INCREMENTAL LEARNING FROM UNBALANCED DATA WITH CONCEPT CLASS, CONCEPT DRIFT AND MISSING FEATURES: A REVIEW

Pallavi Kulkarni and Roshani Ade

Dr. D. Y. Patil School of Engineering and Technology,
Savitribai Phule Pune University, India

ABSTRACT

Recently, stream data mining applications has drawn vital attention from several research communities. Stream data is continuous form of data which is distinguished by its online nature. Traditionally, machine learning area has been developing learning algorithms that have certain assumptions on underlying distribution of data such as data should have predetermined distribution. Such constraints on the problem domain lead the way for development of smart learning algorithms performance is theoretically verifiable. Real-word situations are different than this restricted model. Applications usually suffers from problems such as unbalanced data distribution. Additionally, data picked from non-stationary environments are also usual in real world applications, resulting in the "concept drift" which is related with data stream examples. These issues have been separately addressed by the researchers, also, it is observed that joint problem of class imbalance and concept drift has got relatively little research. If the final objective of clever machine learning techniques is to be able to address a broad spectrum of real world applications, then the necessity for a universal framework for learning from and tailoring (adapting) to, environment where drift in concepts may occur and unbalanced data distribution is present can be hardly exaggerated. In this paper, we first present an overview of issues that are observed in stream data mining scenarios, followed by a complete review of recent research in dealing with each of the issue.

KEYWORDS

Incremental learning, class imbalance, concept class, concept drift, missing features

1. INTRODUCTION

There are various data mining techniques that can be used for analysing problems in real world applications and discovering their solution in a scientific manner. Supervised learning techniques use instances, which have already been pre-classified in some manner. That means each instance has a label, which recognizes the class to which it belongs. Classification is a supervised data mining technique, predicts about data values, using already known results found from data gathered in advance. Classification maps, data into groups of classes established in advance. In short, it tells us which data instance, should belong to which category of data, thereby simplifying the processing of large amounts of data, such as stream data or data from decision support systems. Data evolves over time, as time goes on and rapid increase in volume of data is observed [75; 76].

Conventionally, the supervised learning method attempts to group the data from a static dataset, the instances of which are related to the basic distribution described by a generating function.

DOI : 10.5121/ijdkp.2014.4602

Hence it is presumed that the dataset contains all required information for learning the pertinent concepts. However, this technique has proven impractical for many real world applications, e.g., spam identification, climate monitoring, credit card fraud detection, etc. Data is usually received over time in streams of instances and is not at hand from the beginning. Such data conventionally reaches there in two ways, first is incremental arrival or batch arrival. So the question is to use all the information up to a certain time step t to estimate up to the minute instances coming at time step t+1. Learning in such cases is referred to as incremental learning [35].

If learning from enormous data which is progressing by sequential steps is desired, then incremental learning or online algorithms are best suitable and preferred. Incremental learning algorithm is always fed with input data as the data arrives in sequence, the algorithm computes hypothesis by taking into account order of data arrival, each one gives details of all the data seen so far. Note that the algorithm should be depend on prior hypotheses and the ongoing training data [66].

Incremental learning algorithm should possess all the properties mentioned below:

1. The algorithm should gain knowledge which is additional in recent data.
2. To train already present classifier, the algorithm need not access to the initial data
3. It should keep up the knowledge which it has previously learned
   (No catastrophic forgetting should occur)
4. New data may bring in concept class. The algorithm should be able to recognize and accommodate such a new class which can be introduced over the period of time. [1]

This definition points out that, learning from stream data needs a classifier that can be incrementally modified to get full benefit from newly arrived data, while simultaneously preserving performance of the classifier on matured data underlining the fact to of "stability-plasticity" dilemma which describes how a learning system can be designed to stay stable and unchanged to immaterial affairs , while plastic (i.e. be able to change when necessary in order to deal with different situations) to recent and crucial data (e.g. concept change) .

