International Journal of Wireless Communications and Mobile Computing
2016; 4(2): 18-24
http://www.sciencepublishinggroup.com/j/wcmc
doi: 10.11648/j.wcmc.20160402.12
ISSN: 2330-1007 (Print); ISSN: 2330-1015 (Online)

Identification and Classification of Processing Unit Eligibility for Ubiquitous Computing Using Feature Selection Mechanism and Artificial Neural Network

Patience Spencer[1], Enoch O. Nwachukwu[1, 2]

[1]Department of Computer Science, Ignatius Ajuru University of Education, Rumuolumeni, Rivers State, Nigeria

[2]Department of Computer Science, University of Port Harcourt, Rivers State, Nigeria

Email address:

patsyspency2013@hotmail.co.uk (P. Spencer), enoch.nwachukwu@uniport.edu.ng (E. O. Nwachukwu)

To cite this article:

Patience Spencer, Enoch O. Nwachukwu. Identification and Classification of Processing Unit Eligibility for Ubiquitous Computing Using Feature Selection Mechanism and Artificial Neural Network. *International Journal of Wireless Communications and Mobile Computing.* Vol. 4, No. 2, 2016, pp. 18-24. doi: 10.11648/j.wcmc.20160402.12

Received: February 25, 2016; **Accepted**: March 11, 2016; **Published**: March 28, 2016

Abstract: Ubiquitous Computing is a trending innovation that allows a user to have access to many computers in a transparent manner anytime anywhere thereby enhancing computing confidence. However, the full potential of ubiquitous computing is not yet realised due to challenges including changing location of mobile users, poor network infrastructure, limited system resources, and poor transaction processing model. This work is concerned with the development of a proactive support for active transaction coordination in ubiquitous computing environment. The specific objectives are to identify relevant values of predefined key features of processing units that greatly impact on ubiquitous computing and to predict the processing capability of processing units using relevant values of the predefined features. An object-oriented analysis and system design methodology is employed and the proposed processing unit eligibility identification mechanism and neural network-based classifier is shown to effectively support ubiquitous computing.

Keywords: Ubiquitous Computing, Transaction, Multi-Layer Perceptrons, Neural Network, Feature Selection

1. Introduction

The shift from Mainframe-based Computing paradigm to PC-based Computing paradigm and then to Ubiquitous Computing is indeed a welcome development. Ubiquitous computing allows a user to have access to many computers in a transparent manner anytime anywhere [1] thereby enhancing computing confidence. The full potential of ubiquitous computing has not been realised [2] due to challenges including changing location of mobile users, poor network infrastructure, limited system resources, and poor transaction processing model.

This work presents a proactive platform for the coordination of active transactions in such a way that transaction processing is not affected by any form of system or network challenge. For example, the negative effect of transaction processing systems having frequent disconnections from distributed databases while executing a transaction results in low throughput and response time in ubiquitous computing environment. This is not acceptable as it creates unreliable access to data sources and unnecessary process delays.

The specific objectives of this work are to identify relevant values of predefined key features of processing units that greatly impact on ubiquitous computing and to predict the processing capability of processing units using the relevant values of the selected system attributes. To achieve these objectives, an object-oriented analysis and design methodology have been employed. The objects [3] of the proposed system and their behaviours are modelled using a combination of feature selection mechanism and Artificial Neural Networks. This approach is shown to enable a proactive support platform for transaction management where the scheduling component of the transaction server is given accurate prediction of the most eligible ` processing unit to be scheduled for transaction execution. This in turn impacts on users whose operations depend on information systems.

1.1. Transaction Processing and Ubiquitous Computing

Basically, ubiquitous computing paradigm is concerned with the ability of a user with a mobile computing device (wearable and handheld) to be able to access information residing in different computers as though the information is in the user's computer [1]. Advancements made in wireless networking technology, portable computing devices, and sophisticated mobile software applications greatly contributed to the paradigm shift from PC-based computing to ubiquitous computing. Software applications developed to support ubiquitous computing do not only take advantage of the advancement made in the field to do complex task but also face the challenges that come with the technology [4]. The challenges are mainly related to mobility, interconnectivity, and context-awareness. To achieve effective ubiquitous computing, the physical environment must be influenced digitally through collaborative signal and information processing sensors and control devices installed everywhere in the environment [5]. These devices perceive our environment and automatically adapt to the environment resulting in easy access to data and services. It also takes advantage of Positioning Systems to determine current location of mobile users and also link to other information services. Contrary to traditional mobile computing, context-based (physical, Human, and IT state properties) transaction management components are influenced by active environmental factors dynamically.

