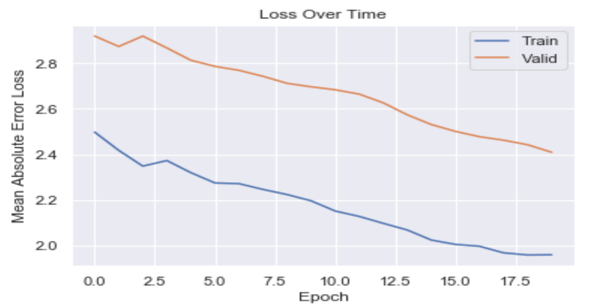


Fig 5. : *Cleaned Data; Window: 5; Epoch: 5*

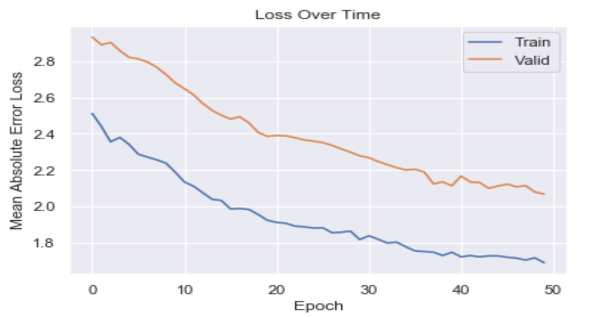


Fig6. : *Cleaned Data; Window: 5; Epoch: 50*

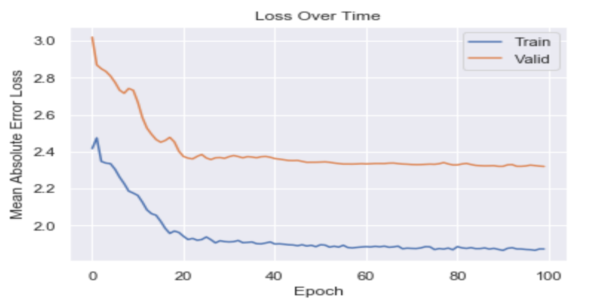


Fig 7. : *Cleaned Data; Window: 5; Epoch: 100*

Bandwidth and Energy utilization for Improved QoS

Here the data collection is carried out periodically (i.e. in seconds, minutes and hourly etc.) and considering different traffic speed, periodicity, events and normalized with proper formatting to train the prediction model and for simulations.

Bandwidth utilization: Based on the experimentations carried out for different project scenarios with prediction and regression models, we found that there is an improved bandwidth utilization. Here the utilization is defined by total data transfer, network speed and duration of data transfer, the equation (1). Bandwidth utilization = (Transferred data volume) / (data transfer duration * network speed)

- (1) As a first step we compute the bandwidth used by the applications as per the events, with an allocated bandwidth of range 10 Mbps to 1 Mbps. The results of bandwidth utilization without the proposed model and with proposed model are shown in graph (*Fig 9.*). From the results it is clear that there is an increased bandwidth utilization with the proposed prediction model. The average utilization would vary between 10 to 30% based on the context and systems under consideration, and for our tested scenarios we find an increase of 10.83% bandwidth utilization using prediction model for the same data volume.

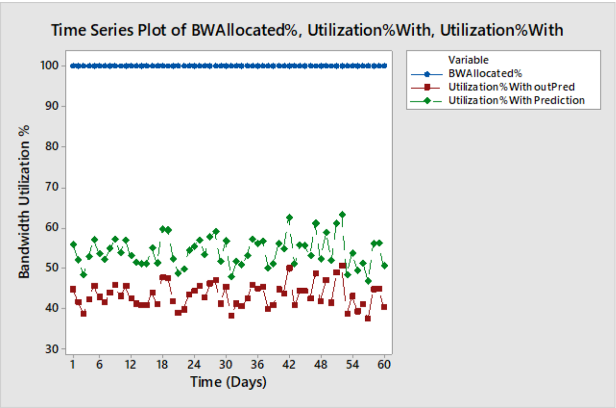


Fig 8. : *Bandwidth Utilization% without-prediction & with-Prediction*

Energy utilization: Normally the energy consumption varies from device to device based on the type, model, battery, usage patterns etc. For example, in our experimentation we have taken a mobile device that consumes an average energy of 0.012 kWH units. From the results it is observed that the average energy utilization is always above when compared to utilization without prediction since the traffic flow and applications requirements are known earlier, and helps in allocating the prescribed energy either to be in sleep mode, hibernate state or active mode, hence this has enhanced the utilization of energy, the same is reflected in the results shown in graph (*Fig. 9*).

