Digital Technologies, 2018, Vol. 3, No. 1, 1-8
Available online at http://pubs.sciepub.com/dt/3/1/1

DOI:10.12691/dt-3-1-1

Modeling the Computational Solution of Market Basket Associative Rule Mining Approaches Using Deep Neural Network

A.A. Ojugo[1,*], A.O. Eboka[2]

[1]Department of Mathematics/Computer Science, Federal University of Petroleum Resources Effurun, Delta State, Nigeria
[2]Department of Computer Science Education, Federal College of Education Technical, Asaba, Delta State, Nigeria
*Corresponding author: ojugo.arnold@fupre.edu.ng

Received September 06, 2018; Revised October 25, 2018; Accepted November 09, 2018

Abstract Data is an important property to everyone and lots of it is generated daily. The large amount of data available in the world today, is stored in repositories, databanks, data warehouses etc. Generated data is further on the rise with the Internet, resulting in the consequent explosion of data and its usage. Data convergence over the Internet, has made it more imperative to analyze data relations due to the tremendous sizes that scales up to petabytes of data. But, there exists inherent challenges of extracting useful data from these large repositories. Thus, focal point of this study is to model a rule-based computational solution to the inherent challenge. We thus propose the use of a market basket dataset mining using a hybrid deep learning associative rule mining heuristic.

Keywords: *market basket, associative rule mining, data mining, predictive, descriptive, deep learning, evolutionary*

Cite This Article: A.A. Ojugo, and A.O. Eboka, "Modeling the Computational Solution of Market Basket Associative Rule Mining Approaches Using Deep Neural Network." *Digital Technologies*, vol. 3, no. 1 (2018): 1-8. doi: 10.12691/dt-3-1-1.

1. Introduction

The advent of the Internet, mobile smartphones and its corresponding services has facilitated an explosion into the amount generated as well as ushered in the realization of Information at your fingertips anytime and everywhere. Advances in the field of ubiquitous computing now provides users with new add-on, consumption and delivery model to information communication tech and its allied services that are scalable, reliable, device-independent as well as on-demand. It focuses on reducing the inherent limitations of mobile devices via the use of computing techniques for storage and processing of data [1,2].

Anomaly detection searches for items that do not conform to an expected pattern. Detected patterns or anomalies then translate to critical, actionable data or outliers, and is applied to various domain tasks. Association rule learning searches for relationships among variables. A supermarket may gather data about how customers purchase various products. Association rules help managers to identify which products are frequently bought together [3]. Thus, data can be used for marketing purposes – termed market basket analysis. The clustering analysis effect therein can help business owners to discover group structures and hidden patterns in the dataset. Classification generalizes known structure and applies them to new data. E.g. A mail program may attempt to classify a mail as genuine or as spam. Regression seeks to find a function that models data with the least error. Summarization provides a compact representation of dataset to include visualization and report generation [4,5].

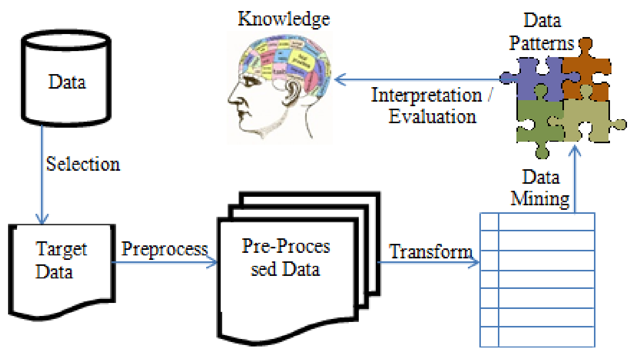

Figure 1. Knowledge Discovery in Database Process [5]

Data mining is simply a technique to process data, select it, integrate it and retrieve some useful information. Data mining tools allow users to analyse, categorize and summarize the relations among data. It discovers useful data within a large amount of relational databases. Data mining performs various activities using technique such as clustering, classification, prediction, association learning etc. Data mining techniques seek to advance this feat as it extracts hidden relationships in data pools, which are located on remotely accessed storage servers or cloud. A great amount of data ranging from commerce, marketing, surveillance fraud detection, scientific discovery, medicine, geographical data system, ecosystem etc – all requires deep analysis and proper decision making, all of which can be achieved via data mining [6].

Data mining thus, uses database analysis to discover the many hidden relations and useful patterns embedded within a data pool, so as to help organizations and users to better focus on essential data in warehouses by predicting future occurrences and behaviors as it grants business the opportunity to make informed, proactive decisions. It use methods like anomaly detection, association rule mining, neural networks, sequence and path analysis, clustering and classification amongst others to resolve issues in the field of Big-Data (as not all information are useful to the user). Thus, it has become very necessary to extract only useful data from the huge amount of data available, and this is dependent on the kind of data to be mined [7,8].