A database user (That is, the physical user) sees a transaction as a single operation whereas a database management system sees a transaction as a collection of several operations that form a single logical unit of work" [6]. Each logical unit of work is referred to as a sub-transaction. Sub-transactions can be distributed to different processing units (database hosts). In a distributed database system, different processing units are connected to one another through a communication network infrastructure [5]. Distributed transaction processing and wireless network infrastructure form a backbone for ubiquitous computing. Transaction management for ubiquitous computing aims at providing mobile users with reliable services in a transparent way anytime anywhere [7]. However, the processing of a single logical unit of work (That is, a transaction) in Ubiquitous Computing Environment as earlier mentioned is faced with unbearable challenges resulting in unreliable access to data sources (that is distributed databases) by transaction processing components [8]. This simply implies that the series of events that are carried out on a single unit of work (such as switching on a fan, registering courses to be taken by a student, retrieving a student's CGPA, confirming stock level of an item in a refrigerator, and the likes) from anywhere and at anytime need a favourable environment to perform optimally.

1.2. Changing Location of a Mobile User and Ubiquitous Computing

In ubiquitous computing environment, the activities of a mobile user cause the user to move about and the user may want to access variety of remote homogenous or heterogeneous databases. To avoid unreliable or denied access to data sources, a transaction management model that supports learning, collaboration, and autonomy is required. Existing database management systems may not have shown these characteristics sufficiently enough to effectively support ubiquitous computing especially as it affects mobile and distributed transaction processing.

2. Materials and Methods

In the proposed work, feature identification is achieved through Perception Reasoning technique [9] whereas transaction protocols execution is achieved through intelligent control [9]. These are key Artificial Intelligence behaviours [9] and are of great concern to innovative researchers in the field of Machine Learning. The perception reasoning mechanism eliminates noisy, irrelevant and redundant data [10]. In so doing, only meaningful features or feature values are fed into the neural network thereby accelerating input data capturing and classification processes.

For example, should a mobile user request for Location Dependent Information, the proposed transaction processing model ensures that the request is processed to a logical conclusion without loss of connectivity to data source. A question that is addressed in this work is "How will transaction processing components predict the most eligible processing unit to be allocated a new transaction or a migrated transaction?" In answering this question, varying relevant values of selected attributes of processing units are identified and used as the classification parameters. Tables 1a, 1b, and 1c show the processing unit parameters used in defining processing units eligibility ratings. Only participating processing units in the ubiquitous computing environment are considered in the classification process.

Table 1a. *Predefined Network Signal Strength Rating.*

Processing Unit Rating	**1**	**2**	**3**	**4**	**5**
Network Signal Strength (%)	0 - 20	20 - 40	40 - 60	60 - 80	80 - 100

Table 1b. *Predefined Processor Speed Rating.*

Processing Unit Rating	**1**	**2**	**3**	**4**	**5**
Processor speed in GHz	0 - 0.6	0.6-1.2	1.2 - 1.8	1.8-2.4	2.4 - 3.0

Table 1c. *Predefined Available RAM Size Rating.*

Processing Unit Rating	**1**	**2**	**3**	**4**	**5**
Available Ram Size in GB	0 - 1.6	1.6 - 3.2	3.2 - 4.8	4.8-6.4	6.4 - 8.0

Information in tables 1a, 1b, and 1c describe the rating of possible input values of Network Signal Strength (NSS), Processor Speed (PS) and Available RAM Size (ARS) of a participating processing unit. The values of these parameters vary at run-time thereby changing the behaviour of the

classifier dynamically. However, the classifier is expected to predict the set of features that qualifies a participating processing unit to be the most eligible for online transaction execution within a specified time.

