RESEARCH ARTICLE OPEN ACCESS

A Survey on Neural Network Centric Classifier and Clustering Approaches

Dr. Nilamadhab Mishra[1]

[1]Post Graduate Teaching & Research Dept., 09 School of Computing, Debre Berhan University, Debre Berhan 445, Ethiopia.

Abstract:

The data mining process takes data from a data warehouse as input and identifies the hidden patterns. It also extracts hidden predictive information from data warehouse through the Neural Networks tools. The progressive statistical and computational deep data mining process attract the business enterprise towards transforming their huge data into business goldmines. Normally, classification is done through supervised learning mechanism and clustering through unsupervised learning mechanism. In this work, a review is prepared based on the pattern credentials through classifier and clustering methods. Some experimentation has been done to learn and train the neural network architecture for the predefined data set with different activation functions. The review and discussion inspect only the neural based classification and clustering methods.

Keywords **— classification, clustering, neural network, activation function, data mining, learning algorithm**

I. INTRODUCTION

Data mining encourages the extraction of hidden predictive information from large databases. It is a powerful technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviours, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were time consuming to resolve. They clean databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations. The neural network centric data mining classifies the objects by learning nonlinearity.

The classification is the process of finding a set of models (or functions) that describe and distinguish data classes or concepts to predict the class of object whose class label is unknown.

The derived model is based on the analysis of a set of training data (i.e. data objects whose class label is known. The neural network centric process records one at a time, and "learn" by comparing their classification of the record with the known actual classification of the record. The errors from the initial classification of the first record is fed back into the network, and used to modify the networks algorithm the second time around, and so on for many iterations [**1**], [**2**], [**3**]. Approximately idiom, a neuron in a neural network is a set of input values and associated weights and a function that sums the weights and maps the results to an output. The input layer is composed not of full neurons, but rather consists simply of the values in data records, that constitute inputs to the next layer of neurons. The next layer is called a hidden layer; there may be several hidden layers. The final layer is the output layer, where there is one node for each class. A single sweep forward through the network results in the assignment of a value to each output node, and the record is assigned to whichever class's node had the highest value.

Clustering analyses data objects without consulting a known class label. Cluster of objects are formed so that object within a cluster have high similarity in comparison to one another but are very

dissimilar to objects in other clusters. Each clustering that is formed can be viewed as a class of objects from which rule can be derived. In clustering problems, you want a neural network to group data by similarity [**4**], [**5**], [**6**]. For example: market segmentation done by grouping people according to their buying patterns; data mining can be done by partitioning data into related subsets; or bioinformatics analysis such as grouping genes with related expression patterns. The Neural Network Clustering Tool will help you select data, create and train a network, and evaluate its performance using a variety of visualization tools.

For the purpose of constructing a classifier it is necessary to determine what parameters influence a decision of ranging a pattern to this or that class. Two problems can arise. First, if the number of parameters is small the situation can develop when the same set of initial data corresponds to examples in different classes [**7**], [**8**], [**9**], [**10**]. It will be impossible to train the neural network then and the system will not work correctly (it is impossible to find the minimum corresponding to such an initial data set). Initial data must not be contradictive. To solve this problem it is necessary to increase dimensionality of the attributes space (number of components of the input vector corresponding to the pattern). But after increasing the dimensionality of the attributes space we can face a situation when a number of attributes would not be enough for training the system and instead of generalization it will simply remember the training samples and will not function correctly. Thus when determining the attributes we have to find a compromise with their number [**11**], [**12**], [**13**], [**14**], [**15**].

Further it is necessary to find a method of representing input data for the neural network, i.e. determine a method of normalization. Normalization is required because neural networks only work with data represented by numbers in the range between 0 and 1 while input data can have arbitrary range or can be non-numerical data at all. Various methods can be used, from simple linear transformation to the required range to multivariate analysis of parameters and non-linear normalization, depending on cross-impact of the parameters.

From a theoretical point of view, supervised and unsupervised learning differ only in the causal structure of the model. In supervised learning, the model defines the effect one set of observations, called inputs, has on another set of observations, called outputs. In other words, the inputs are assumed to be at the beginning and outputs at the end of the causal chain. The models can include mediating variables between the inputs and outputs. With unsupervised learning it is possible to learn larger and more complex models than with supervised learning. This is because in supervised learning one is trying to find the connection between two sets of observations. The difficulty of the learning task increases exponentially in the number of steps between the two sets and that is why supervised learning cannot, in practice, learn models with deep hierarchies [**16**], [**17**], [**18**], [**19**], [**20**].

