ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 5, Issue 3, pp: (53-60), Month: July - September 2017, Available at: www.researchpublish.com

A Robust Mechanism for Artificial Neural Network Context-Aware Recommender Systems (ANN CARS) in Mobile Environment

[1]Samuel King Opoku, [2]Dr. D. Subba Rao

[1]Computer Science Department, Kumasi Technical University, Ghana
[2]Professor, Computer Science, Centurion University, Indian

Abstract: **The quest to develop applications to fully assist humans in our daily activities through the acquisition and utilization of contextual information has been the ultimate concern of researchers in pervasive computing. Many different implementation mechanisms have been adopted over the years. Currently, researchers are focusing on artificial neural network to develop context-aware recommender systems. This paper examines the various architectures and learning algorithms employed in these systems in order to deduce the general trend of implementation whilst assessing their pros and cons through literature survey. The paper goes ahead to recommend robust mechanism for future applications that can be used in mobile environment. The work revealed that the general trend of implementation is through threshold scoring mechanisms, reliance on the internet, complicated learning algorithms and architectures with the view to achieving higher prediction accuracy. These mechanisms ignore the limited capabilities of mobile devices in achieving pervasive computing. The work therefore recommended that there is the need to use simple but adaptive neural network architectures and learning algorithms so that these systems can be implemented on mobile devices without any negative impact on them**

Keywords: **Artificial Neural Network, Context-aware Application, Intelligent Systems, Mobile device, Pervasive Computing, Recommender Systems.**

I. INTRODUCTION

Context-aware applications have been the predominant domain of application development in our 21st century due to their ubiquitous existence and also having the ability to assist users. Context-aware applications use context of the user and environment to adapt themselves to suite users' tasks. According to [1], a context is "Any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application including the user and application themselves". This definition is also adopted in this work. Since the invention of context-aware computing, there have been many implementation mechanisms of context-aware applications from hardware sensors to software sensors and now intelligent decision making systems through Artificial Neural Network (ANN). The use of artificial neural network to design context-aware recommender systems was popularized in 2016. Researchers designed these applications as recommender systems that use contextual information and they are based on internet of things (IoT's). These systems analyse users' activities and recommend things that users may be interested in such as shopping. Different techniques have been used to train the ANN used in these recommender systems. This paper examines the various techniques of employing ANN in implementing context-aware recommender systems (CARS). It assesses the strengths and limitations of the techniques currently employed. It ends with a recommendation to help researchers produce robust systems in the future. This paper is divided into six sections. Section I is the introduction which sets the background and the objectives of the work. Section II focuses on the overview of ANN. Section III designs a methodology for this work. Section IV looks at the current techniques employed in ANN CARS implementation. Section V talks about the need for robust mechanism by looking at the limitations of the techniques used in the preceding section. The last section discusses the robust mechanism for future research

II. OVERVIEW OF ARTIFICIAL NEURAL NETWORK (ANN)

ANN is a computational model that tries to mimic the structure and functionalities of biological neural network. Neurons are connected to one another by connection link. Each link is associated with weights which contains information about the input signal. The weights are adjusted according to the magnitude of the error in a way defined by the learning algorithm. The data used to train the network can be normalized so that the desired results can be achieved. In a neural network, the input data is always known. The output data is sometimes known. ANN solves problems by adjusting its weights through learning. The output is usually of linear behaviour and thus an activation function is required to make the output sometimes non-linear. Typical neuron architecture is illustrated in Figure 1