1.1. Database and Data Warehousing

Databases are classified based on many criteria such as data models, data types, applications involved etc – each of which may require its own data mining technique. Data mining systems can also thus, classified accordingly. Classifying data mining by data models will yield relational, transactional, object-relational and data warehouse mining system. In classifying based on data types handled, it yields spatial, time-series, text, stream data, multimedia, or a World Wide Web mining system [9,10]. In classifying based on kinds of knowledge they mine (i.e. based on data mining functionalities such as data characterization, discrimination, association and correlation analysis) – it can yield ***predictive*** (e.g. classification, regression, time series, prediction etc) and/or ***descriptive*** (clustering, summarization, association rules, sequence discovery, outlier and evolution analysis). A comprehensive data mining system aims to give multiple and integrated mining functionalities [11]. Ultimately, data mining method is classified into ***predictive*** or ***descriptive***. Predictive data mining uses historical data to make inferences about future happenings as it uses observed data to construct model for making presages on unseen future events [12]; While, descriptive data mining aims to locate patterns in the data that offer some details about internal hidden relations as well as characterizing the overall properties of the data and represent it in a meaningful way [13].

1.2. Market Basket Analysis and Its Implication

Majority of businesses apperceived, continues to amass large amount of data from their customers over time. Also, with the growth of e-commerce, businesses now have an exponential quantity of data to store. Data mining (Knowledge Discovery in Databases – KDD) helps to spot patterns, correlations and anomalies in dataset that can avail to engender precise future decisions [14,15]. Thus, in analyzing data for market basket analysis, businesses are more interested in knowing the item sets that frequently occur together during purchase. Data acquired can be useful for adverts, cross-selling, identification of customers based on buying patterns, adjustment of store layouts (i.e. placing items optimally with respect to each other), and for catalogue design [16].

Market basket analysis (MBA) is a data mining methods for analyzing the association of data items and their frequency of occurring together from everyday buying and selling. The rudimental conception here is to find the associated pairs of items in a store from the transactional dataset stored in the database. It can become a springboard thus, to promote certain items together, to control stocks more perspicaciously, and to increase the probability of purchasing. Thus, managers can place such items associated at the neighboring shelf as a means to give the business a better chance of profit making (even though marketing control and controlling the order of goods is also needed). Online shopping is a form of market basket whereby; shoppers directly buy goods from a seller in real time, through the use of the Internet. In the same way web-shop, virtual stores, an online shop or e-shop are the physical analogy, of purchasing products in a shopping mall [17].

1.3. Associative Rule Mining

Many businesses overtime have massive amount of customer purchasing data. Using this data through efficient analysis will help some businesses to know when to stock up inventory so as to reduce payment of demurrage as well as avoid customer loss. Conversely, other businesses may wish to know how, when, which and what goods are ordered as a sub-group and in what combination the purchasing cum viewing behaviour of customers are done. This way, such businesses may seek to cross-promote some goods through deals, strategically place some goods near others, recommend some goods alongside others, and as their goal, effectively guide customers or clients towards efficient purchasing pattern [18].

Data mining has become crucial to help extract and identify utilizable data. The purchase sequence (log) seeks to provide a description of the changes in a buyer's preferences over time. This is known as the concept drift of a customer, and is effectively managed via association analysis yielding the technique association rule mining (ARM). ARM is an effective data mining scheme that employs knowledge discovery in databases (KDD) dataset that often includes data pre-process, selection, transform, mining and interpretation. Succinctly put, the extraction of hidden patterns of predictive data, and transforming such data into understandable structure and pattern, is the focal point and major goal of data mining algorithms. Association analysis is the art of aiming to understand the patterns inherent in large data cluster. Thus, association analysis aims to yield rules or heuristics that seek to learn the underlying probability with which goods are correlated, combined or co-occurs in a basket (above certain threshold) as a customer sought to buy or purchase these goods [19].

Association rule describes how often items are purchased together when a shopper passes through a point of sale thereby registering the contents of his market basket. This, result in the colossal accumulations of basket data – yielding information about items sold in combinations or together. Thus, it seeks to find unsuspected relations by creating frequency with which items (dataset) are combined for prediction. Frequent itemset are those that occur at least a given number of times for shoppers (also called minimum support threshold). Such rules become helpful to aid decisions like promotions, store layout, product pricing, cross-selling amongst others. Association rule mining asserts and recuperates the maximum likelihoods, for any item (or combination of items) to be purchased [20].

2. Association Rule Mining Algorithms

2.1. AIS Algorithm

The algorithm was first proposed by Agrawal, Imielinski, and Swami [3] to create association rule. It improves the quality of databases together with necessary functionality to process decision support queries. Only one item consequent association rules is generated. Thus, the consequent of all those rules only contain one item. For example, rules like $X \cap Y \Rightarrow Z$ can be generated; But, not like $X \Rightarrow Y \cap Z$.

Here, the databases are scanned severally to get frequent itemsets and support count of each item accumulated during the first pass over the database. Based on the minimal support count, items whose support count is less than its minimum value gets eliminated from list of item. Candidate 2-itemsets is then generated by extending frequent 1-itemsets with other items in transaction. At second pass over the database, support count of candidate 2-itemsets are accumulated and checked against the support threshold. Also, candidate (k+1)-itemsets are generated by extending frequent k-item sets with items in the same transaction. The candidate itemsets generation and frequent itemsets generation process iterate until any one of them becomes empty. To make AIS more efficient and robust, the estimation method helps to prune itemsets candidates that have no hope to be large. The unnecessary effort of counting itemsets is avoided. Since all candidate and frequent itemsets are stored in the main memory, memory management is also proposed for AIS when memory is not enough [21].