Hence, the stability-plasticity dilemma poses a cline on which incremental learning classifiers can prevail. Batch learning algorithms (i.e. system of algorithms trained on static data) are on one end of the cline which is representation of stability. These algorithms overlook al new data, instead is aimed fully at formerly learned concepts.

Another end of the cline contain online learning algorithms where the model is altered upon observing the new instance and the instance is discarded immediately.

Even if batch learning systems prevail on one of the end of cline of stability-plasticity dilemma, they are fundamentally not incremental in nature. Because batch learners do not have the capacity of describing any new instances once they have been learned and thus fail to fulfill property 1 of incremental learning. This restraint is alleviated by building ensembles of batch learners, where fresh batch learners can be learned on the fresh data, and then united through a voting system [28; 74].

Ensemble technique (multiple classifier system) is widespread in the area of machine learning, specifically in the incremental environment. An ensemble technique is extracted by merging diverse classifiers. There are various differentiating parameters who help to achieve diversity that in turn entitles each classifier to produce several decision boundaries. Appropriate diversity allows to gain different errors to be made by individual classifier and finally strategic integration of them can cut off the total error in the entire system. The Ensemble can be build up in several ways like:

1) Bagging
2) Boosting – Adaboost is pretty popular algorithm
3) Stacked generalization
4) Mixture of experts

The diversity need can be fulfilled by applying various approaches such as:

1) Training each classifier using several data chunks
2) Training each classifier using several parameters of a given classifier architecture
3) Training each classifier using several classifier models
4) Random Subspace method (training each classifier using several subset of characteristics)

Detail literature survey on the classifier combination approach can be found in Kuncheva's book [83].

Ensemble approach can be used to address every challenge associated with incremental learning as a basic design model. Several solutions for those issues are found that are inherently combined with ensemble systems [78-83].

2. ISSUES OF DATA STREAM LEARNING – CONCEPT CLASS

One of the most important work in incremental learning includes Learn++ group of algorithms. The initial algorithm from Learn++ family is Learn++ which is inspired by Adaboost[66]. Similarities between them are:

1) Sequential generation of classifier ensemble

2) Training data contains bootstrapped samples which are taken from distribution

3) Weight Distribution is altered iteratively

The main difference between them is in the working of distribution update rule, where this rule is better in terms of performance in case of Learn++ as the distribution moves step by step towards the new instances that have not been appropriately learned by the present ensemble in the system. Classifiers are united through weighted majority voting technique, where voting weights are computed by relative performance (in terms of minimum error) of each classifier on instances of training data. If new data arrives, Learn++ creates additional ensemble to learn novel information from it. Learning concept classes need large number of classifiers (referred to as problem of classifier proliferation) for each novel class to be learned in Learn++. So it is possible to learn new concept classes in Learn++ but it does so with added cost [1].

The interesting quality in Learn++. NC is a novel classifier combination technique that lets distinctive classifiers to communicate with each other to decide their voting weights for each test example. Learn++. NC applies different set of rules for updating initialization of the algorithm when recent data comes. Learn++.NC enquire each specific classifiers to cross check their results with respect to classes on which training was given to them. By observing the predictions of another classifiers, each classifier determines whether its result is in line with the classes others are deciding, and the training classes. If this is not the case, then the classifier cut down its vote, or possibly desist from voting all together with all other classifiers in the ensemble system. Hence problem of out-voting is frankly discussed and addressed in Learn++.NC (*N*ew *C*lass), through a special mechanism referred to as dynamic consult and vote [28]. Learn++.NC cannot address the infrequent but difficult to control process of adding new classes and at the same time discarding former ones on a next dataset Learn++.UDNC is an improvement over Learn++.UD [64] and scrounges confidence measure mechanisms from Learn++.NC. It is discussed in next sections of the paper [52].

Gregory et al. proposed a smart hybrid algorithm which is part of Learn++ family and referred to as Learn++.NCS (New Classes with Sampling).It uses a SMOTE as wrapper to make process of learning easier of a minority classes even when former/later classes are discarded/added sporadically in the interval of a testing. The wrapper can be applied to other learning algorithms than Learn++.NC. In other words, Learn++.NCS uses Learn++.NC for the incremental learning of concept classes along with a SMOTE-based sampling wrapper to handle class imbalance [65].