To ensure that the right processing unit is selected without human intervention in a transparent manner, Artificial Neural Network (ANN) approach is employed. ANN is also known as Neural Network and can be described as a computational network that is implemented on a digital computer resulting in Digital Signal Processing (DSP) with interconnected processing entities that mimic biological structures (nervous system) in processing information. It is an innovative and Powerful modelling tool that can be applied to develop predictive models [11]. One of the most attractive properties of ANNs that instigated our choice of approach is the possibility of ANNs to adapt their behaviour to the changing characteristics of the modelled system [12]. Another useful characteristic is the ability of ANNs to effectively manage inter-related outputs in a better way. For example, the eligibility of processing units in ubiquitous computing environment is a function of measurable state information about the processing units (NSS, ARS and PS). In this context, NSS, ARS and PS are interrelated. Different combinations of the values of these system features results in different classes that define the eligibility of a processing unit.

The use of neural network technology to optimize classification processes is not novel but overcoming the challenge of getting the right neural network model to optimally solve the classification problem is what really matters. In this work, a Muiti-Level Perceptron (MLP) Artificial Neural Network is used to systematically solve the classification problem. The optimization of the performance of the neural network is done by applying Cross Entropy-based Back Propagation Algorithm.

The processing capabilities are then used to define the eligibility levels of participating processing units. If the eligibility of a participating processing unit falls under the category with the highest processing capability, that processing unit is presented to the transaction processing model as the most eligible processing unit within a predefined period of time. With this, online migration of active transaction from a processing unit with compromised processing capability to a processing unit with better processing capability is made possible to ensure continues access to data source.

2.1. The Proposed Model

Using a combination of feature selection mechanism made up of collaborative agent and MLP neural network, relevant values of processing units attributes (NSS, ARS and PS) are collected and processed (classified) in order to be used as the basis for prediction of the eligibility level of processing units in the ubiquitous computing environment. Fig. 1 shows the feature selection and input data normalization stages of the best processing unit prediction neural network model whereas fig. 2 is used to represent the neural network-based intelligence model. In fig 1, dynamic values of selected features are collected and separated into five different classes based on the rating of the sensed values of selected system features from the feature space *N* (shown as context info 1, 2, 3…n in fig 1). In this work, the system features are also referred to as context or state information.

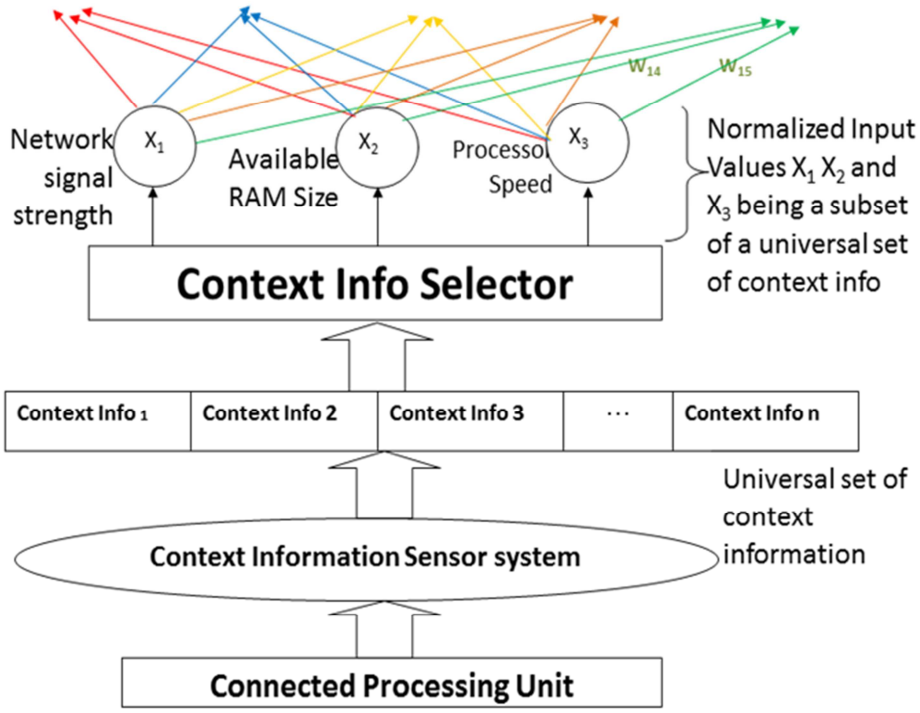

Figure 1. *Feature selection and input data normalization model.*

The "Context Info Selector" is designed to capture values of the predefined system features (That is, NSS, ARS and PS). The dynamic values of these features are then processed for the identification of relevant and significant values that are then fed into the neural network via the input layer of the neural network. The input nodes manage the non-classed vector dataset. Outputs of the input layer are then fed into the hidden layer of the network.