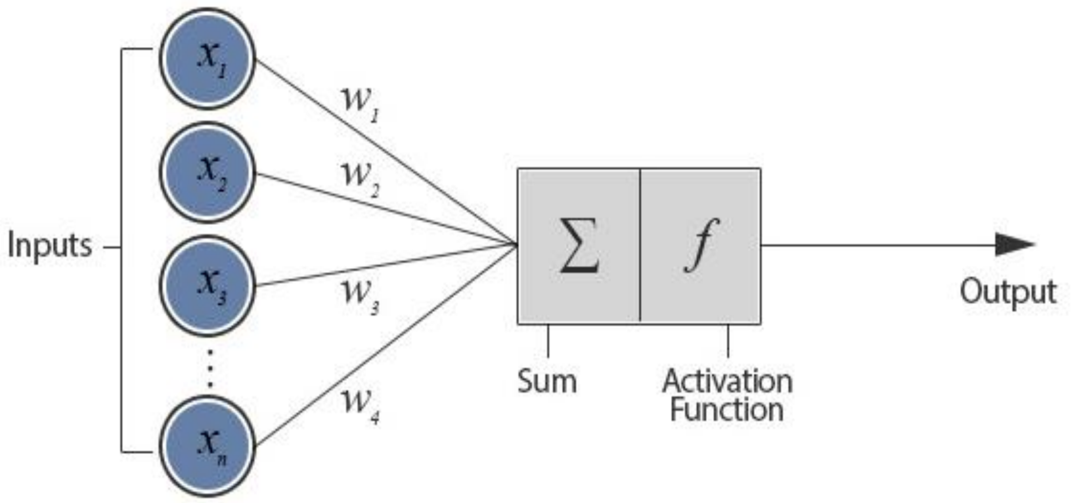

Fig 1: Typical Architecture of Artificial Neuron

From figure 1, the input values are X_i. Each input value has a weight associated with it indicated by Wi for specific i. The summation of the product of Xi and Wi gives the linear behaviour of the neuron. An activation function, f, is applied on the summed value to give the final output. The size of the training data that needs to be used to train the network should correspond to network size [2]. However, data set can be repeated when data set is limited [3] to achieve the desired results. An arbitrary input neuron called bias which has a magnitude of 1 is added to the input neurons and also hidden neurons to increase the flexibility of the model to fit the data in the data space [4], [5].

Ref [6] started the field of neuro-computing by demonstrating, in principle, that simple types of neural networks could compute any arithmetic or logical function. The simplicity of their work was achieved by allowing only binary states (0 and 1) for their neuron which operated under discrete time assumption. The limitation of their work was that weights and the neuron's threshold were fixed in the model and no interaction among network neurons, thus making it non-flexible network. Many researchers from 1944 to 1960 examined the issues governing neuro-computing with the view to designing a human brain-like architecture. Notably is the work of Donald Hebb [7]. His work proposed that classical condition exist due to the properties of individual neurons. He proposed a mechanism for learning in biological neurons which is one of the first learning rules, called Hebb's rule, by providing association between input and output vectors.

In 1962, Rosenblatt succeeded with a perceptron convergence theorem [8] which relied on error-correction learning algorithm for a single-layer perceptron. The perceptron learned from its own weight values, thus fixing the limitation of [6]. Rosenblatt's work became the first practical artificial neural network. Widrow and Smith, contemporaries with Rosenblatt, developed a different type of neural network processing element called Adaptive Linear Neuron or later Adaptive Linear Element (ADALINE) which is also a single-layer neural network but unlike the perceptron, used a threshold logic device that performed a linear summation of input [9]. Unfortunately, the work of Minsky and Papert in [10] weakened the exuberance of researchers of neural network when they showed that a single-layer network can only handle linearly separable problems. However, many real-word problems are non-linear in nature. Though Rosenblatt, Widrow and Smith got to know the limitations of their work, they proposed a new network to handle the limitation but could not modify their learning algorithms to train their network. Research in neural network was relegated for about thirteen (13) years.

