pCWoT-MOBILE: A Collaborative Web Based Platform For Real Time Control In The Smart Space

E.N. Osegi[1*]
National Open University of Nigeria (NOUN)
emmaosegi@gmail.com; nd.osegi@sure-gp.com
234-8025838364

V.I. Anireh
Rivers State University (RSU), Port-Harcourt, Nigeria
anireh.ike@ust.edu.ng

C.G. Onukwugha
Federal University of Technology Owerri (FUTO), Imo State, Nigeria
onukwugha2000@yahoo.com

ABSTRACT
The Internet of Things (IoT) has proven to be a veritable systems approach to the problem of ubiquitous or everyday computing services. This paper presents an idea of using machine intelligence, in a predictive Collaborative Web of Things real time control platform called *pCWoT-MOBILE*, which enables web based technology to be deployed in smart spaces using a pervasive predictive and persuasive framework. We introduce some important performance criteria for evaluating our proposed system. We also show how our system can be applied as a Collaborative Real Time Control (CRTC) device in a smart space.

Keywords: Machine Intelligence, pCWoT, Predictive Real Time Control, Smart Space, Web of Things.

1. BACKGROUND

The Internet of Things (IoT) industry represents an important area of the application of the information age in the society. The industry currently boast of over 1 billion connected IoT devices worldwide, which goes to say that it has come of age. In the academia and industry, there has been renewed interest in researching better models of smart spaces, particularly as it applies to mobile IoT devices and how they can be of very good use in varied number of human and artificially generated tasks. However, this interest is challenged by a number of factors several of which include the cost of operating these devices, integration requirements, switching time, throughput and other mobility/security considerations. Current interests specifically focus on adapting miniature networks of digital electronic components for web-based online things as a service which is good as long as they remain cost-effective, do not lead to service downtimes and are able to infer what happens next; it is therefore worth desirous that we should be able to develop systems that can be useful in this sense.

[1] Corresponding author: Department of Information Technology, National Open University of Nigeria (NOUN)

2. PREVIOUS WORK

The research in the area of Internet of Things (IoT) solutions has been considered somewhat arbitrarily. For instance in the following works **(Ambroz, 2017; Martin-Garin et al, 2018)**, the focus has been on cost effectiveness while on the other hand there have been in a need for security features in IoT device architectures in smart Home Automation Systems (HAS) and production systems **(Jacobsson et al, 2016; Tomiyama & Moyen, 2018)**. In **(Onukwugha & Osuagwu, 2014)**, an end-user programming model was proposed for mobile smart spaces.

In the field of Short Messaging Service (SMS) for real-time control of devices over the air , smart SMS control solutions have been conducted in **(Onukwugha & Asagba, 2014)** and using genetic optimizer in **(Osegi & Enyindah, 2014)** with promising results. However, these techniques have the drawback of SMS delivery delay and compatibility problems with the internet due to diversity of SMS operator requirements/restrictions.

More recently, there has been an increasing call for implementing control and decision making systems with intelligent capabilities **(Vujovic & Maksimovic, 2015)**. Such systems provide the benefits of IoT technology and Artificial or Machine Intelligence (AM-I). However, there still remain a gap to be filled on the ability of such systems to continually make predictions on the possible outcomes or decisions that should follow through time and space.

In this paper, we propose the predictive Collaborative Web of Things (pCWoT) as a candidate prototype smart and IoT capable solution to this important requirement. We show through demonstration simulations how this prototype system can be applied in a real world scenario and set forth the direction for future designs of such systems.

3. STATEMENT OF PROBLEM

The current industry is faced with the challenge of defining adequate smart space models that can allow for a consensus to be arrived at given a number of separate decision making smart space sensor broadcast networks. For instance, if we consider the problem agreeing on the required number of taxis coming in and leaving a metropolis given a number of separate broadcast reports, a logical solution may be to compute the average of these broadcasts. As another example, consider a group of weather monitoring/broadcast stations specifically required to report the temperature of a given location say Zone A using their respective proprietary instruments; it is also required that these reports be continually made available on the internet. The problem may be cast as follows:

How can one predictively arrive at a temperature consensus given the different temperature readings?

Another requirement is the need for web based solutions rather than just internet solutions as the devices may operate through an internet.

4. RESEARCH OBJECTIVE

It is the objective of this research to describe a model-based approach that seeks to provide an answer to the aforementioned problem in the previous section. Stated in clear terms, we seek to develop an effective predictive systems model of a collaborative web of things network (pCWoT) for temperature monitoring and control in the smart space and in addition evaluate its performance using several performance metrics.

5. METHODOLOGY

5.1 Systems model of the Predictive Collaborative Web of Things Network (pCWoT)

A pCWT network is as shown in the flow diagram of Fig.1. It consists primarily of several Broadcast Data Units (BDU), Predictor Processor Units (PPU), and a Monitoring and Control Unit (MCU). The first two main processes are handled by a neural processing algorithm in the cloud while the last process is performed by an embedded microcomputer (actually a microcontroller with detailed circuitry).