In unsupervised learning, all the observations are assumed to be caused by latent variables, that is, the observations are assumed to be at the end of the causal chain. In practice, models for supervised learning often leave the probability for inputs undefined. This model is not needed as long as the inputs are available, but if some of the input values are missing, it is not possible to infer anything about the outputs. If the inputs are also modelled, then missing inputs cause no problem since they can be considered latent variables as in unsupervised learning. In unsupervised learning, the learning can proceed hierarchically from the observations into ever more abstract levels of representation. Each additional hierarchy needs to learn only one step and therefore the learning time increases (approximately) linearly in the number of levels in the model hierarchy [**21**].

The rest of this paper is organized as follows. Section **II** discusses the neural network configuration analysis. Section **III** discusses the data pre-processing approach. Section **IV** discusses the implemented method. Section **V** discusses the discussion and result view. Finally, section **VI** concludes this paper.

II. NEURAL NETWORK CONFIGURATION ANALYSIS

To describe networks having multiple layers, the notation must be extended. Specifically, it needs to make a distinction between weight matrices that are connected to inputs and weight matrices that are connected between layers. It also needs to identify the source and destination for the weight matrices. We will call weight matrices connected to inputs input weights; we will call weight matrices coming from layer outputs layer weights. Further, superscripts are used to identify the source (second index) and the destination (first index) for the various weights and other elements of the network. To illustrate, the one-layer multiple input network shown earlier is redrawn in abbreviated form below. A network can have several layers. Each layer has a weight matrix W, a bias vector b, and an output vector a. To distinguish between the weight matrices, output vectors, etc., for each of these layers in the figures, the number of the layer is appended as a superscript to the variable of interest. You can see the use of this layer notation in the three-layer network shown below, and in the equations at the bottom of the figure. The network shown above has R1 inputs, S1 neurons in the first layer, S2 neurons in the second layer, etc. It is common for different layers to have different numbers of neurons. A constant input 1 is fed to the bias for each neuron. Note that the outputs of each intermediate layer are the inputs to the following layer. Thus layer 2 can be analysed as a one-layer network with S1 inputs, S2 neurons, and an S2xS1 weight matrix W2. The input to layer 2 is a1; the output is a2. Now that all the vectors and matrices of layer 2 have been identified, it can be treated as a single-layer network on its own. This approach can be taken with any layer of the network. The layers of a multilayer network play different roles. A layer that produces the network output is called an output layer. All other layers are called hidden layers. Multiple-layer networks are quite powerful. For instance, a network of two layers, where the first layer is sigmoid and the second layer is linear, can be trained to approximate any function (with a finite number of discontinuities) arbitrarily well. This kind of two-layer network is used extensively in Backpropagation. Here it is assumed that the output of the third layer, a3, is the network output of interest, and this output is labelled as y. This notation is used to specify the output of multilayer networks (**figure-1**) [**22**], [**23**].

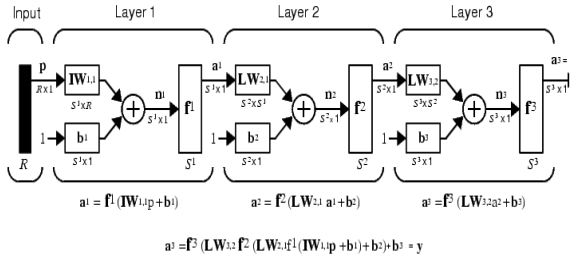

Figure-1: A configurable neural network structure (**sources: - mathwork.com**)

A neural network may be considered as a data processing technique that maps, or relates, some type of input stream of information to an output stream of data‘ Neural Networks (NNs) can be used to perform classification. Any function can be approximated to arbitrary accuracy by a neural network. NNs are consisted of neurons (or nodes) distributed across layers. The way these neurons are distributed and the way they are linked with each other define the structure of the network. Each of the links between the neurons is characterized by a weight value. A neuron is a processing unit that takes a number of inputs and gives a distinct output. Apart from the number of its inputs it is characterized by a function f known as transfer function or action function or learning function.