The formal presentation of the non-class vector (that is, a multi-dimensional input) is given by (1).

$$X = \{x_1 x_2 \ldots x_n\} \quad (1)$$

where: x represents the sensed input value and n represents the dimension of the input space. Applying it to the proposed model, $n = 3$. This means that the proposed model deals with a vector X of three input values of NSS, ARS and PS.

Every input value is first normalised to have an equivalent value that falls within the range of [0, 1]. Equation 2 is an adopted mathematical model used to determine the normalized input values:

$$v' = \frac{v - minA}{maxA - minA}(new\_maxA - new\_mind) + new\_minA \quad (2)$$

where:

- Attribute A corresponds to "Network Signal Strength" or Available RAM Size" or "Processor Speed".
- v is an element of all possible elements of A
- MinA is the minimum value of attribute A
- MaxA is the highest value of attribute A
- New_maxA is 1 and
- New_minA is 0.

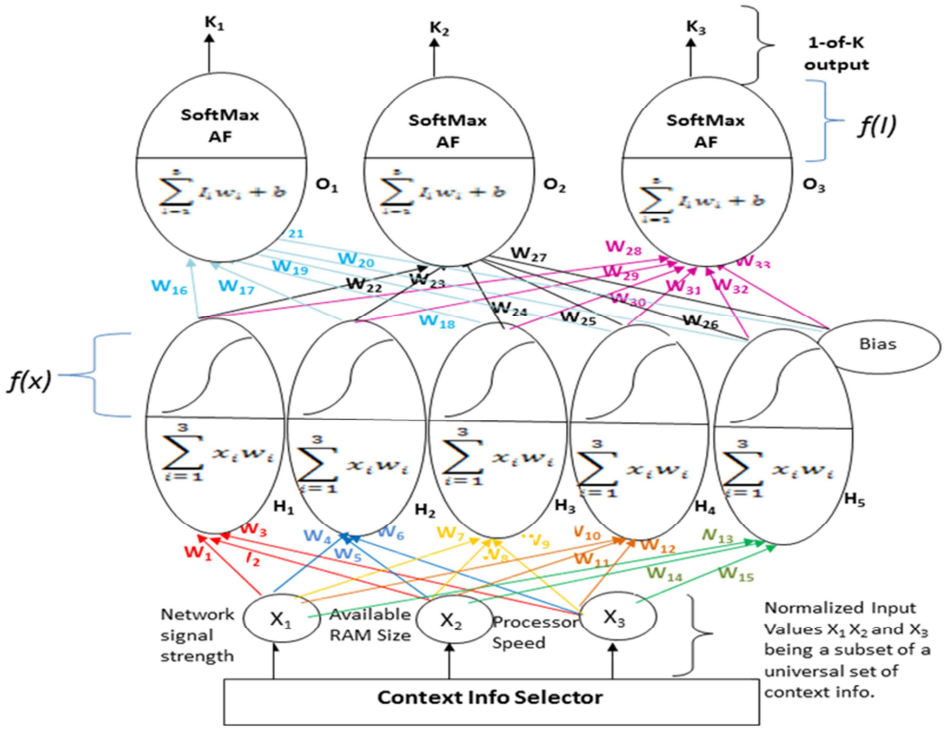

Figure 2. *A multi perceptron feed forward neural network influenced by Log-sigmoid and Softmax activation function used to build back-end mobile agent's intelligence in classifying processing units' eligibility.*

Each input value is sent to all the nodes in the hidden layer. Each node in the hidden layer is limited to a particular task using the logistic sigmoid function. The logistic sigmoid function decides whether an input (represented by a non-class vector of numbers I_i) belong to one class or another. For example, the first processing node in the hidden layer tagged H_1 is designed to identify input values (x_1, x_2, x_3) of the predefined features of participating processing unit under investigation. Expected output Vectors I_i of the nodes in the hidden layer representing five separate categories of perceived input values with rating from 1 to 5 where rating 5 represents the most significant value allowed and rating 1 representing the least significant value allowed. Nodes in the hidden layer are influenced by logistic Sigmoid Activation Function. The log-Sig activation function is a mathematical function used to identify the state of nodes in the hidden layer. Table 2 represents the expected states of nodes in the hidden layer H_1, H_2, H_3, H_4 and H_5 (presented in their un-normalized format).