Monitoring and decision making is taking care of by the embedded micro-computer after correctly decoding the predicted signals from a HTML/XML service with the neural algorithm as back-end.

This idea of using the neural approach is motivated by the need of a machine intelligent systems solution with inherent predictive capabilities for remote collaborative processing; the approach used here is adapted from the original research earlier carried out in **(Osegi & Anireh, 2016)** using a novel neural predictive framework based on an Auditory Machine Intelligence (AMI) algorithm and inspired by neuroscience and biology – this algorithm was initially referred to as the Deviant Learning Algorithm (DLA). However, in the case of this research, we use a simpler neural network computing architectural concept that harnesses the potential of state transitions **(Wacogne et al., 2012)**; this concept is briefly described in a succinct way and manner in the Appendix for readers who may wish to reproduce some of the results obtained here or apply the technique to other domain of applications.

5.1.1 Web Server Architecture

As a cloud computing model, p-CWoT needs to communicate to hardware remotely via a web server. Fig.2 describes architecture of this concept. In this illustration, a mobile Internet Service Provider (ISP) services internet requests to a host server which coordinates a PHP MyAdmin database and an Apache web service. A HTML front-end allows for end user interactivity while an XML layer adds an adaptive coding interface to prediction signals generated by the AMI algorithm in the cloud and eventually stored in control data storage for real time access using a p-CWoT capable device.

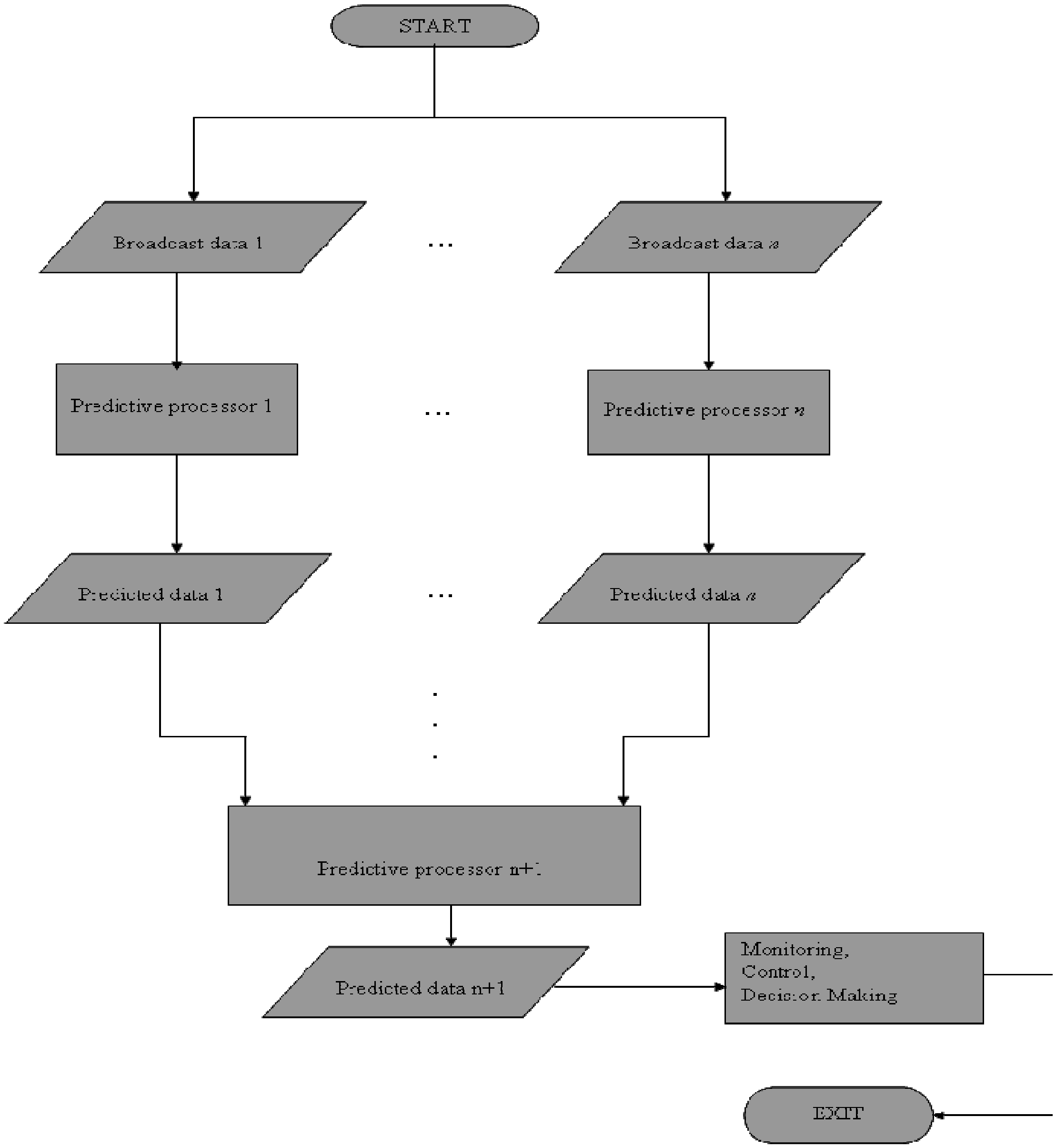

Fig.1: A Predictive Collaborative Web of Things Network (pCWoT)