III. DATA PREPROCESSING

Data filtration is the initial data collection and filtration. Today’s real world databases are highly susceptible to noisy, missing and inconsistent data. The data be preprocessed in order to improve the quality of the data and consequently, of the mining result”. There are number of data preprocessing technique. Data cleaning encourages removing noisy, inconsistent data. Attributes have no recorded value or unusual values. Data cleaning is done by filling the missing data or by smoothening noisy data. Data quality includes in all three datasets there are no missing values. However very often in datasets, there exist samples that do not

comply with the general behavior or model of the data. Such data samples are called outlier. The data integration includes merge data from multiple sources into a coherent data store such as warehouse or a data cube. The data transformation includes data are transformed on consolidated into forms appropriate for mining. Data transformation can involve Smoothening-one approach is by replacing each bin value by bin median.

Second approach is smoothening by boundaries i.e. the min and max values are funded as boundary. The generalization includes low level data are replaced by high level data use of concept hierarchies. The data normalization includes normalization improve the accuracy and efficiency of mining algorithm involving distance measurements. There are many methods for normalization, such as min-max normalization, z-score normalization, and normalization by decimal scaling.

In Data formation context, we divide the datasets into subsets. These subsets form two major categories, sets that will be used to define the parameters of the models and sets that will be used to measure their prediction ability. The neural networks are adjusted (trained) on a part of the available data and tested on another part. Again we will use Set B to adjust the parameters of the models and Set C to measure their prediction ability. This way we will be able to make comparisons of the performance of both types of models on the same dataset. In this study we will use the term 'Training set' for Set B and 'Test set' for Set C. additionally, due to the nature of the parameters adjustment of the neural network models we need to divide the training set (set B) into three new subset [**24**], [**25**].

IV. METHOD DISCUSSIONS

The mechanism of weights update is known as training algorithm. There are several training algorithms proposed in the literature. We will give a brief description of those that are related with the purposes of our project study. The algorithms described here are related to feed-forward networks. A NN is characterized as feed-forward network "if it is possible to attach successive numbers to the inputs and to all of the hidden and output units such that each unit only receives connections from inputs or units having a smaller number". All these algorithms use the gradient of the cost function to determine how to adjust the weights to minimize the cost function. The gradient is determined using a technique called back propagation, which involves performing computations backwards through the network. Then the weights are adjusted in the direction of the negative gradient.

The training algorithm of back propagation involves four stages.

1. Initialization of weights
2. Feed forward
3. Back Propagation of errors
4. Updating of the weights and biases.

A three-layer neural network consists of an input layer, a hidden layer and an output layer interconnected by modifiable weights represented by links between layers. The feed forward operations consists of presenting a pattern to the input units and passing (or feeding) the signals through the network in order to get outputs units (no cycles!)

In weights Adjustment context, the power of NN models lies in the way that their weights (inter unit-connection strengths) are adjusted. The procedure of adjusting the weights of a NN based on a specific dataset is referred as the training of the network on that set (training set). The basic idea behind training is that the network will be adjusted in a way that will be able to learn the patterns that lie in the training set. Using the adjusted network in future situations (unseen data) it will be able based on the patterns that learnt to generalize giving us the ability to make inferences. In our case we will train NN models on a part of our time series (training set) and we will measure their ability to generalize on the remaining part (test set). The size of the test set is usually selected to be 40% of the samples. The way that a network is trained is depicted by the plotted m- figure. Each sample consists of two parts the input and the target part (supervised learning). Initially the weights of the network are assigned random values (usually within [-1 1]). Then the input part of the first sample is presented to the network. The network computes an output based on:

the values of its weights, the number of its layers and the type and mass of neurons per layer.

In context to learning rate, Larger the learning rate the bigger the step. If the learning rate is made too large the algorithm will become unstable and will not converge to the minimum of the error function. If the learning rate is set too small, the algorithm will take a long time to converge. Methods suggested for adopting learning rate are as follows. Start with a high learning rate and steadily decrease it. Changes in the weight vector must be small in order to reduce oscillations or any divergence. A simple suggestion is to increase the learning rate in order to improve performance and to decrease the learning rate in order to worsen the performance.