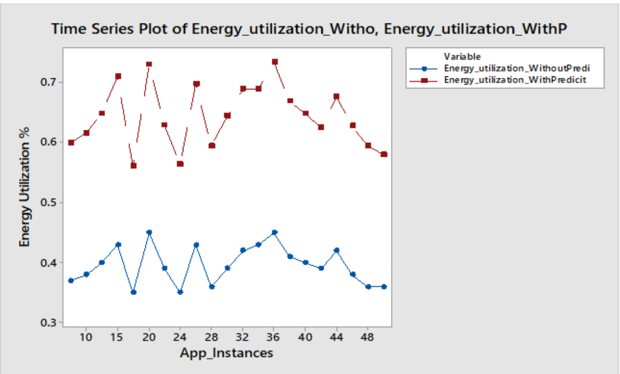


Fig 9. : *Energy Utilization% without-prediction & with-Prediction*

VI. CONCLUSION AND FUTURE WORK

Machine Learning (ML) techniques have an increased implementations in all technology domains including IoT and many business domains. QoS has become a critical issue in IoT environments since there is a huge explosion of connecting devices, sensors, data and usage. ML techniques gives very convincing results to facilitate the required QoS in IoT environment. From the recent reviews the quality of service (QoS) and machine learning (ML) to improve the system performance is emphasized more to address some of the critical challenges. Hence, this paper consider the adoption of ML for QoS provisioning in IoT environment.

The work proposed includes a multi-layer neural network prediction model for optimal resources utilization in IoT environment for QoS provisioning. The model uses MNN based LSTM learning for predicting the events with real time data from the civil construction projects. The model is verified with and without prediction and the result shows an improved bandwidth and energy utilization, that in turn has provided the required QoS measured in terms bandwidth and energy utilization. From the results it is concluded that the prediction model helps the system to allocate the required resources as per the traffic, applications and events in turn enhancing the resources utilization for achieving required QoS. In the future this model can be improved with other QoS parameters like security, data accuracy etc. and may be considered with advanced techniques like block chain and other learning algorithms.

VII. ACKNOWLEDGMENT

Thanks to Jain University for providing access to resources and providing opportunity to work on research activities. Thanks to colleagues of ISE department at MSRIT for support and ePrayag Software Private limited for facilitating the test environment and data for verifying the results.

REFERENCES

1. Xuyu Wang, Xiangyu Wang "RF Sensing in the Internet of Things: A General Deep Learning Framework" IEEE Communications Magazine (Volume: 56, Issue: 9 , Sept. 2018).
2. Muhammad Shafique, Theocharis Theocharides, Christos-Savvas Bouganis, Muhammad Abdullah Hanif et al. "An overview of next-generation architectures for machine learning: Roadmap, opportunities and challenges in the IoT era" 2018 Design, Automation & Test in Europe Conference & Exhibition
3. Woochul Kang; Daeyeon Kim et al "Poster Abstract: DeepRT: A Predictable Deep Learning Inference Framework for IoT Devices" 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)
4. He Li ; Kaoru Ota ; Mianxiong Dong "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing" IEEE Network 2018
5. Sunil Kumar ; Manish Kumar Pandey et al. "Comparative study on machine learning techniques in predicting the QoS-values for web-services recommendations" International Conference on Computing, Communication & Automation:15-16 May 2015, Date Added to IEEE Xplore: 06 July 2015
6. Roohollah Amiri, Hani Mehrpouyan et al. "A Machine Learning Approach for Power Allocation in HetNets Considering QoS" arXiv.org: arXiv:1803.06760v1 [cs.IT] 18 Mar 2018
7. Pu Wang ; Shih-Chun Lin ; Min Luo "A Framework for QoS-aware Traffic Classification Using Semi-supervised Machine Learning in SDNs" 2016 IEEE International Conference on Services Computing (SCC): 27 June-2 July 2016, Date Added to IEEE Xplore: 01 September 2016