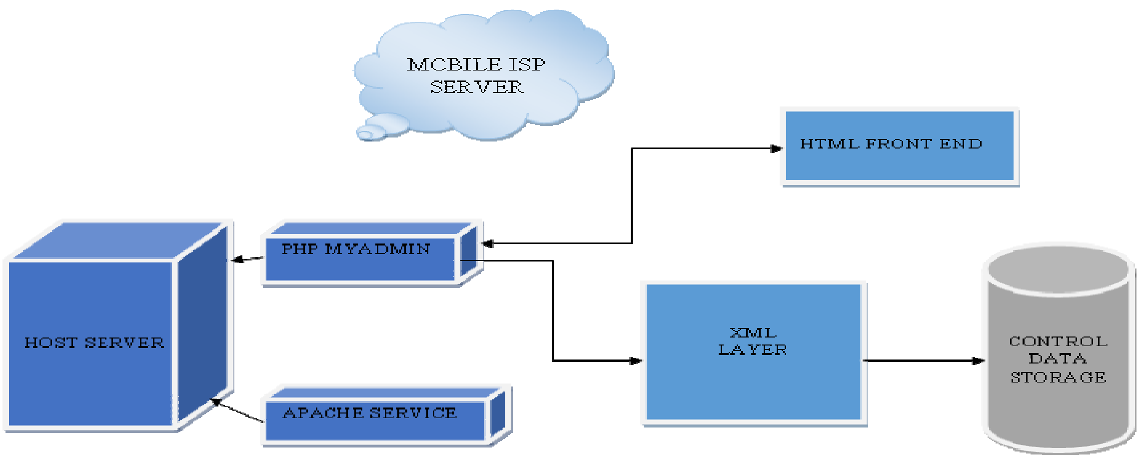

Fig. 2: Web-server Architecture of the pCWoT framework

5.2 Activity-Performance Criteria for Mobility

The pCWT uses a set of activity-performance criteria to evaluate whether or not it does desired function on time and with minimal failure mode due to lost signals. This subsection presents some of the key performance measures used in this research study. In developing this formal set of criteria, we borrow some of the models used from the study of Cognitive Radio Networks (CRNs) as earlier presented in (Esmaeelzadeh et al., 2013). We define our prototype pCWoT device as a sensor node that can participate in data communication and signal processing activities

5.2.1 Node Activity

To model sensor node activity, we use the Markovian death/birth ON/OFF process to determine the level of connectedness to a Web of Things (WoT) service as in Eq.1:

$$N_{active} = \frac{r_{con}}{r_{con} + r_{discon}} \qquad (1)$$

where,

r_{con} = connection frequency of the Sensor Node

r_{discon} = disconnection frequency of the Sensor Node

5.2.2 Sensing Accuracy Activity

Here we follow a somewhat different approach from that used in (Esmaeelzadeh et al., 2013) that is easier to interpret. We define the sensing accuracy as the number of correctly decoded control (prediction) signals obtained from a sample of transmitted signals for a given transmission duration.

$$S_{accuracy} = \left(\frac{n_{correct}}{n_{correct} + n_{incorrect}} \right) * 100 \qquad (2)$$

where,

$n_{correct}$ = number of correctly decoded prediction signals

$n_{incorrect}$ = number of incorrectly decoded prediction signals

5.2.3 Throughput Performance

The throughput defines a data bit-time ratio. Here we define the throughput as the number of correctly decoded prediction signals that actually terminate successfully at the sensor node:

$$S_{throughput} = \left(\frac{n_{correct}}{t_{last} + t_{first}} \right) \tag{3}$$

where,

t_{first} = receive time of the first correctly decoded prediction signal at the sensor node.

t_{last} = receive time of the last correctly decoded prediction signal at the sensor node.

5.2.4 Data packet delay

This describes how long it takes the correctly decoded prediction signals to terminate successfully at the sensor node:

$$t_{delay} = \left(\frac{\sum_{i=1}^{N_p} t_i^r}{N_p} \right) \tag{4}$$

where,

t_i^r = receive time of the i[th] correctly decoded prediction signal at the sensor node.

N_p = $n_{correct}$.