2.2. SETM Algorithm

Here, each candidate itemsets is generated on-the-fly by scanning the database and counting at the end of the pass. New candidate itemsets are generated in same way as in AIS; But, transaction identifier (TID) of the generating transaction is saved with the candidate itemset in a sequential structure. It separates candidate generation process from counting. At the end of a pass, the support count of each itemsets is determined by aggregating the sequential structure. It has same demerit as the AIS and for each candidate itemset, there are many entries as its support value [5,7,22].

2.3. Apriori Algorithm

It is used in frequent itemset mining and association rule learning – the algorithm uses a level-wise search, where k-itemsets (with *k*-items) are used to explore (k+1) itemsets, to mine frequent itemsets from transactional database for Boolean association rules. Thus, frequent subsets are extended one item at a time so as to generate each candidate. Then, groups of candidates are tested against the data. To count candidate item sets efficiently, Apriori uses breadth-first search method and a hash tree structure. It identifies frequent individual items in the database and extends them to larger and larger item sets so long those itemsets appear sufficiently often in the database. It determines frequent itemsets that can be used via association rules that highlight the general trends in the database. It two major drawbacks are: (a) its complex candidate generation process uses most of the time, space and memory, and (b) requires multiple scan over the database to generate candidate itemsets [5]. The pseudocode is thus:

```
CI_k : Candidate itemset having size k
FI_k : Frequent itemset having size k
FI_1 = {frequent items};
For (k=1; FI_k != null; k++) do begin
CI_k+1 = candidates generated from FI_k;
For each transaction t in database D do
Increment count value of candidates in CI_k+1 contained in t
FI_k+1 = candidates in CI_k+1 with min_support
End
Return FI_k;
```

Many variant exists for Apriori-model to mine itemset by eliminating candidates generated and storing its support count in specialized data structure [23]. Apriori-TID is a variant of traditional Apriori [7]. The process of candidate itemset generation is same as the traditional Apriori. Another set C' is generated of which each member has TID of each transaction and itemsets present in this transaction. The C'-set generated is used to count the support of each candidate itemset. Its merit is that, in the later passes, AprioriTID outperforms traditional Apriori [22].

Another variant is the ***Apriorihybrid*** introduced by Agrawal in 1994. It uses apriori data at its initial stage of the process; It then switches to Apriori transaction ID soon as it suspects a candidate item sets is passing into the memory [24]. In a nutshell, the AprioriHybrid adopts an Apriori in the earlier pass and adopts the AprioriTID at the later stages. Though the ***hybrid*** model is more efficient as well as outperforms both the traditional Apriori and Apriori-TID combined – it has the inherent challenge of encoding dataset and resolving the structural dependencies imposed on it by hybridization of the underlying heuristics used to form such hybrid. Another issue experienced is the ability to properly encode data using the underlying heuristics.

2.4. Frequency Pattern-Growth Algorithm

It is scalable and efficient for mining itemsets to generate frequent patterns by pattern fragment growth. It extends the prefix-tree structure to store crucial, compressed data of frequent patterns generated (i.e. FP-tree). It outperforms other model [25] via its two-phased structure. In phase ***one***, the model counts the number of times an item occurs (attribute-value pairs) in the dataset, and goes ahead to store them in a 'header table'. In phase ***two***: it builds the FP-tree structure by inserting instances of each item sorted in a descending order based on their frequency in the dataset; And, allows quick processing in tree. Items whose instances do not meet a minimum coverage support threshold are discarded [14]. The tree is highly compressed at its root when many instances shares frequent items. It uses recursive (compressed version) processing of main dataset to grow large itemsets directly; Instead of generating candidate items. Growth starts at the bottom of the header table (having longest branches). It finds all instances that match given condition. New tree is created as the counts projected from original FP-tree (corresponding to set of instances conditional on the attribute). Thus, each node gets the sum of its children's count. Also, recursive growth ends when no individual items conditional on attribute meets minimum threshold. Processing continues on the remaining header items of original FP-tree. Once recursive process is complete and all large item sets with minimum coverage is found, association rule creation begins [7,8,26].

To break the two drawbacks [27] of other models, FP-growth is used and it requires constructing an FP-tree, which implements the divide-and-conquer strategy that scans the database twice. During the first database scan, it first computes a list of frequent itemset sorted in descending order (F-List); And on the second scan, the database is compressed into a FP-tree [28]. Thus, the model performs mining on FP-tree recursively. Its demerit is that in finding frequent itemsets, it searches and constructs trees recursively. Its merits include: (a) frequent itemsets are generated with two passes over the databases; (b) there are no candidate generation process. Instead, the 2-sub-processes of frequent patterns generation process include construction of the FP-tree, and generation of the frequent patterns from the FP-tree; (c) if and when counters are incremented, some of its pointers are maintained between nodes to contain the same item, by creating singly linked lists; (d) the more the paths that overlap, the higher the compression, and (e) FP-tree may fit in memory as frequent itemsets are extracted from the FP-Tree [29]. FP is constructed over dataset using 2 passes as thus:

Pass 1:
1. It scans the data and find support for each item.
2. Discard infrequent items.
3. Sort frequent itemset in descending order based on their support and builds FP-tree such that common prefixes can be shared.