Table 2. *Expected output vector data set at nodes H_1 to H_5 of the hidden layer (in their un-normalized format).*

Hidden layer Node Name	Dataset of expected values of predefined attributes		
	NSS	ARS	PS
H_1	[0, 20]	[0, 1.6]	[0, 0.6]
H_2	[20, 40]	[1.6, 3.2]	[0.6, 1.2]
H_3	[40, 60]	[3.2, 4.8]	[1.2, 1.8]
H_4	[60, 80]	[4.8, 6.4]	[1.8, 2.4]
H_5	[80, 100]	[6.4, 8.0]	[2.4, 30]

The connection strengths (that is, the value of weight that solves the given task optimally) of the inputs are determined arbitrarily and are normally small in value (lies between -∞ to +∞). The sum product of the input values and their associated connection strengths given by (3) are fed into the hidden layer.

$$I_i = \sum_{i=1}^{n} w_i x_i \quad (3)$$

where:

i is the node number, I is the net input of the i^{th} node, w_i is the i^{th} weight associated with the i^{th} node and x_i is the i^{th} input associated with the i^{th} node.

The Net Inputs from the hidden layer fed into the nodes of the output layer are given by (4), (5), and (6):

$$Z_1 = w_{16}I_1 + w_{17}I_2 + w_{18}I_3 + w_{19}I_4 + w_{20}I_5 + w_{21}b \quad (4)$$

$$Z_2 = w_{22}I_1 + w_{23}I_2 + w_{24}I_3 + w_{25}I_4 + w_{26}I_5 + w_{27}b \quad (5)$$

$$Z_3 = w_{28}I_1 + w_{29}I_2 + w_{30}I_3 + w_{31}I_4 + w_{32}I_5 + w_{33}b \quad (6)$$

It means that the net input from the input layer to the hidden layer is a function of x ($f(x)$) and the net input from the hidden layer to the output layer is a function I ($f(I)$).

The net inputs I_i plus their corresponding connection weights are passed through the logistic-sigmoid activation function in the processing nodes for category validation as each node represents a different category.

The hidden layer output plus their respective weighted bias b (usually within the range of 0 to 1) are in like manner fed into the output layer where only significant categories of the predefined feature values are classified using the Softmax activation function as the category validation tool. Note that the bias b is added to normalize the net input.

The three processing nodes in the output layer represent the three most significant sets of predefined feature values. These set of values are chosen because they give significant support to ubiquitous computing.

Table 3 shows the un-normalized expected states of the three most significant categories of processing unit K_1, K_2, and K_3 (that is, output). The normalised output of each output node (O_1, O_2, and O_3) rages from 0 to 1. Zero (0) represents the lowest (or no) probability of correct match and 1 represents the highest probability of correct match.

Table 3. *Expected classed vector datasets at node O_1 O_2, and O_3 output layer (in their un-normalized format).*

significant category name	Dataset of expected values of predefined attributes		
	NSS	**ARS**	**PS**
K_1	[40, 60]	[3.2, 4.8]	[1.2, 1.8]
K_2	[60, 80]	[4.8, 6.4]	[1.8, 2.4]
K_3	[80, 100]	[6.4, 8.0]	[2.4, 30]

Only one out of the three possible output nodes (see fig. 2) is expected to give a probability value of 1 at the end of each iteration. However, there is a high possibility that, more than one output node will fire (a situation where the value of calculated output is 1). When the state of more than one output node is 1, it means that the eligibility status of the processing unit under investigation is not certain (indicating erroneous prediction). When this happens, the neural network is subjected to take error correction training in order to give a non ambiguous prediction.