5.3 Physical Computing Control Model of the Predictive Collaborative Web of Things (pCWT) device

A prototype physical computing model concept of the proposed pCWT network device comprising primarily of an embedded microcomputer (uC1), a communication device (CC1), and an actuator (A1) is as shown in Fig.3. The embedded microcomputer is based on the Arduino microcomputer which uses the Atmega328P microcontroller device. The job of the embedded microcomputer is to coordinate the activities of the communication device and the actuator based on some already predefined sequence of coded program instructions (firmware). The communication device uses a GSM shield with the major interface lines (in blue) connected as shown. The Arduino microcomputer sends commands (control signals) to the servo motor which in turn can be used to operate (start/or stop) a process. A smart communication architecture and microcomputer firmware flow diagram for smart communication with a pCWoT device system is as illustrated in Fig.4 and 5 respectively.

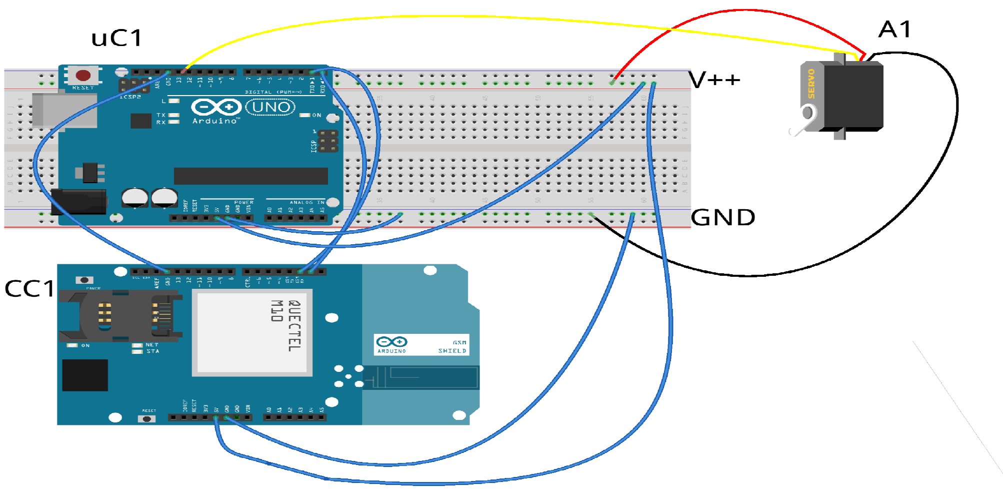

Fig. 3: A prototype physical computing model concept of a pCWoT device

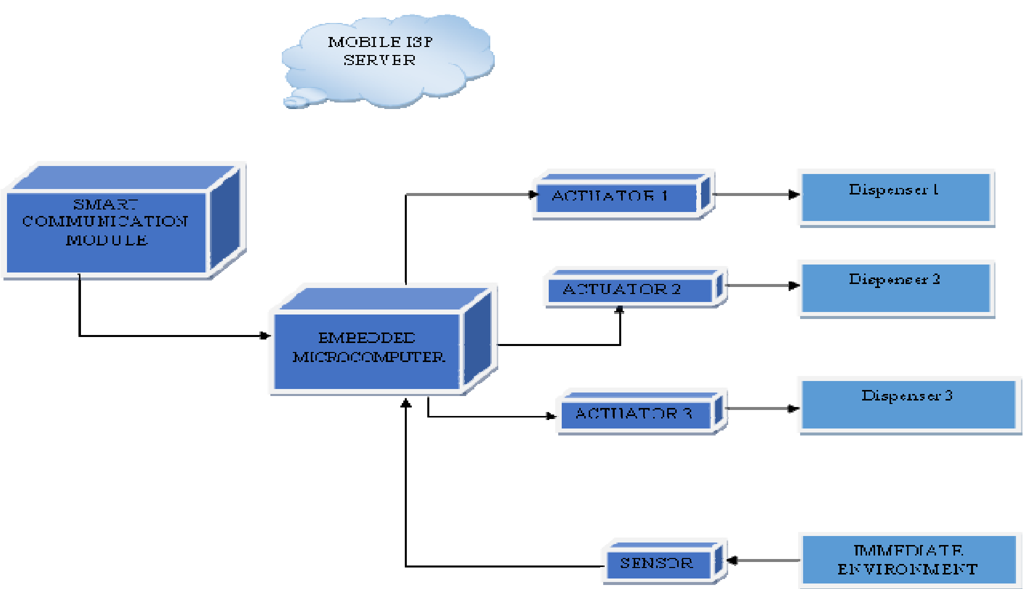

Fig. 4: Smart communication architecture of a pCWoT device system

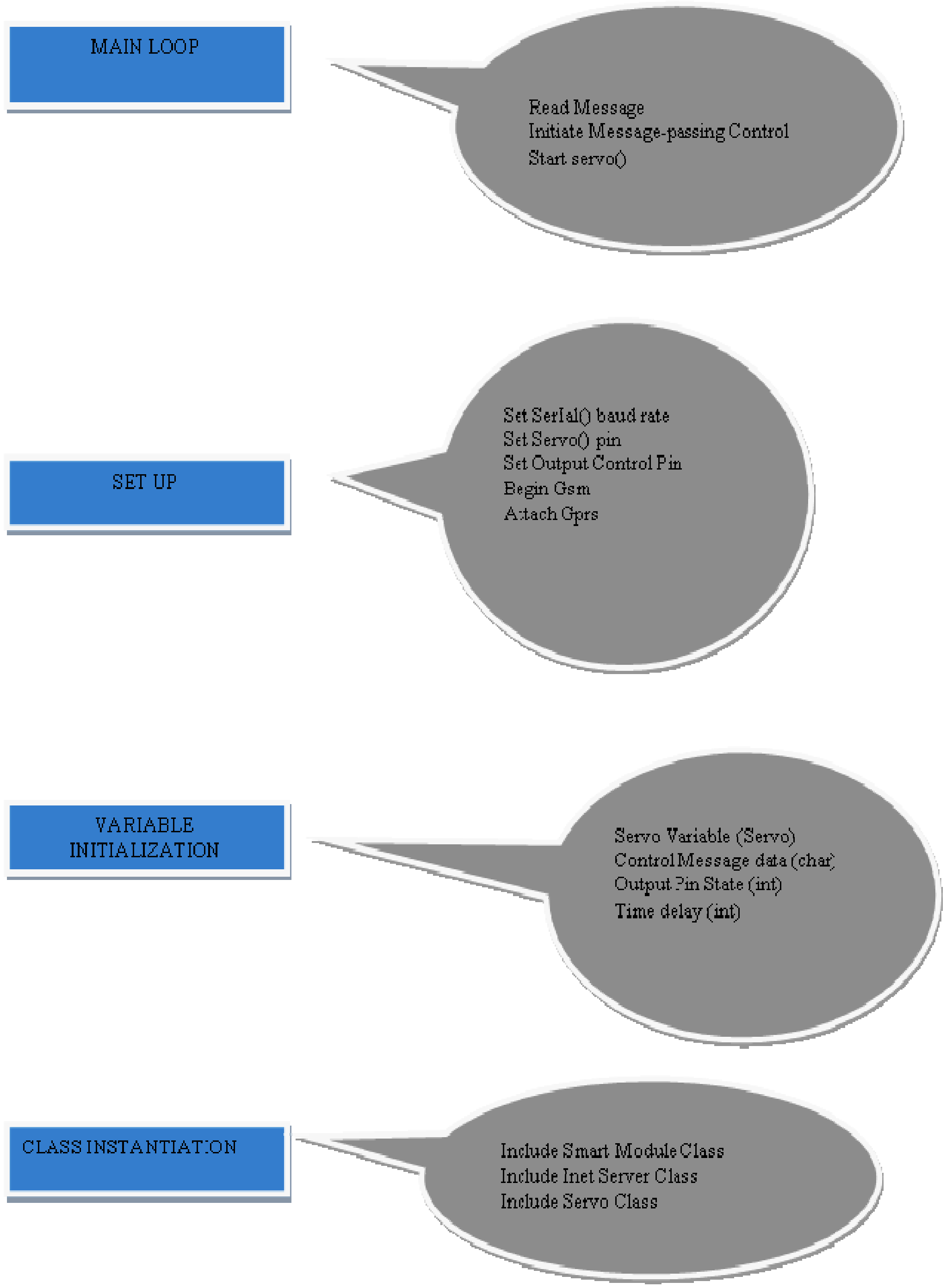

Fig.5: Smart communication firmware of a pCWoT device system

6. EXPERIMENTAL SIMULATION DETAILS, RESULTS

The simulated experiments are conducted using the concept of a real-time prototype predictive Collaborative Web of Things (pcOWT) systems model developed in the earlier sections. Initial experiments are demonstrated using open source tools such as Arduino, Processing and Wiring **(Banzi, 2009; Margolis & Weldin, 2011; Barragan, 2004; Severance, 2011)** and a prediction tool for Auditory Machine Intelligence (AMI) built in the PHP languagge. A sample data is presented in Table1. These data are a subset of temperature readings taken at 10min intervals; these readings may be conducted for the first day of the week and from 9:.00a.m to 12:00noon. Embedded code developed in Arduino C++ using the Ardino IDE and real time results using the aforementioned metrics are presented as a supplementary material; data simulations are presented here for comprehension.

Table1. Data for analysis

s/n	Weather Station 1	Weather Station 2	Weather Station 3
1	25	26	29
2	25	26	29
3	27	27	26
4	25	26	29
5	25	26	29
6	25	26	29
7	25	26	29

6.1 Task

We revisit the task of controlling a water chiller/heater system earlier described in **(Osegi et al., 2017)**. The goal is to continually predict the single temperature from 3 hypothetical weather stations and use the predicted temperature for controlling the operation of the steering arm of an over-temperature electromechanical device implemented as a servo.