In sequential learning a given input pattern is propagated forward, the error is determined and back propagated, and weights are updated. In batch learning the weights are updated only after the entire set of training network has been presented to the network. Thus the weight update is only performed after every epoch.

In context to stop training, a significant decision related with the training of a NN is the time on which its weight adjustment will be ceased. As we have explained so far over-trained networks become over-fitted to the training set and they are useless in generalizing and inferring from unseen data. While under-trained networks do not manage to learn all the patterns in the underlying data and due to this reason under perform on unseen data. Therefore there is a tradeoff between over-training and under-training our networks. The methodology that is used to overcome this problem is called validation of the trained network. Apart from the training set a second set, the validation set, which contains the same number of samples, is used. The weights of the network are adjusted using the samples in the training set only. Each time that the weights of the network are adjusted its performance (in terms of error function) is measured on the validation set. During the initial period of training both the errors on training and validation sets are decreased. This is due to the fact that the network starts to learn the patterns that exist in the data. From a number of iterations of the training algorithm and beyond the network will start to over fit to the training set. If this is the case, the error in the validation set will start to rise. In the case that this divergence continues for a number of iterations the training is ceased. The output of this procedure would be a not over fitted network. After describing the way that a NN works and the parameters that are related to its performance we select these parameters in a way that will allow us to achieve optimum performance in the task we are aiming to accomplish. The methodology will follow in order to define these parameters is described in the next paragraph.

One of the major advantages of neural nets is their ability to generalize. This means that a trained net could classify data from the same class as the learning data that it has never seen before. In real world applications developers normally have only a small part of all possible patterns for the generation of a neural net. To reach the best generalization, the dataset should be split into three parts: First, the training set is used to train a neural net. The error of this dataset is minimized during training. Second, the validation set is used to determine the performance of a neural network on patterns that are not trained during learning. And thirdly, a test set is prepared for finally checking the overall performance of a neural network [**26**], [**27**], [**28**].

V. EXPERIMENT AND RESULT VIEW

IRIS & Breast cancer dataset do not fluctuate randomly. In this work, I discuss the experiment for identifying the patterns. The reports of the results are implemented and analyzed. Different neural network model are used with variable size hidden units using different activation function to select the best model and its performance was evaluated on testing data set.

A. DISCUSSION OUTCOMES ON IRIS DATASET [29]

%70 training set
%30 testing set
Function: Hyperbolic tangent (tanh)
Sample: Random

Table-1: Classifications accuracy analysis on IRIS based on NN1, NN2, and NN3

Alpha	Model	Epoch	Time	Structure	Accuracy
0.1	NN1	1000	20.485s	4--3	20%
0.06	NN1	1000	21.345s	4--3	30%
0.1	NN2	300	5.81s	4--5--3	40%
0.06	NN2	500	8.251s	4--5--3	80%
0.1	NN3	250	3.900s	4-4-2-3	99.9
0.06	NN3	6000	80.600s	4-6-5-3	99.984

The **Table-1** states that in NN3 structure, the classification accuracy is much more as compared to NN1 and NN2 structures. The **figure-2** identifies the number of pattern clusters in the IRIS dataset.

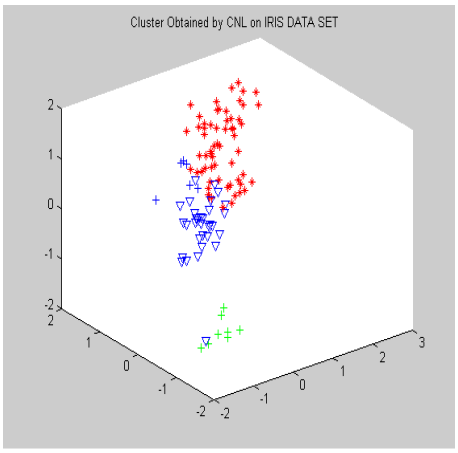

Figure-2: Cluster pattern analysis on iris dataset

B. DISCUSSION OUTCOMES ON BREAST CANCER DATASET [30]

%80 training set
%20 testing set
Function: Hyperbolic tangent (tanh)
Sample: Random

Table-2: Classifications accuracy analysis on breast cancer based on NN structure (10-4-4-2)

Alpha	Model	Time	Accuracy
0.05	NN	12.367s	99.9456

In **table-2**, I take learning rate alpha= 0.05 and find that in 200 epochs, the 99.9% classification accuracy can be achieved on breast cancer based on NN structure (10-4-4-2). In **figure-3**, an actual cluster pattern analysis is made based on breast cancer dataset.