Pass 2:
1. Here nodes correspond to items and it has a counter.
2. It reads one transaction at a time and maps it to a path.
3. Fixed order used allowing paths to overlap if transactions share the items.

Pseudocode is as thus:

Procedure FP-Growth (Tree *T*, *A*) {
If tree *T* contains a single path *P*
Then For Each combination of the nodes in the path *P* DO
generate pattern *B* U *A* with support = minimum support nodes in B
Else For each H_i in the header of the Tree *T* DO
{
Generate pattern B = Hi U A with support = Hi support:
Construct B's conditional pattern base and B's conditional FP-tree that is the Tree B;
Then call FP-Growth (Tree B, B) }
}

3. Deep Learning Neural Network

Deep neural network has successfully been implemented in systems that seek to learn useful features and construct multi-layer networks from a vast amount of training data. Forecast accuracy is improved using DNNs, allowing more information regarding the raw dataset to be obtained. DNN has deep architectures including multiple hidden layers—and each hidden layer alone conducts a non-linear transformation from the previous layer to the next [30,31]. Through deep learning, proposed by Hinton et al. [32], a DNN is trained according to two sections namely: (a) pre-trained, and (b) fine-tuned procedures [33].

3.1. Auto-Encoder

Ma et al [21] The Auto-Encoder is a type of unsupervised three-layered neural network whose output target is an input data shown in Figure 1. The Auto-Encoder includes both encoder and decoder networks. The encoder network transforms input data from a high-dimensional space into a low-dimensional space; While, the decoder network remodels the input from the previous step. The encoder network is defined as encoding function $f_{encoder}$ – with encoding process as in Eq. 1 – where x^m is a data point and h^m is the encoding vector obtained from x^m:

$$h^m = f_{encoder}\left(X^m\right). \tag{1}$$

The Decoder network is a reconstruction function denoted as $f_{decoder}$ and is described as below, where x^m is the decoding vector obtained from h^m from Eq. 2. Other specific algorithms for encoding and reconstruction functions includes Eq 3 to Eq. 5 respectively. See Ma et al [21] for more details and approaches on sparse auto-encoders, denoising auto-encoders and more.

$$X^m = f_{decoder}\left(h^m\right) \tag{2}$$

$$logsig: f_{encoder}\left(X^m\right) = \frac{1}{1+e^{-X^m}} \tag{3}$$

$$satline: f_{encoder}\left(X^m\right) = \begin{Bmatrix} 0 & if\ X^m \le 0 \\ z & if < X^m < 1 \\ 1 & if\ X^m \ge 1 \end{Bmatrix} \tag{4}$$

$$pureline: f_{encoder}\left(X^m\right) = X^m. \tag{5}$$

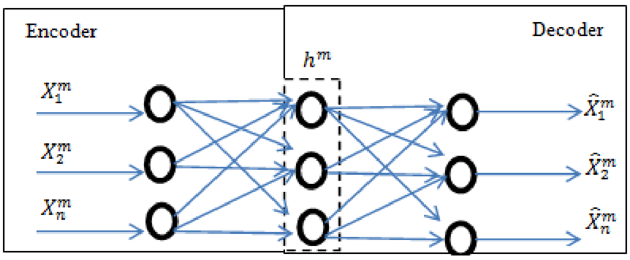

Figure 2. Architecture of an Auto-Encoder and Decoder in Deep Neural Network

3.2. Pre-Training

N auto-encoders can be stacked to pre-train an N-hidden-layer DNN. When given an input dataset, the input layer and the first hidden layer of the DNN are treated as the encoder network of the first auto-encoder. Next, the first auto-encoder is trained by minimizing its reconstruction error. The trained parameter set of the encoder network is used to initialize the first hidden layer of the DNN. Then, the first and second hidden layers of the DNN are regarded as the encoder network of the second auto-encoder. Accordingly, the second hidden layer of the DNN is initialized by the second trained auto-encoder. This process proceeds until the Nth auto-encoder is trained to initialize the final hidden layer of the DNN. Thus, all hidden layers of the DNN are stacked in an auto-encoder in each training N times, and are regarded as pre-trained. This pre-training process is proven to be significantly better than random initialization of the DNN and conducive to achieving generalization in classification [32,34].