He et al. put forward a universal adaptive incremental learning framework called ADAIN which is having capacity to learn from stream (continuous) data, amassing knowledge over time, and employing such experience to enhance later learning and prediction (classification) performance. ADAIN framework has capability to handle and detect single or multiple new concept classes. It considers input data stream as raw data on which processing has to be done by the distribution function and finally hypothesis will be generated. This framework also includes a mapping function whose aim is to stipulate a quantitative judgment of the learning proficiency of the recent piece of data (note that the size of this piece of data will have crucial influence on the eventual learning results) based on formerly trained classifier. Additionally this mapping function also referred to as non-linear regression model, puts up a link from previous wisdom to the recently accepted data, and adjusts (i.e. quality of being adaptable) such experience to data sets received later. In this paper, two major design strategies are simulated of the given framework known as ADAIN.MLP and ADAIN.SVR. They are distinguished by the basic implementation of the model they are using inside the mapping function. By observing performance of ADAIN comparatively with other existing approaches, Learn++ algorithm is the most assertive one. Learn++ is set on the "Ensemble-of-Ensemble" technique. Due to use of this "Ensemble-of-Ensemble" policy, Learn++ tends to acquire much more computational resources. Additionally, when the data set is of huge size, the required time of Learn++ increases exponentially, which restricts its ability to scale and ability to tackle large-scale stream data [86-87].

Therefore, we can say that ADAIN achieves the goal of incremental learning as it successfully unites formerly learned experience into recently accepted data to enhance learning from current raw data, and in the meantime it accumulates such knowledge over time to encourage decision-making in future. However it does not address issues of concept shifting and class imbalance [69-73].

Several classification simulations can be used as base classifiers in multiple classifier system. Each classifier gets a chunk of data distribution in the form of some signals or we can call it as a pattern.

This pattern describes how the system is going to operate and once applied to the ensemble of classifiers can generate output in the form of fault. By considering his idea, Roozbeh et al.[68] proposed a new method where the MultiLayer Perceptron (MLP) from Neural Networks[67] are used as the base classifier in the ensemble system which can in turn be used for training of ensemble and testing of unseen instances further. Three layer MLP are employed because they are able to differentiate between complex and nonlinear decision boundaries and also are adaptable of behaving like weak learners by sufficient parameter updation mechanism. Here, the detection of unknown classes in future data is computed by thresholding the normalized weighted average of outcomes (NWAO) of the base classifiers in the ensemble system. Noticing of new fault classes is crucial in applications such as nuclear industry systems. For that purpose, threshold analysis with NWAO is proposed which sets some limits on the values and thereby simplifies the task of detecting fault classes in nuclear power system application. Class assignment is performed with the help of DW-CAV routine of Learn++.NC algorithm [28] in this work.

3. ISSUES OF DATA STREAM LEARNING – CONCEPT DRIFT

Data is always generated by some function. Conventional system of data mining algorithms assumes that each dataset is produced from a specific, static, hidden function. It means that uniform function is used for training and testing of the data. This assumption may fail in case of stream data, i.e. the function which produces instances at time step t may not be the identical function as the one that creates instances at time step t+1 (next time instance).This possible variation in the underlying function is referred to as concept drift. In other words, concept drift might be viewed in a more abstract sense as a hindrance caused by inadequate, unknown or unobservable characteristics in a dataset, an event called as *hidden context* [24]. Here, the underlying phenomenon which lends a real and static picture over time for each class is regrettably concealed from the learner's vision. New data is generated by some hidden function and learner is not aware of this fact and therefore the concept drift is unpredictable. If the generating function for the drifting concepts was already recognized, one could simply learn a suitable classifier for each pertinent concept, and use the appropriate classifier for all recent data (which is called as the multitask learning problem). In the unavailability of such knowledge, then, we should configure a unified classifier that is able to acquire such swings in concepts over time. Diagram below shows the relationship among concept drift, model adaptation, knowledge (wisdom) sign-over and time series analysis. It is quite clear from the figure that all this features are dependent on time.