6.2 Data Simulation Results and Discussions

Simulations using the sample data are as shown in Fig1-3 corresponding to weather station 1-3. The simulation plots are performed with a forecast horizon set to 4 and a back-step parameter set to 0. The first plot (Fig.1) is the weather station 1 time series data for training the AMI algorithm and the last two the deviant continual and look-ahead predictions respectively. From the last two plots, it is obvious that there will probably be an increase in the prediction temperature coming from weather station 1. The second and third plots (Figures 2 and 3) are the results for weather station 2 and 3 respectively. These results are similar to that obtained using the first dataset, however, there is a slight increase in prediction values which interestingly is indicative of the increases in the original time series values.

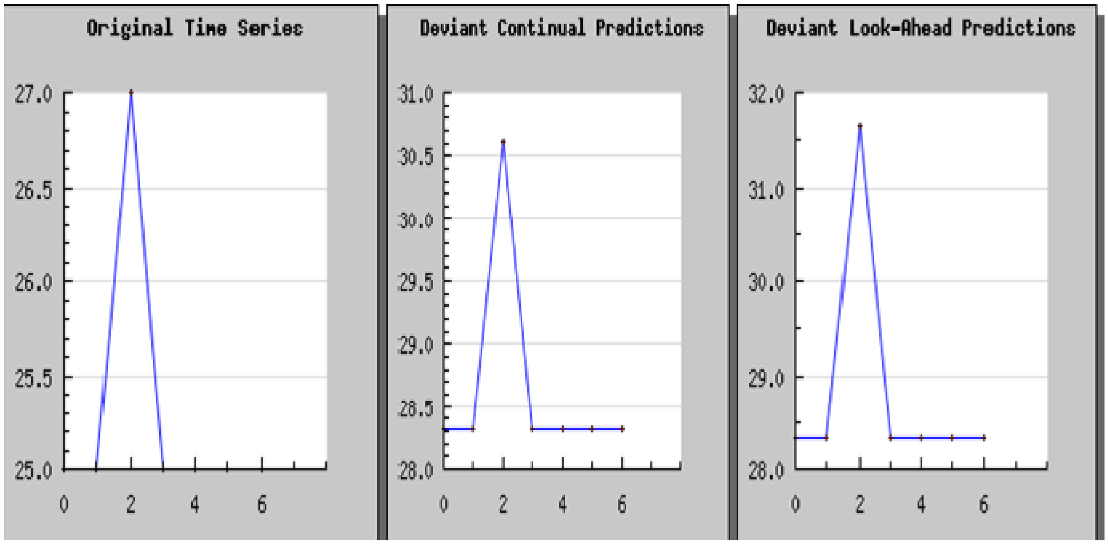

Fig.1: The AMI algorithm continual and look-ahead predictions for Weather station 1 sample data

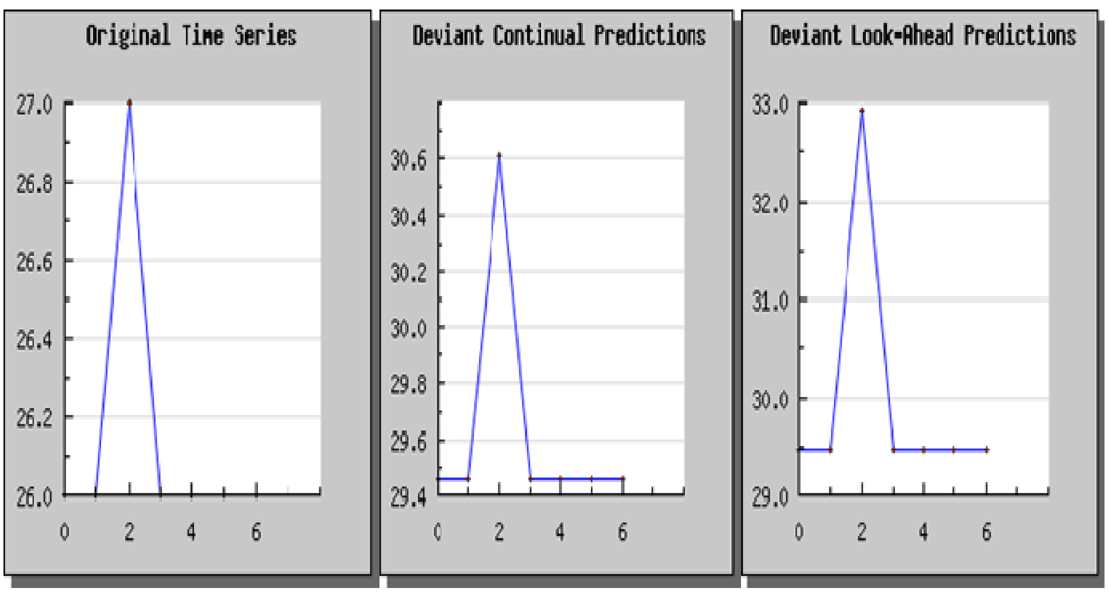

Fig.2: The AMI algorithm continual and look-ahead predictions for Weather station 2 sample data