The state of each output node is determined by passing their corresponding net inputs through Softmax Activation function which is given by (7):

$$K_{i=\frac{1}{1+exp^{(-Z)}}} \quad (7)$$

where: $Z = \sum w_{ji} I_{ji} + b_i$

w_{ji} *is* the connection weight from the hidden *layer to the output layer*

I_{ji} is the vector of input values from the hidden layer to the output layer.

b_i is the added bias.

K_i represents a possible class of processing unit eligibility such that:

K_1 represents a dataset of values of NSS, ARS, and PS of the processing unit that qualifies it to be the least eligible processing unit at a particular point in time

K_2 represents a dataset of values of NSS, ARS, and PS of the processing unit that qualifies it to be the second to the most eligible processing unit at a particular point in time and

K_3 represents a dataset of values of NSS, ARS, and PS of the processing unit that qualifies it to be the most eligible processing unit at a particular point in time

Table 3 shows the accepted relevant and significant values of context information used to determine the eligibility level of processing units in the ubiquitous computing environment. The output of each output node K_i is validated using (8):

$$K_i = \begin{cases} 1 \; if \; Z \geq T \\ 0 \; if \; Z < T \end{cases} \quad (8)$$

where:

$Z = \sum_{i=1}^{n} w_i I_i + b_i$ (weighted sum of inputs or observations including the bias).

T is the predefined rule also known as threshold (used to aid the supervision of the node state validation process).

2.2. Training the Network

A training sample consists of the sensed input vector X and the expected output vector of a node denoted by Y [13]. A multi-input vector, and multi-output vector is represented as $\{X_i, Y_i\}$. If the calculated output vector K_i do not match the expected output vector Y_i, the network is then turned using Cross Entropy (CE) error correction function (where the values of the weights and biases are modified guided by a threshold value). The error correction algorithm starts from the output layer through the input layer (That is, Back propagating the error) for re-computation of the neural network with adjusted connection weights and possibly biases. This process continues until the weights or biases that solve the problem optimally is found.

In Implementing a back propagation with CE error function, the sum of the log to the base e of each calculated network

output K_i times their corresponding expected outputs Y_i is determine first, and then, the negative of the sum (that is, the result) is taken. For example, if the calculated output at node O_1 is a classed vector K_1 consisting of the elements NSS (%), ARS (GB) and PS (GHz) with values that fall within the expected ranges of the values of the elements at node O_1, then K_1 is said to match the expected output vector Y_1 at node O_1. The input and output vectors of the three output nodes of the proposed classifier neural network are formally represented in (9), (10), and (11). Now if the values of Ki are far from the values of their corresponding Yi, then, the state of the node is described as "0". However as the values of the elements gets closer to the expected values, the probability of getting the expected match gets higher. A probability of "1" indicates that K_i correctly matches Y_i. The CEs associated with K_1, K_2 and K_3 outputs are determined using the adopted formulas (15), (16) and (17). The expected outputs correspond to 0 (For no, there is no match) or 1 (For yes. there is a match).

$$K_1 = \{k_1, k_2, k_3\} \mid \{[40, 60], [3.2, 4.8], [1.2, 1.8]\} \quad (9)$$

Where, k_1 is the calculated value of NSS, k_2 is the calculated value of ARS and k_3 is the calculated value of PS.

$$Y_1 = \{y_1, y_2, y_3\} \mid \{[40, 60], [3.2, 4.8], [1.2, 1.8]\} \quad (10)$$

Where, y_1 is the expected value of NSS, y_2 is the expected value of ARS and y_3 is the expected value of PS.

In a similar manner, using the same state information, the outputs of node O_2 and O_3 are represented as:

Outputs at O_2

$$K_2 = \{k_4, k_5, k_6\} \mid \{[60, 80], [4.8, 6.4], [1.8, 2.4]\} \quad (11)$$

$$Y_2 = \{y_4, y_5, y_6\} \mid \{[60, 80], [4.8, 6.4], [1.8, 2.4]\} \quad (12)$$

Outputs at O_3

$$K_3 = \{k_7, k_8, k_9\} \mid \{[80, 100], [6.4, 8.0], [2.4, 3.0]\} \quad (13)$$

$$Y_3 = \{y_7, y_8, y_9\} \mid \{[40, 60], [3.2, 4.8], [1.2, 1.8]\} \quad (14)$$

The subscripts of the various vector elements are serially numbered for easy identification.