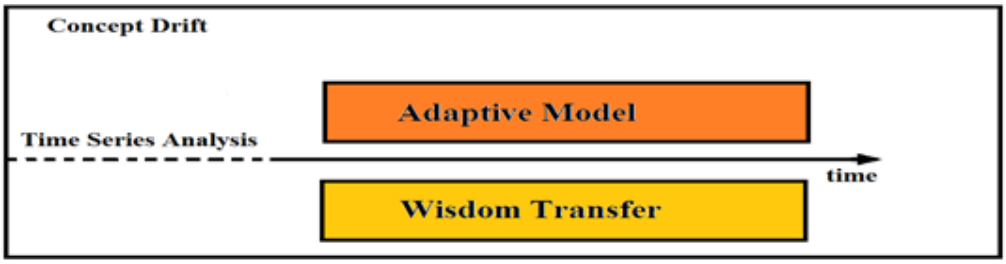

Figure 1. Relation of concept drift with its features

3.1 Formal definition of Concept drift

Concept drift occurs in an environment which is commonly referred to as Non stationary environment. In such cases we assume that at time step t, the algorithm A is presented with a set of labeled instances$\{X_0,\ldots,X_t\}$, where X_i is a v-dimensional vector and each instance has corresponding class label y_j. If an unlabeled instance comes at time t+1 as X_{t+1}, then the algorithm is expected to provide class label for X_{t+1}. This is a predicted label of given instance. Once this is done, the real label Y_{t+1} and a new testing instance X_{t+2} comes so that we can go for it's testing. Moreover, we call the hidden function f_h producing the instance at time instant t as f_t. *Concept drift* is said to come up when the underlying generating function (f_h) alters over time. There are many styles in which this alteration can take place. Bayes' theorem can calculate the probability that instance Xt_{+1} belongs to class *ci* as:

$$P(Ci|Xt+1) = \frac{P(ci)\,P(Xt+1|Ci)}{P(Xt+1)}$$

This well- known theorem gives relationship between prior and posterior probabilities of instances and their classes.

Concept drift can take place with respect to any of the three main variables in Bayes' theorem

1) p(ci) can alter(prior class distribution)
2) p(Xt+1|c) can alter (class distribution)
3) p(c|Xt+1) can alter (posterior class distribution)

There are also several forms of concept drift which can be categorized into two types as, real concept drift and virtual concept drift.[22,23] If the distribution of instances is altering corresponding to alteration in the class prior probabilities or change in class distribution is taking place then virtual concept drift is said to occur. In this case underlying concept means posterior distribution of instances remains same as before. This may create problems for the learner, as such updates in the probability distribution may alter the error of the learned system, even if the concept did not alter. Some parts of target concepts may have gone unobserved by the learner, because of alteration of distribution such instances may become more common Sine the learner never seen such data, it could not learn the concept and hence should be retrained. This kind of virtual drift is particularly applicable when the data stream displays unbalanced classes.

Also, real concept drift can be interpreted as a modification in the class boundary (i.e. change in the posterior distribution). This alteration indicates more fundamental change in the generating function.

Note that real concept drift need alteration in the model and virtual concept drift also require the same. This leads to the fact that final result is same. Moreno-Torres et. al. [21] stipulate clearer survey of various forms of "dataset drift", as well as their Sources.

Quinonero-Candela [25], Minku [26], and Kuncheva [27], [28] stipulated complete summaries for featuring several types of concept drift with respect to its *speed*, *randomness*, and *cyclical* nature. Drift speed is the displacement rate in *prior probability of an instance* from one time step to the later one. Substantial displacement within a step represents fast (rapid) drift and usually results in high classifier error. Gradual (slow) drift, however, sounds in compact displacements which in turn leads to lower classification error, and as a result, is harder to notice.