3.3. Fine-Tuning

This is a supervised process that improves the performance of a DNN. The network is retrained, training data are labelled, and errors calculated by difference between real and predicted values are back-propagated using stochastic gradient descent (SGD) for all multi-layer networks. SGD randomly selects data samples, and iteratively updates gradient direction with weight parameters. Best gradient direction is obtained with a minimum loss function. The merit of SGD is that it converges faster than traditional gradient descent methods do, and does not consider the entire dataset, making it suitable for complex neural networks [21] with SGD equation as below:

$$E = \frac{1}{2}\sum_{j=1} M\,(y_i - t_i)^2 \quad (6)$$

where E is the loss function, y is the real label and t is the network output. The gradient of weight parameter w is obtained by the derivative of the error equation – so that with the gradient of the weight w_{ij}, the updated SGD equation is defined by Eq. 5 below:

$$W_{ij}^{new} = W_{ij}^{old} - \eta\,.\left(y_j - t_j\right).y_j\left(1 - y_i\right).h_i \quad (7)$$

where η is the step-size and it is greater than 0, h is number of hidden layers in the DNN. This process is tuned and optimized by the weights and threshold based on the correctly labelled data in the DNNs. Thus, the DNN can learn important knowledge for its final output and direct the parameters of entire network to perform correct classifications.

4. Materials and Method

This paper presents an overview of association rule mining algorithms. Algorithms are discussed with proper example and compared based on some performance factors like accuracy, data support, execution speed etc.

4.1. Motivation / Statement of Problem

Despite the numerous benefits offered by mobile computing, performing market basket analysis still faces big challenges particularly with the concept of drift and feature evolution.

1. Classifying transactional data pose issues to data mining with concept drift, concept evolution, infinite length and feature evolution. Theoretically, a data stream is infinite in length – making it impractical to store as well as utilize such historical data for training; A naive solution will seeks to ascertain item pairs in each market basket, store assiduously and then run an algorithm finding the most frequent pairs. Such solution however, is impractical as many market baskets containing billions of different item pairs (combination). It is plausible to expect that certain pairs are much more popular than others. Thus, the need to employ stochastic modeling.
2. Previous research on transactional data makes stationary assumption through training cum testing associative rule mining algorithm on observed dataset acquired from the same population. This deprives the model of the needed flexibility to adapt to non-existent data that were not in the database from the outset and the robustness required to handle feature evolution and concept drift inherent in transactional dataset streams.
3. Ever increasing need and demand for portable computing continues to necessitate development of Market Basket models for such platforms with its major consideration as limited speed and memory of such computing.

Thus, we seek the computational proposed model-solution for hybrid associative rule mining for market basket analysis to resolve the challenges of feature evolution and concept drift; And, thus ensure the effective classification of transactional data stream to handle the issues of big-data. This is because as data are generated and stored – the model needs to be flexible enough to adapt to new data itemset generated on-the-fly, and robustness to ensure the model is capable of being readily and easily reused on such dataset with no alterations to the system.

4.2. Rationale for the Study

Most mathematical, machine learning models are inspired by evolution, biological and behavioural population. They search a space through hill-climbing method which is flexible, easily adapted to changing states and suited for real-time app – such that model guarantees high global convergence in multimodal task. Initialized with random pool, it allocates increasing trials to regions of high fitness to find optima. Once a peak is found, model restarts with another randomly chosen, starting point. Its simplicity, well suited for dynamic feats/phenomena of many local maxima – makes them appropriate. Each random trial is done in isolation and as search progresses, it allocates its trials evenly over space and still evaluates as many points in regions found to be of low fitness as in regions found to be of high fitness. Its demerit is its inadequacy for linear model with small regions surrounded by low fitness – making such functions/models, difficult to optimize.

5. Experimental Proposed Framework Solution

5.1. The DNN Framework

Several classification engines (BPNN, SVM, RF, Bayes and DNN) perform well given the advantages of their algorithms, which can efficiently handle complex classification problems; hence, these models can be successfully applied to intrusion detection. They also usually perform poorly when facing the complex randomness and camouflage of network intrusion dataflow. Therefore, in this section, the proposed approach is employed to solve the above problems. First, the training data subsets divide the training process and calculate centre points by SC from each training point. Second, each training data subset is trained by the corresponding DNNs, where the scale of DNNs is the same as the number of clusters. In this way, the DNNs have learned different characteristics from each subset. Third, the testing data subsets are divided into test datasets by SC, which uses the previous cluster centres in its first step, and these subsets are applied to detect intrusion attack types by pre-trained DNNs. Finally, the output of every DNN is aggregated for the final result of the intrusion detection classifiers.

The proposed model-based solution is divided into 3-steps:

1. **Step 1**: Dataset is divided into training and testing groups. The training dataset is clustered by the basket and this output is regarded as training subset. The cluster centres from the training dataset clustering process are stored to serve as initialization cluster centre for generating the testing dataset clusters. Because basket data feats indicate similar attributes of each type in the raw dataset, points in the training dataset with similar features are aligned into groups and regarded as same subset. In order to improve DNN-model for each approach, the performance, different cluster numbers and values of sigma are considered. The number of clusters ranged from 2 to 6 and sigma from 0.1 to 1.0. Samples are assigned to one cluster by similarity. The minimum distance from a data point to each cluster centre is measured in algorithm. Each point is assigned to a cluster. Training subsets generated by clusters are given as input to DNNs. In order to train the different subsets, the number of DNNs is equal to the number of data subsets. The architecture of each DNN consists of five layers: two hidden, one input, one softmax and one output layer(s) respectively. Two hidden layers learn feats from each training subset and top layer is a five-dimensional output vector. Each training subset which are generated from the kth cluster centre in clustering process, regarded as input data to feed into *k*th DNN, respectively. These trained sub-DNN models are marked sub-DNN 1 through sub-DNN *k*.
2. **Step 2**: Testing dataset (subset of raw dataset) is used to generate *k*-datasets. The previous cluster centres obtained from cluster in Step 1, are initialization cluster centres of the cluster algorithm in this step. The test sub-dataset are denoted as Test 1 through Test k.
3. **Step 3**: The *k*-test data subsets are fed into k sub-DNNs, which were completed by the k training data subsets in Step 1. The output of each sub-DNN is integrated as the final output and employed to analyse positive detection rates. Then, confusion matrix is used to analyse mining performance of generated rules.