8. Shuochao Yao et al. "Deep Learning for the Internet of Things" www.computer.org/csdl/magazine: May 2018, pp. 32-41, vol. 51
9. Hardeep Singh "Performance Analysis of Unsupervised Machine Learning Techniques for Network Traffic Classification", 2015 Fifth International Conference on Advanced Computing & Communication Technologies; 21-22 Feb. 2015 Date Added to IEEE Xplore: 06 April 2015
10. Jie Tang ; Dawei Sun "Enabling Deep Learning on IoT Devices", Computer (Volume: 50, Issue: 10 , 2017); Page(s): 92 – 96; Date of Publication: 03 October 2017
11. Liumeng Song ; Kok Keong Chai et al. "QoS-Aware Energy-Efficient Cooperative Scheme for Cluster-Based IoT Systems", IEEE Systems Journal (Volume: 11 , Issue: 3 , Sept. 2017) Page(s): 1447 – 1455; Date of Publication: 23 June 2017
12. Vivienne Sze ; Yu-Hsin Chen et al, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey", Proceedings of the IEEE (Volume: 105 , Issue: 12 , Dec. 2017), Page(s): 2295 – 2329, Date of Publication: 20 November 2017
13. Janice Canedo et al. "Using machine learning to secure IoT systems" Conference Paper · December 2016 DOI: 10.1109/PST.2016.7906930 Conference: 2016 14th Annual Conference on Privacy, Security and Trust (PST)
14. Junmin Park, Hyunjae Park et al. "Data compression and prediction using machine learning for industrial IoT", 2018 International Conference on Information Networking (ICOIN), Date of Conference: 10-12 Jan. 2018, Date Added to IEEE Xplore: 23 April 2018
15. Ling Li ; Shancang Li "QoS-Aware Scheduling of Services-Oriented Internet of Things", IEEE Transactions on Industrial Informatics (Volume: 10 , Issue: 2 , May 2014); Page(s): 1497 – 1505 Date of Publication: 17 February 2014
16. Khanouche Mohamed Essaid et al. "Energy-Centered and QoS-Aware Services Selection for Internet of Things", IEEE Transactions on Automation Science and Engineering 13(3):1-14 · April 2016
17. Laizhong Cui, Shu Yang, Fei Chen et al. "A survey on application of machine learning for internet of things", International Journal of Machine Learning and Cybernetics, August 2018, Volume 9, Issue 8, pp 1399–1417
18. Raouf Boutaba, Mohammad A. Salahuddin et al. "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities", Journal of Internet Services and Applications, December 2018
19. Vijaya Kumar B P et al. "Prediction-based location management using multilayer neural networks", Journal of the Indian Institute of Science, Vol 82, No 1 (2002)
20. Xingquan Zuo ; Guoxiang Zhang "Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud", IEEE Transactions on Automation Science and Engineering (Volume: 11 , Issue: 2 , April 2014), Page(s): 564 – 573, Date of Publication: 02 August 2013
21. Adnan Akbar, Francois Carrez et al. "Predicting complex events for pro-active IoT applications", 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT): 14-16 Dec. 2015, Date Added to IEEE Xplore: 21 January 2016
22. Christos Chrysoulas, Maria Fasli et al. "A service oriented QoS architecture targeting the smart grid world & machine learning aspects", 2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech), 13-15 July 2016, Date Added to IEEE Xplore: 01 September 2016
23. Ravi C Bhaddurgatte and Vijay Kumar B P, "A Review: QoS Architecture and Implementations in IoT environment", Research & Reviews: Journal of Engineering and Technology, Sept 2015
24. Ravi C Bhaddurgatte and Vijay Kumar B P, " A Cross Layer QoS Framework for Hetrogeneous IoT Environment", IEEE International Conference, ICIETES, Sept 2018, "to be published"
25. Nguyen Cong Luong, Zehui Xiong et al. "Optimal Auction for Edge Computing Resource Management in Mobile Blockchain Networks: A Deep Learning Approach", arXiv.org , arXiv:1711.02844, Fri, 17 Nov 2017 03:09:47 UTC
26. A Kanawaday et al. "Machine learning for predictive maintenance of industrial machines using IoT sensor data", 2017 8th IEEE International Conference, ieeexplore.ieee.org
27. Tushar Desai, Hitesh Shah, "Energy efficient link adaptation using machine learning techniques for wireless OFDM", 2016 International Conference on Inventive Computation Technologies (ICICT), 26-27 Aug. 2016, Date Added to IEEE Xplore: 26 January 2017
28. Takamitsu Iwai, Akihiro Nakao et al, "Adaptive mobile application identification through in-network machine learning", 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), 5-7 Oct. 2016, Date Added to IEEE Xplore: 10 November 2016
29. Per Lynggaard et al, "Using Machine Learning for Adaptive Interference Suppression in Wireless Sensor Networks", IEEE Sensors Journal (Volume: 18 , Issue: 21 , Nov.1, 1 2018), Page(s): 8820 – 8826, Date of Publication: 24 August 2018
30. Zaijian Wang ; Yuning Dong, "Internet multimedia traffic classification from QoS perspective using semi-supervised dictionary learning models", China Communications (Volume: 14 , Issue: 10 , Oct. 2017), Page(s): 202 – 218, Date of Publication: 14 November 2017
31. B. P. Vijay Kumar, P. Venkataram, "A Neural Network-Based Connectivity Management for Mobile Computing Environment", Nov 2003, IEE Proceedings - Communications
32. Ian Goodfellow, "CHAPTER 10. SEQUENCE MODELING: RECURRENT AND RECURSIVE NETS",www.deeplearningbook.org
33. Yu Wang, "A new concept using LSTM Neural Networks for dynamic system identification", 2017 American Control Conference (ACC), 24-26 May 2017, Date Added to IEEE Xplore: 03 July 2017
34. B.P. Vijay Kumar; P. Venkataram et al, "A LP-RR principle-based admission control for a mobile network", Published in: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) (Volume: 32 , Issue: 4 , Nov. 2002), Page(s): 293 – 306, Date of Publication: Nov. 2002
35. B.P. Vijay Kumar; P. Venkataram, "A LP-based Admission Control Using Artificial Neural Networks for Integrated Services in Mobile Networks", Wireless Personal Communications: An International Journal archive, Volume 20 Issue 3, March 2002, Pages 219-236