Drift randomness is a significant descriptor in being able to show good judgment about quality between non-stationary and noisy data, and can be illustrated as the change in a distribution over a short spell of time. Randomness can be viewed in terms of its frequency and magnitude: high variance between two spells of time denotes a highly unstable environment which, as this level raises, reaches a condition where the environment cannot gain new knowledge. The cyclical nature of drift is an event that can be seen in different real-world applications such as weather forecasting, climate monitoring, nuclear power claim modeling. In such situations, class definitions alter in such a way that a last settings of the environment may happen again or number of times (i.e. recur) after some spell of time. Recurrence which can be seen in such environments is one of a two kind as periodic or random. Concept drift learning systems do not address the problem of concept class i.e. addition or removal of new classes; as such phenomenon are featured more by Concept change.

Therefore speaking in the more general terminology such *learning in nonstationary environments* or *nonstationary learning* (NSL) is to refer to any drift or change without considering its kind and nature. [2]

There are some general guidelines to shape up a system for learning in non-stationary environments.

1) Necessity of truly incremental or one pass learning where access to previous data is strictly not allowed for future training. It means that any instance is processed only once with training and at that time only the knowledge must be retrieved and summarized so that it can be used in model building process [1]

2) It is known fact that the most recent dataset is a portrayal of the present environment. So the knowledge must be grouped based on its relevance to the present environment, and be dynamically brought up to date as latest data arrives.

3) The learner should have a system to resign itself when former and recently learned knowledge dispute with each other. Moreover, there should be a system for keeping track of both the incoming data and the learner's performance on recent and existing data for the intention of *complexity reduction*, *problematizing*, and *fading*.

4) The learner must have a system to neglect or omit information that is no longer relevant, but selectively with the added capability to look back such information if the drift or change chase a cyclical nature.

5) Knowledge should be incrementally stored after certain chunk of time interval so that it can start working to generate the wise speculation for an unknown (or unlabeled) data instance through any instant of time in the learning routine.[29]

3.2 Existing approaches for handling concept drift

Concept drift algorithms can be categorized in different ways, like:

1) Online vs. batch approaches
2) Single classifier vs. ensemble-based approaches
3) Incremental vs. non-incremental approaches
4) Active vs. passive approaches

Various concept drift learning algorithms exploit some kind of a sliding window over the incoming data, where the group of instances that fall within the window are assumed to be stationary, and a novel classifier is produced for each such group of data. STAGGER [3] and FLORA [4] are the very first systems who used this passive batch-based *instance selection* method. Some versions of FLORA algorithms consist of an active drift detection system, using an adaptive window narrowing or widening depending on whether the drift is fast or gradual [4]. FLORA calls the classifiers as relevant, irrelevant or potentially relevant by assessing them on the most current data. Each classifier sustains a counter based on the number of accurately classified instances, and classifiers are cut off based on their relevance in the present data window. But such an approach can result in catastrophic forgetting [5]. Another bunch of active concept drift methods make use of control charts, such as CUSUM (cumulative sum). Alippi and Roveri's *just-in-time* (JIT) classifiers [6-8], and their more latest *intersection of confidence intervals* (ICI) rule [9] are cases of such methods. Information theoretic measures, such as entropy, mutual information, or Hoeffding bounds of individual features have also been used for noting drift and changing a classifier like a decision tree [10-12]. Several methods among them also consists of a FLORA-like windowing system, including Hulten et al.'s concept adapting very fast decision tree (CVFDT) [13] or Cohen et al.'s incremental online-information network (IOLIN) algorithms [14;15].

Most of the methods of drift detection are generally fairly able to gain success in detecting curt changes, but may clash with slow drift. The Early Drift Detection Method (EDDM) is particularly

designed for slow drift as an online active method[16]. It keeps an eye on the distance between the errors of a classifier and compares their mean to a threshold. EDDM not only indicates the drift, but also it has the capability to raise a warning if required.

3.3 Ensemble techniques as a solution for concept drift

Ensemble systems are also called as multiple classifier systems (MCS), have been designed and successfully carried out for learning in non-stationary environments [17]. With each and every new dataset arrived, new classifiers are added and many times older one are removed to create an ensemble model to keep track of the environment.