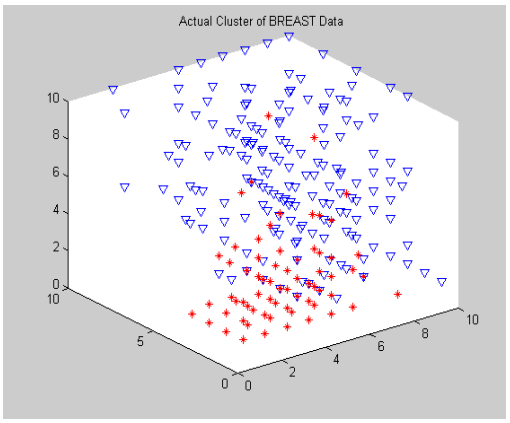

Figure-3: Cluster pattern analysis on breast cancer dataset

VI. CONCLUSIONS

The success of neural network architecture depends heavily on the availability of effective learning algorithms. The theoretical strength of the "Back propagation neural network" is yet to be used in hundreds of technologies and more accurate result may lead if bigger the database size, if more no. of dimension involves, if more correct data without noise and out layer, and if the data provide by multiple agencies. The neuro-fuzzy approach has considerable industrial applicability due to its high embedding compatibility with heterogeneous microcontroller devices. The de-fuzzification process provides a method of extracting a crisp value from the fuzzy quantifiers as approximate representative values. The machine learning technics are implemented in many areas of knowledge discovery and semantic knowledge analytics to explore the application intelligence. The different frameworks and algorithms are designed and explored for knowledge discovery, representation, semantic analytic, and inferences. The real world applications are modeled through smart architectures, algorithms, and frameworks to accomplish the knowledge analytics tasks.

ACKNOWLEDGMENT

The author would like to express thanks to the Post Graduate Teaching & Research Dept., at School of Computing, Debre Berhan University, Ethiopia for supporting this research.

REFERENCES

[1] *Gabrys, B., & Bargiela, A. (2000). General fuzzy min-max neural network for clustering and classification. IEEE transactions on neural networks, 11(3), 769-783.*

[2] *Demuth, H., & Beale, M. (1993). Neural Network Toolbox for Use with Matlab--User'S Guide Verion 3.0.*

[3] *Mishra, N., Lin, C. C., & Chang, H. T. (2014). Cognitive inference device for activity supervision in the elderly. The Scientific World Journal, 2014.*

[4] *Demuth, H., & Beale, M. (2000). Neural network toolbox: for use with Matlab: computation, visualization, programming: User's guide, version 4. The Mathworks.*

[5] *Mishra, N., Lin, C. C., & Chang, H. T. (2015). A cognitive adopted framework for IoT big-data management and knowledge discovery prospective. International Journal of Distributed Sensor Networks, 11(10), 718390.*

[6] *Nørgård, P. M. (1997). The Neural Network Based System Identification Toolbox: For use with MATLAB.*

[7] *Mishra, N., Lin, C. C., & Chang, H. T. (2014, December). A cognitive oriented framework for IoT big-data management prospective. In Communication Problem-Solving (ICCP), 2014 IEEE International Conference on (pp. 124-127). IEEE.*

[8] *Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). Neural network design (Vol. 20). Boston: Pws Pub.*

[9] *Chang, H. T., Mishra, N., & Lin, C. C. (2015). IoT Big-Data Centred Knowledge Granule Analytic and Cluster System for BI Applications: A Case Base Analysis. PloS one, 10(11), e0141980.*

[10] *Duin, R. P. W., Juszczak, P., Paclik, P., Pekalska, E., De Ridder, D., Tax, D. M. J., & Verzakov, S. (2000). A matlab toolbox for pattern recognition. PRTools version, 3, 109-111.*

[11] *Mishra, N., Chang, H. T., & Lin, C. C. (2014). Data-centric knowledge discovery strategy for a safety-critical sensor application. International Journal of Antennas and Propagation, 2014.*