During the relegation period, some researchers worked on neural network as signal processors. Teuvo Kohonen [11] and James Anderson [12] independently developed new neural networks that could act as memories. Kohonen's work proposed correlation matrix model for associative memory. The model was trained using Hebb's rule to learn association between input and output vectors. The mathematical structure of the network was emphasized. Anderson's work proposed a "linear associator" model for associative memory. The model was also trained using generalization of the Hebb's rule to learn the association between input and output vectors. However, Anderson's work emphasized the physical aspect of the network. Stephen Grossberg [13] was very active investigating self-organising networks which was based on visual system consisting of short-term and long-term memory mechanisms. The work ended up with a new architecture called adaptive resonance theory (ART) network. These research efforts survived with inability to recognise certain patterns [14] since the input data should be orthogonal or uncorrelated [7].

During the 1980's, personal computers and workstations became more powerful which served as an ingredient for active neural network research. In 1982, John Hopfield [15] revived neural network publicly by designing an associative memory model through statistical mechanics from physicists' perceptive which was based on energy function. His model gave birth to recurrent neural network but however limited by the fact that there was no synchronization between elements making it practically impossible to handle real-world problems. In the same period, back-propagation algorithm for training multilayer perceptron networks was discovered independently by several different researchers. The algorithm was popularized by David Rumelhart and James McClelland [16]. This algorithm was the answer to the criticisms Minsky and Papert made in 1969 which strengthened the field of neural networks. Since the 1980's, thousands of papers have been written, neural networks have found countless applications and the field had been humming with new theoretical and practical works. Other research works focus on adaptation and modification of previous works to suite particular task. For instance, [17] trained special type of recurrent neural network to predict the next character in a sequence by predicting the probability distribution for the next character and then sample a character for the distribution. Notably, a single neuron has no usefulness in solving real-life problem [18] but a network of neurons arranged in layers is capable of solving real-life problem in a non-linear, distributed, parallel and local way [19].

III. METHODOLOGY OF THE WORK

Survey papers solely on ANN applications were ignored since the focus of the study was to look at ANN in context-aware applications but not general applications of ANN. Likewise survey papers solely on recommender systems were also ignored. There have been many mechanisms for implementing recommender systems and also many mechanisms for implementing context-aware applications. This paper looks at the use of artificial neural network in implementing recommender systems that takes into consideration the concepts of context-aware computing. The following procedures were employed in the study:

1. Identify when ANN CARS were popularized
2. Gather research papers from the year of popularization onwards
3. Arrange the papers in chronological order
4. For each paper, determine
   a. Architectural design and learning algorithm of ANN employed
   b. The strength of the methodology employed
   c. The gap(s) filled by the methodology and the gap(s) unhandled
5. Identify the trend of application development
6. Produce limitations of the techniques employed
7. From the limitations, produce recommendations for future research work

Figure 2: below illustrate the methodology using flowchart.