5.2. The DNN Algorithm

Our proposed DNN model adopts a deep learning approach to categorize market basket data; to fine-tune its weights and thresholds using Backpropagation learning. The input vectors map low-dimensional space with DAEs and SAE [21] to discover patterns in the market basket datasets. The DNN algorithm is detailed in Algorithm below.

Input: dataset, cluster number, number of hidden-layer nodes HLN, number of hidden layers HL.
Output: Final prediction results
/*Note the symbols of "/_" and "_/" represent comments in this algorithm.*/

1. Divide raw dataset into two components: training and a testing dataset.
   /*get the largest matrix eigenvectors and training data subsets*/
2. Obtain the cluster centres and cluster results. Here, the clustering results are regarded as training data subsets.
   /*Train each DNN with each training data subset*/
3. The learning rate, denoising and sparsity parameters are set and the weight and bias are randomly initialised.
4. HLN is set 40-nodes for first and 20-nodes for second hidden layer.
5. Compute sparsity cost function
6. Parameter weights and bias are updated
7. Train k sub-DNNs corresponding to the training data subsets.
8. Fine-tune the sub-DNNs by using backpropagation to train them.
9. Final structure of trained sub-DNNs is obtained and labelled with each training data subset.
10. Divide test dataset into subsets with SC. Cluster centre parameters from the training data clusters are used.
11. Test data subsets are used to test corresponding sub-DNNs, based on each corresponding cluster centre between the testing and training data subsets. /*aggregate each prediction result*/
12. Results are generated by each sub-DNN, are integrated and the final outputs are obtained.
13. **return** classification result = final output

6. Result Findings and Discussion

6.1. Classification Accuracy

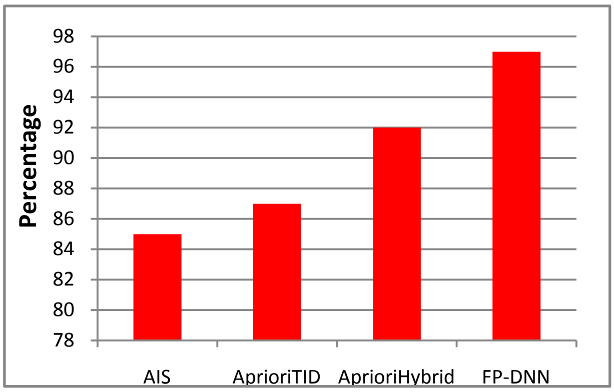

Figure 3. Prediction Accuracy of Basket Dataset Clustering with model-based preprocessor trained DNN Algorithm

6.2. Processing Speed

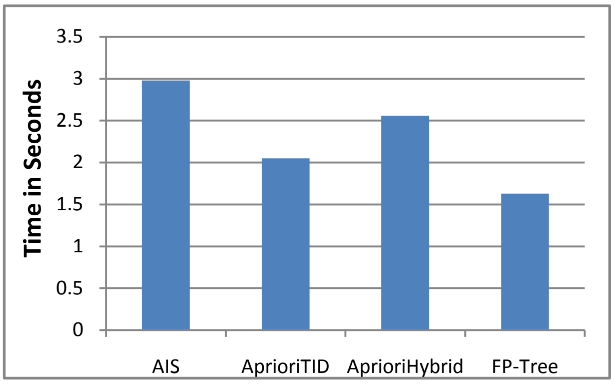

Figure 4. Mean Processing Time in Seconds of Basket Dataset Clustering with model-based preprocessor trained DNN