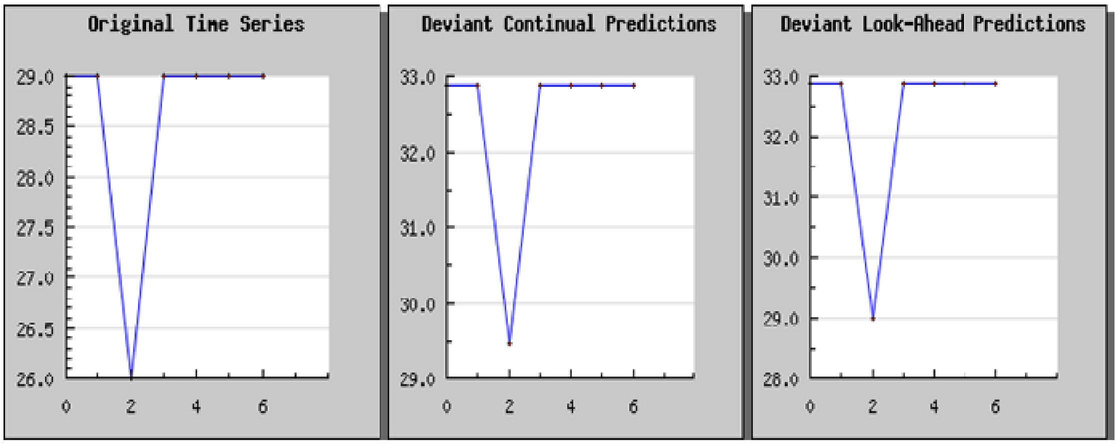

Fig. 3: The AMI algorithm continual and look-ahead predictions for Weather station 3 sample data

7. CONCLUSION

The Internet of Things (iOTs) present an obvious challenge to iOT researchers, industry experts and players alike one of which is the smart and intelligent generation and interpretation of agreeable set of results for control and decision making. We have presented an approach and pragmatic modelling tool called p-CWoT, in the context of a predictive and Collaborative Web of Things framework. Currently, this is work in progress and experiments are being conducted on real hardware. Some directions for furthering this work may include the integration of miniature swarm intelligence and evolutionary algorithms to assure a global optimization space and allow for a more robust intelligent decision making scheme. It is hoped that this ideas of ours can serve as an area for research and deep consideration by the academia and more importantly as a prototype model for industry players wishing to deploy the pCWOT technology in their area of operations.

8. CONTRIBUTIONS TO KNOWLEDGE

This research study has made the following original contributions:

- The development of a predictive Collaborative Web of Things (pCWoT) model for real-time control.
- The enhancement and validation of several performance metrics for evaluating the aforementioned.

APPENDIX: Mathematical Model of Auditory Machine Intelligence (AMI) algorithm (Phase-1 only)

Consider an initial input-output data set that may change through time as,
$S_n = \{S_1, S_2, \cdots, S_n\}$ for n-time steps of the observer.
We shall call this set the 'evoked potential' set or simply the EP; this set will contain sequences of standard signals and a deviant signal; the standards and deviant signal in turn gives a mismatch prediction accomplished by a deviant mismatch operation between both classes of signals.
Predictions in the AMI algorithm occur in two phases:
Phase-1 – low level pre-prediction phase
Phase-2 – high level post-prediction phase.
In this document, we mathematically analyse only the first phase of this algorithm.
Prior to our analysis, we make the following assumptions:

- Real world observations are numeric univariate data entities (or data elements) and are referred to here as the data sequence; this sequence belongs to the EP set.
- In the context of continual data sequencing, a set of previous (past) data elements are called standards while the current data element is referred to as a deviant
- All data elements are sequentially observed and analysed by a user hand-coded deviant mathematization program.
- In the future, and at each time step of the observation, a deviant program operation gives a sequence of future (predicted) mismatch states; an EP counter is incremented at each stage (time step, t) of the prediction until the desired number of time-stepped iterations is met.

Next, we describe the phases and operations in Phase-1 that can be performed by an AMI program.

Phase-1 or low level (pre-prediction) phase:

As before, let an EP be represented by the model expression:

$$S_{ep} = \begin{cases} \{S_1, S_2, \cdots, S_n\}, \\ \qquad S_{ep} \subset S_{ep+1} \end{cases} \tag{A.1}$$

where,

n = total number of sequences.