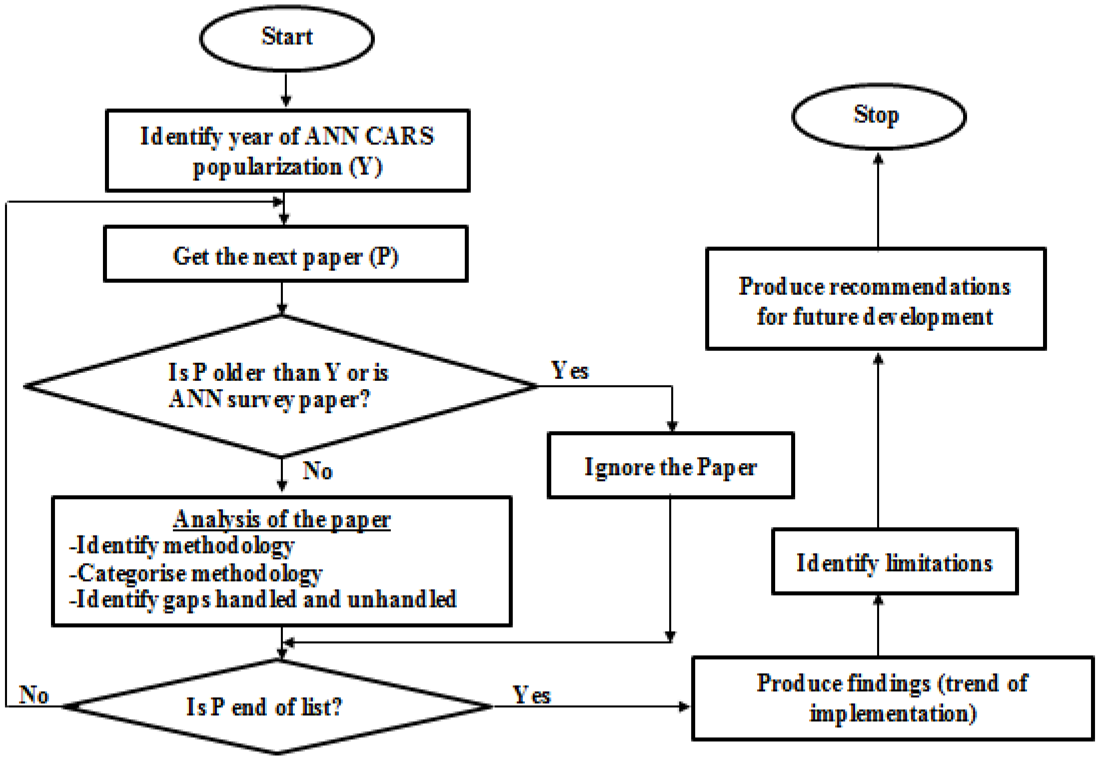

Fig 2: Overview of the Methodology

IV. CURRENT TECHNOLOGIES IN DEVELOPING ANN CARS

The two major types of learning algorithms had been employed in ANN CARS implementations. These are unsupervised and supervised learning algorithms. Unsupervised learning is used when, for a given input, the exact numerical output of a network is unknown. The network must organize itself in order to be able to associate clusters with units. In supervised learning, the output computed by the network is observed and the deviation from the expected answer, which is known in advance, is measured. The weights are adjusted according to the magnitude of the error in a way defined by the learning algorithm [19]. There also exist, though, another learning algorithm called Reinforcement learning in which an agent tries to learn by evaluating the feedback signal and adjusts itself accordingly. To start with unsupervised learning implementations, [20] tracked users' activities using probabilities of applications that follow Zipf's law and log-normal distribution for autonomous classification. Another mechanism under unsupervised learning which has been used has to do with deep learning algorithm. Ref [21], [22] and [23] proposed a latent factorization mechanism based on deep learning technique on traditional artificial neural network to boost collaborative filtering for cross domain recommender system.

Regarding supervised learning, many algorithms have been developed. Some of them are just variations of basic ones. The common approach depends on error correction mechanisms in which the error is propagated backwards. The different implementation techniques under the said mechanism are Gradient Descent, Newton's method, Conjugate gradient, Quasi-Newton and Levenberg Marquardt [24]:

- **Gradient Descent** method is also called steepest descent. It is the simplest training algorithm which requires information from the gradient vector and hence it is a first order method. The severe drawback is that it requires so many iterations. It is recommended algorithm when we have very big neural networks since it stores the gradient vector (size n) and it does not store the Hessian matrix (size n^2).
- **Newton's Method** is a second order algorithm because it makes use of the Hessian matrix. Its objective is to find better training directions by using the second derivatives of the loss function. However, this method has a limitation such that the exact evaluation of the Hessian and its inverse are quite expensive in computational terms.

- **Conjugate Gradient** can be regarded as intermediate between gradient descent and Newton's method. It is motivated by the desire to accelerate the typical slow convergence associated with gradient descent. This method also avoids the information requirements associated with the evaluation, storage and inversion of the Hessian matrix as required by the Newton's method.
- **Quasi-Newton Method** or variable matrix methods is developed to handle the drawbacks of Newton's method which is computational expensive. These methods instead of calculating the Hessian directly and then evaluating its inverse, building up an approximation at each epoch of the algorithm. This approximation is computed using only information on the first derivatives of the loss function. It is faster than gradient descent and conjugate gradient and the exact Hessian does not need to be computed and inverted.
- **Levenberg Marquardt Algorithm** is also called damped least-squares method. It has been designed to work specifically with loss functions. It takes the form of a sum of squared errors. It works without computing the exact Hessian matrix. Instead, it works with the gradient vector and the Jacobian Matrix. However, it requires a lot of memory and thus not recommended for a big data set and big neural network.