6.3. Convergence Time

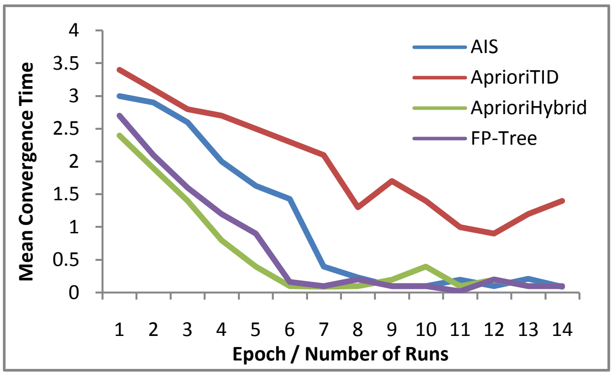

Figure 5. Convergence time of matches

6.4. Discussion of Findings

The proposed model-based solution as preprocessor and its fusion with DNN model – a total of 56-rules were generated for each solution approach. Top rules were found to have fitness range [0.8, 0.865] and are estimated 80% good to be used in classification of market basket clustering dataset. This implies that achieving a set of good rules – is much better than single optimum rule, which in turn is better for such clustering dataset. 22-out-of-56 rules have profile for candidate itemset 3 and above, so that the rules search for groups of 3 and above itemset. This in turn, increases chances of detecting basket data as well as also improves the generality of rules, providing the ability for new itemset and corresponding generated rules – to be added to the knowledgebase. The rule generator used a population of 400, w1 = 0.2, w2 = 0.8, 5000 epoch-evolutions and 0.05 probability of a gene to be mutated respectively.

6.5. Implementation Tradeoffs

Result trade-offs are as follows [6]:

a. Results: Researchers often display flawed and unfounded results, to validate new/modified model; Rather, than re-test limitations, insufficiency, biasness and inabilities of existing ones. Negative results are less valuable and most of such models aim to curb non-linearity and dynamism in the task they are predicting alongside discovering feats and underlying properties of the historic datasets used, to train, cross validate and test such models.

b. Efficiency: Researchers can often use figure to show how well their prediction is in agreement with observed values (even with their limited dataset used for training a model that is often times squeezed). Some plot for observed and predicted values are often not easily distinguishable – as such modelers do not even provide numerical data to support their claim for their system (though their model is in good agreement with observed values). Some measure of goodness does not provide the relevant data.

c. Insufficient Test: Validation compares observed versus predicted values. Many studies suffer from inadequate dataset. If model aims to predict dynamic state, such ability should not be demonstrated with misleading results of limited dataset; and inconclusive and unclear contributions. Model must be adequately tested with methods laid bare so that process can be repeated to validate the usefulness and authenticity of such models.

7. Conclusion

Association rule mining plays an important role in market basket analysis by finding frequency of occurrence of items in a transactional dataset. Through use of ARM, every business organization aims at maximizing the social relationship with the customer.

References

[1] Awodele, O., Izang, A.A., Kuyoro, S Osisanwo, F. (2016). Big Data and Cloud Computing Issues. *International Journal of Computer Applications*, *133*(12), 0975-8887.

[2] Annie, L and Kumar, A., (2012). Market basket analysis for a Supermaket Based on Frequent item Mining. *International Journal of Computer Science, IX*(5), pp 34-43. Retrieved January 10, 2017.

[3] Agrawal, R., Imielinski, R and Swami, H., (1993). *Mining association between sets of items in massive databases*, In Proc. of ACM SIGMOD Conference on Management of Data, *pp 207-216*. Chicago, Washington DC, USA.

[4] Agrawal, R and Srikant, R. (1994). *Fast algorithms for mining association rules in large databases*, Proc. of 20th Int. Conf. on Very Large Databases, pp.487-499). Retrieved January 23, 2017.

[5] Kumbhare, T.A and Chobe, S., (2014), *An overview of association rule mining algorithms*, International Journal of Computer Science and Information Technology, 5(1), pp 927-930, ISSN: 0975-9646.

[6] Ojugo, A.A., Yoro, R.E & Okobah, I.P., (2018). A *comparative study of market basket analysis rule mining approaches*, Technical Report of the Federal University of Petroleum Resources Effurun (FUPRE-TRON-1023), pp 312-337.

[7] Girotra, M., Nagpal, K., Inocha, S and Sharma, N., (2013). *Comparative Survey on Association Rule Mining Algorithms*, International Journal of Computer Applications (0975-8887), 84(1) – No 10.

[8] Han, J., Pei, H and Yin, Y., (2000). *Mining frequent patterns without candidate generation*, Conference on the Management of Data, pp 310-314.

[9] Vijayata, W., & Debajyoti, M. (2014). Mobile Agent based Market Basket Analysis on Cloud. *India. 2014 International Conference on IT.* (pp. 49-62). Pune, India: Department Of Information Technology, Maharashtra. Institute of Technology Pune, India.

[10] Woo, J. (2015). Market Basket Analysis using Spark. *ARPN Journal of Science and Technology, V*(4), 207-209.

[11] Jason, B., (2013). A Tour of Machine Learning Algorithms. *Journal of Intelligent Learning Systems and Applications*, *9*(4), 50-62.

[12] Kamalraj, R., Kannan, A., Vaishnavi, S., & Suganya, V. (2012). A Data Mining Based Approach for Introducing Products in SaaS (Software as a Service). *International Journal Of Engineering Innovation & Research, I*(2), 210-214.

[13] Ming-Syan, C., Han,J and Yu, P.S., (2016). *Data mining: an overview from a database perspective*, IEEE Transactions on Knowledge and Data Engineering, Volume:8, Issue: 6 ISSN: 1041-4347, 866-883.

[14] Dhanabhakyam, M and Punithavalli, M. (2011). *A Survey on Data Mining Algorithm for Market Basket Analysis*, *Global Journal of Computer Science and Technology, XI*(11), 13-21.