[12] *Simpson, P. K. (1992). Fuzzy min-max neural networks. I. Classification. IEEE transactions on neural networks, 3(5), 776-786.*

[13] *Mishra, N., Chang, H. T., & Lin, C. C. (2015). An Iot knowledge reengineering framework for semantic knowledge analytics for BI-services. Mathematical Problems in Engineering, 2015.*

[14] *Mishra, N. (2011). A Framework for associated pattern mining over Microarray database. International Journal of Global Research in Computer Science (UGC Approved Journal), 2(2).*

[15] *Hepner, G., Logan, T., Ritter, N., & Bryant, N. (1990). Artificial neural network classification using a minimal training set- Comparison to conventional supervised classification. Photogrammetric Engineering and Remote Sensing, 56(4), 469-473.*

[16] *Mishra, N., Chang, H. T., & Lin, C. C. (2018). Sensor data distribution and knowledge inference framework for a cognitive-based distributed storage sink environment. International Journal of Sensor Networks, 26(1), 26-42.*

[17] *Lippmann, R. P. (1989). Pattern classification using neural networks. IEEE communications magazine, 27(11), 47-50.*

[18] *Mishra N, (2017). "In-network Distributed Analytics on Data-centric IoT Network for BI-service Applications", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN: 2456-3307, Volume 2, Issue 5, pp.547-552, September-October.2017.*

[19] *Huang, J., Wang, Y., Tan, T., & Cui, J. (2004, August). A new iris segmentation method for recognition. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on (Vol. 3, pp. 554-557). IEEE.*

[20] *Patnaik, B. C., & Mishra, N. (2016). A Review on Enhancing the Journaling File System. Imperial Journal of Interdisciplinary Research, 2(11).*

[21] *Chavent, M. (1998). A monothetic clustering method. Pattern Recognition Letters, 19(11), 989-996.*

[22] *Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In Neural networks for perception (pp. 65-93).*

[23] *Heermann, P. D., & Khazenie, N. (1992). Classification of multispectral remote sensing data using a back-propagation neural network. IEEE Transactions on Geoscience and Remote Sensing, 30(1), 81-88.*

[24] *Cannas, B., Fanni, A., See, L., & Sias, G. (2006). Data preprocessing for river flow forecasting using neural networks: wavelet transforms and data partitioning. Physics and Chemistry of the Earth, Parts A/B/C, 31(18), 1164-1171.*

[25] *Chang, H. T., Li, Y. W., & Mishra, N. (2016). mCAF: a multi-dimensional clustering algorithm for friends of social network services. Springer Plus, 5(1), 757.*

[26] *Delen, D., Walker, G., & Kadam, A. (2005). Predicting breast cancer survivability: a comparison of three data mining methods. Artificial intelligence in medicine, 34(2), 113-127.*

[27] *Chang, H. T., Liu, S. W., & Mishra, N. (2015). A tracking and summarization system for online Chinese news topics. Aslib Journal of Information Management, 67(6), 687-699.*

[28] *Abbass, H. A. (2002). An evolutionary artificial neural networks approach for breast cancer diagnosis. Artificial intelligence in Medicine, 25(3), 265-281.*

[29] *Gabrys, B., & Bargiela, A. (2000). General fuzzy min-max neural network for clustering and classification. IEEE transactions on neural networks, 11(3), 769-783.*

[30] *Wu, Y., Giger, M. L., Doi, K., Vyborny, C. J., Schmidt, R. A., & Metz, C. E. (1993). Artificial neural networks in mammography: application to decision making in the diagnosis of breast cancer. Radiology, 187(1), 81-87.*

Authors Profile

Dr. Nilamadhab Mishra is currently an Assistant Professor in Post Graduate Teaching & Research Dept., at School of Computing, Debre Berhan University, Ethiopia. He accomplishes his PhD in Computer Science and Information Engineering from Chang Gung University, Taiwan. He moreover publishes numerous peer reviewed researches in Thomson Reuter's ranked SCI journals & IEEE conference proceedings, and serves as reviewer and editorial member in peer reviewed Journals and Conferences. Dr. Mishra's research areas focus on Network Centric Data Management and Knowledge Discovery, IoT Data Science and Knowledge Analytics, Business Intelligence, and Cognitive Applications exploration. He has 15 years of Academic Teaching and Research Experience.