Ref [25] used scaled conjugate gradient algorithm to train an artificial neural network to provide recommendation to users at gas station, restaurant and attractions. Ref [26] designed a smart home technology for recognizing and detecting human activity using artificial neural network. Their work compared three algorithms – Quick Propagation, Levenberg Marqurdt and Batch Back Propagation algorithms. Batch algorithm computes the average error gradient across all nodes before updating the weights once at the end of epoch. Quick propagation is a variation of Newton's method but employs batch algorithm. They concluded that Levenberg Marquardt has better performance than Quick Propagation and Batch Back Propagation

Another arena which is also employed in ANN CARS implementation is genetic algorithm. This can be implemented using supervised or unsupervised learning algorithm [27] depending on how the fitness is evaluated. If the evaluation of the fitness is done by comparing them to known targets, then it becomes supervised. However, if the fitness is evaluated by testing offspring's ability to perform specific task, then it becomes reinforcement learning. If the fitness function is the ratio of the distance within a cluster to the ratio of the distance between clusters, then unsupervised learning is achieved. Fundamentally, genetic algorithm is used for searching and optimization problems. Decision tree mechanism can also be supervised or unsupervised [28]. Supervised learning is achieved when it is used for handling tasks that require target variable. However, unsupervised learning is obtained with clustering task. To pull decisions from multiple classifiers in data analysis, ensemble meta-algorithms are usually employed. There are three meta-algorithms, namely, bagging, boosting and stacking [29]. Boosting is primarily used to reduce bias and variance in supervised learning environment. It converts weak learners to strong ones. Bagging, also called Bootstrap aggregating on the other hand is used to improve the stability and accuracy of algorithms used in statistical classification and regression. Stacking works like boosting except that it determines the models that performed well and those that performed badly based on the given input data. Ref [30] presented an empirical analysis of Genetic Algorithm optimized Artificial Neural Network (GA-ANN), decision tree, bagging and boosting to perform the reasoning of the context. They concluded that GA-ANN produced correct recommendations with higher accuracy and efficiency.

V. WHY THE NEED FOR ROBUST MECHANISM

The methodologies employed in these recommenders have limitations. They rely solely on the internet in its operations. There are no universal recommendations for new users and new items. Its performance is therefore subjective [25]. Users need to make enough number of ratings of items before their interest can be deduced [20] so do new items; the system needs enough number of ratings for the item by users before it can be recommended [31]. These recommenders, therefore, employ scoring thresholds based on users' feedback to improve accuracy. Users who neglect the recommendations turn to have higher thresholds and lower accuracy which affects the system adversely [25]. Another problem associated with current recommender systems is sparsity [23] in which the number of available ratings is smaller than the number of unknown ratings in the user-item matrix. This makes it very difficult to provide accurate rating predictions. Items with a few ratings and users with unusual tastes are ignored in the recommendation prediction. Recommender systems tend to be over-specialized. Users are limited to recommendations related to the items they liked in the past only [25]. Systems also get confused due to repeated network calls when users return to the same location more than once [31]. Some algorithms have adverse effects on mobile devices and therefore cannot be implemented as mobile applications [32]. For instance, deep learning which requires more than one hidden layer and GA-ANN are computational-intensive algorithms that involve a cascade of many layers of nonlinear processing units for extracting and transforming features. To get maximum performance, recurrent neural networks are usually used [21]. They are therefore not desirable for mobile devices with limited processing capability.