CE error at output node O_1 is given by equation 15:

$$E_{K1} = -((Y_1(\ln(k_1))) + (Y_2(\ln(k_2))) + (Y_3(\ln(k_3)))) \quad (15)$$

CEE at output node O_2 is given by equation 16:

$$E_{K2} = -((Y_4(\ln(k_4))) + (Y_5(\ln(k_5))) + (Y_6(\ln(k_6)))) \quad (16)$$

CEE at output node O_3 is given by equation 17:

$$E_{K3} = -((Y_7(\ln(k_7))) + (Y_8(\ln(k_8))) + (Y_9(\ln(k_9)))) \quad (17)$$

Mean Cross Entropy error

To determine the Mean Cross Entropy Error E_m for the three-element dataset described in section 2..2, the negative values of the node wise errors (That is, (15), (16), and (17)) are summed up and divided by 3 (since there are three classed vectors). Mathematically this is represented by (18).

$$E_m = (E_{K1} + E_{K2} + E_{K3}) \div 3 \quad (18)$$

The mean cross entropy error E_m is to be reduced to zero (0) or very close to zero for the performance of the neural network to be accepted. It is important to note that CE error function essentially ignores all computed outputs which do not correspond to an expected output of "1" [10]. It is used as error correction function in this work because it also serves as a stop condition for the neural network learning without exceeding the number of required training iterations [10]. This improves the net performance of the classifier. Now if a node's computed state is close to its expected state for all training inputs *x*, the error will be close to zero. For example, if the expected network state Y_1 is 0 and the calculated sate K_1 is 0 for input *I*. this means that the network is performing well on the input. In a similar way, if Y is 1 and K is very close to 1, the network is performing well. Hence, the processing cost (error) will be low in as much as K is close to Y. This implies that the error update that is propagated backwards from each output node is directly proportional to the difference between expected values of NSS, ARS and PS and the calculated values of NSS, ARS and PS.

Figure 3 is a representation of a transaction processing model showing the Transaction Server where the proposed (agent-based) feature selector and neural network-based processing unit classifier are implemented.

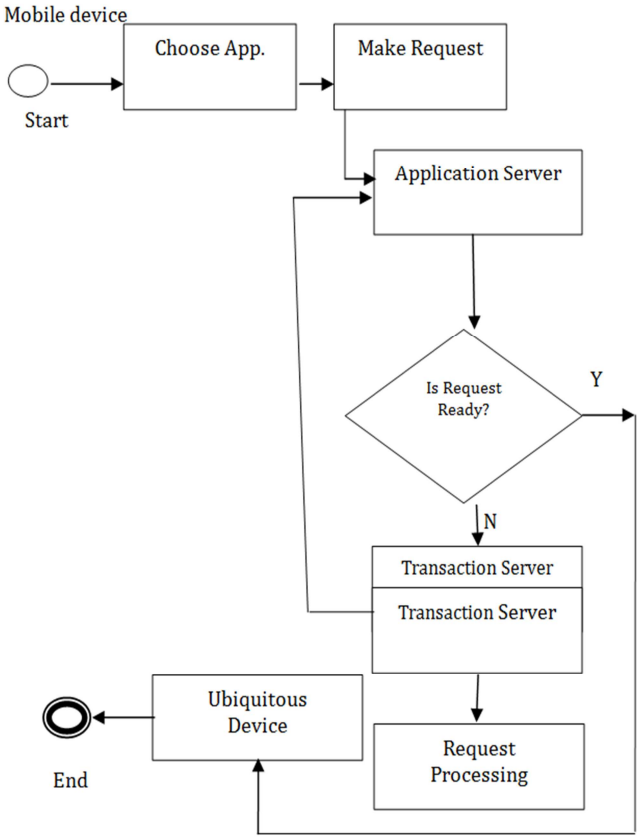

Figure 3. *A System Flow diagram showing the Transaction Server where the proposed Feature Identifier and Neural Network-based Processing Unit Classifier are implemented.*