[15] Marafi, S. (2014). *Market Basket Analysis with R*. Retrieved October 16, 2016, from Marafi Salem Blog: http://www.salemmarafi.com/code/market-basket-analysis-with-r/.

[16] Rosaria, S. (2015). *How to build a Market Basket Analysis Engine*. Retrieved from Analyticbridge: http://www.analyticbridge.datasciencecentral.com/profiles/blogs/how-to-build-a-market-basket-analysis-engine.

[17] Chiemeke, S and Ojugo, A.A., (2018), *Inventory management and prediction using the market basket analysis associative rule mining: memetic algorithm based approach*, Accepted paper in the Journal of Computer Science Application – Nigerian Computer Society, 56(4), pp 21-34.

[18] Ojugo, A.A., (2018). *Unsupervised cluster prediction model for market basket analysis: quest for better alternatives to association rule mining approach*, In Proc. of the 21st ISTEAMS Conf. on Intelligent Systems and Management of Data, pp121-135. Retrieved September 24.

[19] Maheshwari, N., Pandey, N. K and Agarwal, P., (2016). Market Basket Analysis using Association Rule Learning. *International Journal of Computer Applications*, 0975-8887. Retrieved Feburary 12, 2017, from www.ijcaonline.org.

[20] Meherwar, F and Maruf, P., (2017). Survey of Machine Learning Algorithms for Disease Diagnostic. *Journal of Intelligent Learning Systems and Applications*, 1-16.

[21] Ma, T., Wang, F., Cheng, J., Yu, Y and Chen, X., (2016). *A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks*, Sensors, 16, pp1701.

[22] Komal Khurana, Mrs. Simple Sharma, *A Comparative Analysis of Association Rules Mining Algorithms*, International Journal of Scientific and Research Publications, Volume 3, Issue 5, May 2013 ISSN 2250-3153.

[23] Bhalodiya, D., Patel, K and Patel, C., (2013). *An efficient way to find frequent pattern with dynamic programming approach*, *Nirma University International Conference On Engineering, Nuicone-2013*, pp 45-56.

[24] Prithiviraj, P and Porkodi, R., (2015). *A comparative analysis of association rule mining algorithms in data mining*, American J. of Computer Science and Engineering Survey, 3(1), pp 98-119.

[25] Agarwal, R., Aggarwal, C and Prasad, V., (2001). *A tree projection algorithm for generation of frequent itemsets*, Journal of Parallel and Distributed Computing, pp 350-371.

[26] Shendge, P and Gupta, T., *Comparitive Study of Apriori & FP Growth Algorithms*, Indian journal of research, Volume 2, Issue 3, March 2013.

[27] Sotiris Kotsiantis, Dimitris Kanellopoulos, *AssociationRules Mining: A Recent Overview*, GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pp. 71-82.

[28] Gagandeep, K and Aggarwal, S., (2013), *Performance Analysis of Association Rule Mining Algorithms*, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, ISSN: 2277 128X.

[29] Parita Parikh, Dinesh Waghela, *Comparative Study of Association Rule Mining Algorithms*, *Parita Parikh* et al, UNIASCIT, Vol 2 (1), 2012, 170-172, ISSN 2250-0987.

[30] Schmidhuber, J., (2015). *Deep learning in neural networks: An overview*, Neural Network. Vol. 61, pp 85-117.

[31] Bengio, Y., Courville, A & Vincent, P., (2013). *Representation learning: A review and new perspectives.* IEEE Transaction on Pattern Anal. Mach. Intelligence, Vol. 35, pp 1798-1828.

[32] Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P & Sainath, T.N., (2012). *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups*. IEEE Signal Process. Mag., Vol. 29, pp 82–97.

[33] Glorot, X and Bengio, Y., (2010). *Understanding the difficulty of training deep feedforward neural networks*, In Proc. of 13th International Conf. on Artificial Intelligence and Statistics (AISTATS'10), Sardinia, Italy, 13-15 May 2010; pp. 249-256.

[34] Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P and Bengio, S., (2010). *Why does unsupervised pre-training help deep learning?*, Journal of Machine Learning Res., Vol. 11, pp. 625-660.

[35] Borgelt, C. (2014). *Find frequent itemsets and association rules with the Apriori Algorithm.* Retrieved from C. Borgelt's home page: http://www.borgelt.net/doc/apriori/apriori.html.

[36] Gagandeep, K and Aggarwal, S., (2013), *Performance Analysis of Association Rule Mining Algorithms*, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, ISSN: 2277 128X.

[37] Ish Nath Jha Samarjeet Borah, *An Analysis on Association Rule Mining Techniques*, International Conference on Computing, Communication and Sensor Network (CCSN) 2012.

[38] Ojugo, A.A and Yoro, R.E., (2018). *Intelligent lightweight market basket associative rule mining for smartphone cloud-based application to ease banking transactions*, In Proc. of the 21st ISTEAMS Conf. on Intelligent Systems and Management of Data, pp 23-37.

[39] Qiankun Zhao, Sourav S. Bhowmick, *Association Rule Mining: A Survey*, Technical Report, CAIS, Nanyang Technological University, Singapore, 2003.