VI. RECOMMENDATIONS FOR FUTURE RESEARCH

There have been impressive implementations of recommender systems using artificial neural network. These systems have brighter prosperity in the near future due to the stability of artificial neural network and also its ability to tolerate imprecision and partial truth. With the implementation of complex mechanisms for training the neural network complemented with its complicated architecture, more accurate results are achieved. However, these recommender systems are implemented to enhance context-aware computing. Mobile computing plays very important role in pervasive computing. There is therefore, the need to consider the limited capabilities in terms of storage and processing power of mobile devices such as mobile phones and tablets when these recommender systems are implemented. Thus learning algorithms and architectures of neural networks should be controlled though not compromising accuracy. Since a single neuron is practically useless in application development [18], a multi-layer perceptron instead of recurrent network could be employed in a supervised environment using gradient descent mechanism of back-propagation algorithm for training due to its simplicity and adaptability [24] through adaptive learning rate and momentum mechanism to achieve accurate results [33].

Another area that needs attention is the threshold scoring mechanisms for users and items employed in these applications. This limitation turns to make these recommender systems specialized systems for users without universal applicability. According to [34], users' background has effect on preference. It is therefore necessary to create an application to suite users with particular background characteristics. Researchers are therefore encouraged to conduct investigations into the dynamics of users' background in order to develop applications with universal applicability complemented with advanced specialization. This will eliminate the threshold scoring mechanisms currently adopted. The work of [32] supported this recommendation when they concluded that there is the need to determine the effect of users' static profile data on dynamic environment.

Finally, because these implementations rely solely on the internet, there is the need to develop robust algorithms to minimize frequent internet browsing. Users' will be more worrying if these applications turn to drain their browsing data unreasonably. Robust algorithms should be developed that keep significant findings in the local database of the user whilst occasionally connecting to the internet for update. These updates can be done at the background when users are using the internet for browsing

REFERENCES

[1] Dey, A. K., and Abowd, G. D. (2000). Towards a better understanding of context and context-awareness. Workshop on the What, Who, Where and How of Context-Awareness, affiliated with the 2000 ACM Conference on Human Factors in Computing Systems (CHI 2000)

[2] Larose, D. T. (2005). Discovering knowledge in data: An introduction to data mining. Wiley-Interscience

[3] Opoku, S.K. (2013). Agent-based automated negotiation system for handling vague data using artificial neural network with adaptive back propagation algorithm. International Journal of Advances in Computer Science and Technology (IJACST), 2(11), 247-255

[4] Lee, S., Ryu, J. H., Won, J. S., & Park, H. J. (2004). Determination and application of the weight for landslide susceptibility mapping using an artificial neural network. Engineering Geology, 71(3), 289-302

[5] Heaton, J. (2016). Introduction to neural network for java, 2nd Ed, Heaton Research, USA

[6] McCulloch, W. S., and Pitts, W. (1943), A logical calculus of ideas immanent in nervous activity. Bull, Mathematical 'Biophysics, 5, pp. 115-133

[7] Hebb, D. O. (1949). The organization of behavior. Wiley, New York.

[8] Rosenblatt, R. (1962). Principles of neurodynamics, Spartan Books, New York

[9] Widrow, B. & Smith, F. W. (1963). Pattern-recognizing control systems. Computer and information sciences symposium proceedings, Spartan Books, Washington DC

[10] Minsky, M., and Papert, S. (1969). Perceptrons: An introduction to computational geometry. MIT Press, Cambridge, Mass

[11] Kohonen, T. (1972). Correlation matrix memories. IEEE Transactions on computers, 21, 353-359

[12] Anderson, J. A. (1972). A simple neural network generating an interactive memory. Mathematical Biosciences, 14, 197-220

[13] Grossberg, S. (1976). Adaptive pattern classification and universal recording. I. parallel development and coding of neural feature detectors. Biological cybernetics, 23, 121-134

[14] Bechtel, W. & Abrahamsen, A (1993). Connectionism and the mind. Oxford, UK:Blackwell

[15] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. In proceedings of National Academy of Sciences. USA 79, 2554-2558

[16] Rumelhart, D. E. & McClelland, J. L. (1986). Parallel distributed processing: Exploration in the microstructure of cognition. MIT Press, Cambridge, Mass.