3. Result and Discussion

For a timely and reliable performance, the proposed feature selector and neural network-based most eligible processing unit predictor is shown to intelligently create an effective platform for a scheduler in transaction processing systems for ubiquitous computing environment. The feature selection component of the proposed system transmits only predefined values of network signal strength, available RAM size, and processor speed to the multi-perceptron neural network where classification is based on processing units' processing capacity within a given period of time. It is very essential to allocate new or migrated transaction to processing units with the required processing capacity to avoid frequent disconnection from data sources and unnecessary delays resulting from network and system resource related issues. Allocating transaction to the most eligible processing unit is shown to provide uninterrupted access to data sources. This in turn enhances Response Time and Transaction Throughput.

4. Conclusion

Allocating transaction to processing units with sufficient RAM size, high Processor Speed and good Network Signal Strength affects transaction processing in ubiquitous computing environment positively. These are key transaction processing system context that must not be over looked when designing a any transaction processing model for ubiquitous computing environment.

The implementation of transaction processing models with no context-awareness [14] or poor context management skills and possible failure prediction mechanisms is unacceptable as it creates room for undesirable challenges during the execution of distributed transactions. To solve this problem, this work presents a transaction processing model with a proactive transaction server.

Using an object oriented analysis and design methodology [9], this work showed how a combination of feature selection mechanism and Artificial Neural Network are used to create an effective platform for a proactive transaction management at the backend of a transaction processing model. Implementing a back propagation with cross-entropy error function when training the proposed MLP neural network classifier gives a transaction server the context awareness strength it needs to make accurate prediction of the most eligible processing unit to be allocated a transaction at any point in time.

References

[1] Nwachukwu, E. O. (2010). *Information Technology: The Albatross of Our Time: an Inaugural Lecture*. University of Port Harcourt.

[2] Byeong-Ho, K. A. N. G. (2007). Ubiquitous computing environment threats and defensive measures. *Int. J. Multimedia Ubiquit. Eng*, *2*(1), 47-60.

[3] Filip, M. J., Karunungan, K. L., Kramer, J. C., Lee, L. C., Moore, D. L., Shih, C. C., and Sydir, J. J. (1995). *U.S. Patent No. 5,414,812*. Washington, DC: U.S. Patent and Trademark Office.

[4] Puder, A., Römer, K., and Pilhofer, F. (2006). *Distributed systems architecture: a middleware approach*. Elsevier, UK.

[5] Chong, C. Y., and Kumar, S. P. (2003). Sensor networks: evolution, opportunities, and Challenges. *Proceedings of the IEEE*, *91*(8), 1247-1256.

[6] Silberschatz, A., Korth, H. F., Sudarshan, S. (2002) *Database System Concepts* McGraw Hill, New York.

[7] Tang, F., and Li, M. (2012). Context- adaptive and energy-efficient mobile transaction management in pervasive environments. *The Journal of Supercomputing*, *60*(1), 62- 86.

[8] Silberschatz, A., Korth, H. F., and Sudarshan, S. (2006) *Database System Concepts* McGraw Hill, New York.

[9] Beetz, M., Buss, M., and Wollherr, D. (2007). Cognitive technical systems—what is the role of artificial intelligence? In *KI 2007: Advances in Artificial Intelligence* (pp. 19-42). Springer Berlin Heidelberg.

[10] Bengio, Y., and LeCun, Y. (2007). Scaling learning algorithms towards AI. *Large-scale kernel machines*, *34*(5).

[11] Rughani, A. I., Dumont, T. M., Lu, Z., Bongard, J., Horgan, M. A., Penar, P. L., and Tranmer, B. I. (2010). Use of an artificial neural network to predict head injury outcome: clinical article. *Journal of neurosurgery*, *113*(3), 585-590.

[12] Karlik, B., & Olgac, A. V. (2011). Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, *1*(4), 111-122.

[13] Huang, G. B., Wang, D. H., and Lan, Y. (2011). Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, *2*(2), 107-122.

[14] Schilit, B., Adams, N., & Want, R. (1994, December). Context-aware computing applications. In *Mobile Computing Systems and plications, 1994. WMCSA 1994. First Workshop on* (pp. 85-90). IEEE.