These methods generally go for a passive drift detection along with a fixed ensemble size (number of classifiers), where the most former member (e.g. Street's Streaming Ensemble Algorithm (SEA) [18], and Bifet's adaptive Hoeffding tree bagging [19]) or the ensemble member which gives minimum performance (e.g.Tsymbal's Dynamic Integration [20], Kolter and Maloof's, Dynamic Weighted Majority (DWM) [30]) is substituted with a recent one.

Although there are disagreements on which scheme to use for concept drift applications, for merging classifiers voting mechanism is preferred quite often.

Tsymbal merges a proximity measure with classifier performance to compute the voting weights, with classifiers whose training and test data are in the identical region of feature space being offered more weights [20]. Gao, recommends simple majority voting [33], highlighting that weights based on classifier error on drifting data is not giving enough information for later datasets. Other versions of ensemble methods consists of [34-36].

Hybrid methods that combine active detection, sliding windows and classifier ensembles have also been stated, such as random forests with entropy [37] and Bifet's novel integration of a Kalman filter merged with an adaptive sliding window algorithm, namely ADWIN [38].

Bifet et. al. has launched a software suite similar to WEKA, called Massive Online Analysis (MOA) at [39], which contains implementations of ADWIN and a various other tools for stream data mining with concept drift. Another application where active ensemble method id implemented is Hoens et. al.'s latest work that merges random subspace approach with Hellinger distance to observed drift in traits of the data [44].

Ensemble based concept drift algorithms also refer to some of the algorithms from Learn++ family such as the incremental Learn++.NSE algorithm [29;41-44], which produces a novel classifier with each batch of data that arrive. The classifiers are merged via dynamically weighted majority voting. Weights are developed from the time-adjusted error of each and every classifier in the ensemble, sigmoidally averaged over all environments. Learn++.NSE is able to keep track of several environments, such as gradual, fast, abrupt, and cyclical drift. It is able to recognize most and least relevant classifiers, give them high and low voting weights, and notice when a classifier turn relevant again, whether the environment and its settings peruse a cyclic nature. The algorithm is also able to take in class which are added or removed. Only thing it cannot do is it is not designed to handle class imbalance.

4. ISSUES OF DATA STREAM LEARNING – CONCEPT IMBALANCE

Class imbalance happens when a dataset does not contain (approximately) identical number of instances form each class, which can be drastic in many situations [45].Imbalanced data learning is a major issue and its detailed survey with assessment matrix and state-of-art techniques to

address the problem is available in [46]. A simple solution to this problem is under/over sampling of majority/minority class data. But this approach has pitfalls: under-sampling discards instances from the majority class, without considering usefulness of those instances. On the other side, oversampling generates duplicates of the minority examples, which might turn classifier to overfit the minority class examples. There are smarter mechanisms like condensed nearest neighbor (CNN) rule which discards instances from the majority class that are remote from the decision boundary [47]. In one more approach, Tomek joins that under-samples the majority class by clarifying a distance between examples from various classes [48].

An intelligent approach, is used by Chawla's SMOTE (*Synthetic Minority Oversampling TEchnique*) algorithm [49], which inhabits the minority class feature space by tactically putting artificial instances on the line segment connecting two minority examples. SMOTE has been proved to enhance the classification accuracy on the minority class over other standard mechanisms. Also, SMOTEBoost [50] merges SMOTE and AdaBoost.M2 to further improve *F*-measure and recall. Bagging ensemble variation (BEV) was launched [51], which trains classifiers by applying bagging method on entire minority class data and subsets are taken for training from majority class data. Learning from unbalanced data is an issue which was addressed in an algorithm called Learn++.UD [64] but it is not having capability to learn new concept classes. Learn++.UDNC is joint solution to the problem of unbalanced data as well as learning from new concept class. In other words it is capable to incrementally learn latest concept classes from unbalanced datasets. Alternatively, Learn++.UDNC collectively implements new qualities of several algorithms within the Learn++ family, like a class specific weighting approach, normalized preliminary confidence measures and introduces new transfer function that is able to bring down the confidence bias of a sub-ensemble which is trained on a majority class[52]. DataBoost-IM algorithm discovers difficult to classify minority class examples, based on which generates new artificial data [53]. Classifier precise methods, like rebalancing planned to deal with support vector machines (SVMs), have also been launched recently [54; 55].