[17] Sutskever, I., Martens, J. & Hinton, G. E. (2011). Generating text with recurrent neural networks. In Proceedings of the 28th International Conference on Machine Learning (ICML-11) (pp. 1017-1024)

[18] Wakuya, H., & Shida, K. (2001). Bi-directionalization of neural computing architecture for time series prediction. III. Application to laser intensity time record “Data Set A”. In Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on (Vol. 3, pp. 2098-2103). IEEE

[19] Krenker, A., Bester, J., & Kos, A. (2011). Introduction to the Artificial Neural Networks, Artificial Neural Networks-Methodological Advances and Biomedical Applications, Prof. Kenji Suzuki (Ed.), ISBN: 978-953-307-243-2

[20] Lee, J., Lee, K., Jeong, E., Jo, J., & Shroff, N. B. (2017). CAS: Context-aware Background Application Scheduling in Interactive Mobile Systems. IEEE Journal on Selected Areas in Communications

[21] Lian, J., Zhang, F., Xie, X., & Sun, G. (2017, April). CCCFNet: A Content-Boosted Collaborative Filtering Neural Network for Cross Domain Recommender Systems. In Proceedings of the 26th International Conference on World Wide Web Companion (pp. 817-818).

[22] Jing, H., & Smola, A. J. (2017, February). Neural survival recommender. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (pp. 515-524). ACM.

[23] Martin, A., Zarate, P,. & Camillieri, G. (2017). A multi-criteria recommender system based on users’ profile management. In Multiple Criteria Decision Making (pp. 83-89). Springer International Publishing

[24] Quesada A. (2017, April 30). 5 algorithms to train a neural network. Retrieved from: https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network

[25] Salam Y. A (2016, February). A proactive multi-type context-aware recommender system in the environment of internet of things. (Master’s thesis Birzeit University)

[26] Mehr, H. D., Polat, H. & Cetin, A (2016, April). Resident activity recognition in smart homes by using artificial neural networks. In smart Grid Congress and Fair (ICSG), 2016. 4th International Istanbul (pp 1-5). IEEE

[27] Guerra, Luis, Laura M. McGarry, Víctor Robles, Concha Bielza, Pedro Larrañaga, and Rafael Yuste. "Comparison between supervised and unsupervised classifications of neuronal cell types: a case study." Developmental neurobiology 71, no. 1 (2011): 71-82

[28] Blockeel, Hendrik, Luc De Raedt, and Jan Ramon. "Top-down induction of clustering trees." arXiv preprint cs/0011032 (2000).

[29] Blockeel, Hendrik, Luc De Raedt, and Jan Ramon. "Top-down induction of clustering trees." arXiv preprint cs/0011032 (2000)

[30] Sachdeva, N., Dhir, R., & Kumar, A. (2016, August). Empirical analysis of Machine Learning Techniques for context aware Recommender Systems in the environment of IoT. In Proceedings of the International Conference on Advances in Information Communication Technology & Computing (p. 39). ACM

[31] Alexander Jr, E. E. (2017). AppXimity: a context-aware mobile application management framework (Doctoral dissertation, University of Victoria)

[32] Rodríguez-Hernández, M. D. C., & Ilarri, S. (2017). Researching Context-Aware Recommendation Systems in Mobile Environments. Jornada de Jóvenes Investigadores del I3A, 2, 59-60.

[33] Hassan, M., & Hamada, M., Performance comparison of featured neural network trained with back-propagation and delta rule techniques for movie rating prediction in multi-criteria recommender systems. Informatica, 40(4), 409, 2016

[34] Opoku, S. K., & Appiah, S. (2016). Automating students' activities in Higher Educational Institutions, International Journal of Computer Applications Technology and Research, 5(11), 693 - 697.