AUTHORS PROFILE

Mr. RAVI C BHADDURGATTE: received his M. Tech degree in Computer science and Engineering from Manipal Institute of Technology (M I T), Manipal, Mangalore University with distinction in the year 1997, Bachelor's degree in Computer Science and Engineering from Govt. B. D. T. College of Engineering, Davanagere, Mysore University, India. He has 5 years of teaching experience and 21 years of experience in IT Services and currently working as Managing Director at ePrayag Software Pvt ltd. and pursuing research (Ph. D) at Jain University, Bangalore, India. His major areas of interest and research are Database systems, Business Intelligence and data warehousing, Big data and Analytics, Internet of Things (IoT), QoS in IoT, M2M communications, Cloud Computing, SDN and Mobile Computing.

Dr. VIJAYA KUMAR B. P. [SMIEEE] received his Ph. D degree from the Department of Electrical Communication Engineering, Indian Institute of Science (IISc), Bangalore in 2003, M. Tech degree in Computer Science and Technology from Indian Institute of Technology, IITR, with honours in 1992 and Bachelor's degree in Electronics and Communications from Mysore University with Distinction in the year 1987. He is currently heading the Department of Information Science and Engineering, M.S.Ramaiah Institute of Technology, Bangalore, Karnataka, where he is involved in research and teaching. His major areas of research are Computational Intelligence applications in Mobile, Ad-hoc and Sensor networks. He is having 26 years of Teaching and 20 years of Research experience. He has published around 65 papers which include journals, conferences and project proposals. His name is listed in Marquis Who's Who in the World, and Marquis Who's Who in Asia, 2012. He is a Senior Member of International Association of Computer Science and Information Technology (SMIACSIT), IEEE Senior Member and Life member of India Society for Technical Education (LMISTE).

Ms. KUSUMA S. M. [MIEEE, LMISTE] is working as an Assistant Professor in Telecommunication Engineering department, RIT. Completed her undergraduate from Mysore University in Electronics and Communication Engg., M. Tech in VLSI and Embedded systems from Visveswarayya Technological University, and presently pursuing her research work towards Sensors deployment in Embedded Sensor networks. Her areas of interests are Wireless Sensor Networks, Wireless Mobile Communication, Internet of Things, Embedded System Design, MIMO Technology, Machine learning and 4g/5g Technology.