The input standards are usually sparse but are not necessarily constrained to be so; the sparse operation follows the overlap technique earlier proposed in Ahmad & Hawkins (2015); however the standards obtained by overlap may be fine-tuned for amplification giving a sparse real number set instead of sparse binary set as follows:

$$S^*_{ep(i)} = \mathrm{N}_{(j,i)} * \mathrm{A}_f \qquad \begin{aligned} i &= 1,2,3,\cdots,n \\ j &= 1,2,3,\cdots,c \end{aligned} \tag{A.2}$$

where,

N = uniform random number with values that fall between 0 and 1
A_f = the amplification factor
c = number of cells (cell mini-columns)

We then select the best input SDR for the DLA mathematization stage i.e. we obtain the pattern that maximizes the overlapping mini-columns using a roulette wheel operation as:

$$S_n^* = S_{ep(c_{best})}^* \quad \text{(A.3)}$$

where,

$$c_{best} = c, \quad if : o_{\max} == o_{\varsigma(c)} \quad \text{(A.4)}$$

$$o_{\varsigma(c)} = \sum S_{ep(c)}^* \quad \text{(A.5)}$$

$$o_{\max} = \max\left(o_{\varsigma(c)}\right) \quad \text{(A.6)}$$

The deviations of the standard from the deviant in the overlapping EP set (EP') for each mini-column may be computed as:

$$S_{dev} = \left\| S_{n-1}^* - S_{stars} \right\| \quad \text{(A.7)}$$

where the deviant is defined as:

$$S_{deviant} = S_{n-1}^* \quad \text{(A.8)}$$

and the standards are expressed as:

$$S_{stars} = S_{n-2}^* \quad \text{(A.9)}$$

The mean deviation for the EP' is computed from Eq.7 and Eq.8 as:

$$S_{dev(mean)} = \frac{\left(\left(\frac{\sum [S_{dev}]}{(n-1)}\right) + S_{deviant}\right) - 2}{n+1} \quad \text{(A.10)}$$

A Phase-1 deviant prediction is then given by:

$$S_{pred} = S_{n-1}^* + S_{dev(mean)} \quad \text{(A.11)}$$

The expression given in Eq.A.11 shows that there will obviously be an expected change in sequence prediction length through time; this is validated by the Model Adjustment Theory (Lieder et al, 2013).

REFERENCES

1. Ambrož, M. (2017). Raspberry Pi as a low-cost data acquisition system for human powered vehicles. *Measurement*, *100*, 7-18.
2. Barragán, H. (2004). Wiring: Prototyping physical interaction design. *Interaction Design Institute, Ivrea, Italy*.
3. Jacobsson, A., Boldt, M., & Carlsson, B. (2016). A risk analysis of a smart home automation system. *Future Generation Computer Systems*, *56*, 719-733.
4. Esmaeelzadeh, V., Berangi, R., Sebt, S. M., Hosseini, E. S., & Parsinia, M. (2013). CogNS: a simulation framework for cognitive radio networks. *Wireless personal communications*, *72*(4), 2849-2865.
5. Lieder, F., Daunizeau, J., Garrido, M. I., Friston, K. J., & Stephan, K. E. (2013). Modelling trial-by-trial changes in the mismatch negativity. *PLoS computational biology*, *9*(2), e1002911.
6. Martín-Garín, A., Millán-García, J. A., Baïri, A., Millán-Medel, J., & Sala-Lizarraga, J. M. (2018). Environmental monitoring system based on an Open Source Platform and the Internet of Things for a building energy retrofit. *Automation in Construction*, *87*, 201-214.
7. Massimo, B. (2011). Getting started with Arduino. *Make: Books*.
8. Margolis, M. (2011). *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*. " O'Reilly Media, Inc.".
9. Onukwugha, C. G., & Asagba, P. O. (2013). Remote control of home appliances using mobile phone: A polymorphous based system. *African Journal of Computing & ICT*, *6*(5), 81-90.
10. Onukwugha, C. G., & Osuagwu, O. E. (2012). Critical analysis of mobile devices based end-User programming of smart-Spaces. *Journal of Emerging Trends in Computing and Information Sciences*, *3*(7).
11. Osegi, N. E., & Enyindah, P. (2015). GOEmbed: A Smart SMS-SQL Database Management System for Low-Cost Microcontrollers. *African Journal of Computing & ICT*, *8*(2).
12. Osegi, E. N., & Anireh, V. I. (2016). Deviant Learning Algorithm: Learning Sparse Mismatch Representations through Time and Space. *arXiv preprint arXiv:1609.01459*.
13. Osegi, E. N., Wokoma, B. A., & Bruce-Allison, S. A. (2017, July). Leveraging Entrepreneurship through the design of Artificial Intelligence Projects. In *1st International Engineering Conference Port-Harcourt Polytechnic School of Engineering Technology. Theme: Entrepreneurship innovation through engineering technology for national development, 2017*.
14. Severance, C. (2014). Massimo banzi: Building arduino. *Computer*, *47*(1), 11-12.
15. Tomiyama, T., & Moyen, F. (2018). Resilient architecture for cyber-physical production systems. *CIRP Annals*.
16. Vujović, V., & Maksimović, M. (2015). Raspberry Pi as a Sensor Web node for home automation. *Computers & Electrical Engineering*, *44*, 153-171.
17. Wacongne, C., Changeux, J. P., & Dehaene, S. (2012). A neuronal model of predictive coding accounting for the mismatch negativity. *Journal of Neuroscience*, *32*(11), 3665-3678.