5. ISSUES OF DATA STREAM LEARNING –JOINT PROBLEM OF CLASS IMBALANCE AND CONCEPT DRIFT

A massive amount of applications in non-stationary environments with concept drifting data sources are affected from class imbalance (e.g. Climate monitoring, spam e mail identification, credit card fraud detection, network intrusion detection system)**.**Ensemble approaches are the one which are mostly used when handling concept drift, in addition they have been proved useful for combating class imbalance [36].

A learning algorithm for non-stationary environment from imbalanced data has been stated by Gao et al. [36; 57]. The algorithm called as *uncorrelated bagging* (UCB), is based on a bagging framework that trains classifiers on a subset of the majority class instances (which is selected by some user defined threshold parameter) and the combination of all minority class examples observed so far (present + former positive instances). There are several drawbacks of this approach due to its implicit assumption. It is not a true one pass (incremental) algorithm either. Chen and He's *Selectively Recursive Approach* (SERA) learning algorithm goes for a similarity measure to choose former minority examples that are most close to those in the most recent dataset [58]. SERA is less liable to challenges of the minority class drifting compared to UCB [60], and may be implemented in either of two methods: i) create only one classifier on each dataset; or ii) Apply biased bagging, *BBagging,* which itself by its own raises number of sampling weights of the minority data. SERA omits instances from the present training set by what it considers as not useful by using a Mahalanobis distance measure. SERA is updated to apply an ensemble technique in [59; 60]. But, this technique is also not strictly single pass, as it accesses previous data.

Therefore both approaches written above perform best when the minority data concept is stationary and/or the former (minority class) data can be stored for later use. Xioufis et. al. came up with a window-based approach that employs a *k*-NN for multi-label classification for data that includes issues like concept drift and class imbalance [61].

Ditzler and Polikar proposed two ensemble based techniques that can learn in a broad area of concept drift in nonstationary environment that addresses heavy class imbalance problem, also avoids the main limitation of previous methoss, namely, amassing minority data and employing some portions of former data [62]. The approaches known as Learn++.NIE and Learn++.CDS, are truly incremental (one pass) methods that do not need access to former data, and they do not build up minority data to balance the class balance. Smartly, Learn++.CDS applies SMOTE to rebalance the classes and its instances, whereas Learn++.NIE employs sub-ensembles with bagging, along with alternate error measures to learn from imbalanced data. Both algorithms are capable to gain recent knowledge and save old information about the environment and its settings, which is specifically useful for recurring concepts (e.g. spiral dataset). Particularly, Learn++.CDS is a union of the two algorithms, Learn++.NSE [7] and SMOTE [46], existing methods for concept drift and imbalanced data issues, respectively. The robust approach used SMOTE to rebalance the classes employing artificial (synthetic) minority class data, also used Learn++.NSE on rebalanced data.

Learn++.NIE, is more intelligent approach which employs a different technique that is mainly based on two pillars: 1) Employ bagging based sub-ensembles to cut off class imbalance (without producing synthetic data, and without collecting minority data); and 2) Apply different measures that gives stress on both class-specific performances to weigh classifiers.

Learn++.NIE algorithm performs well, if both minority and majority class definitions are changing and a strong balancing is required on both minority and majority classes to achieve better performance . It has the unique characteristic to hold performance for class specific recall [63].

6. ISSUES OF DATA STREAM LEARNING –MISSING FEATURES

The unity and completeness of data are necessary for any classification system. A trained classifier requires special training to address this challenge and cannot operate examples with lost characteristics. Missing data in real world applications is a common scenario: malfunctioning sensors, wrong pixel information, blank answers to questions asked in surveys, failed equipment, medical tests that cannot be monitored under some specific situations, etc. are all common applications in real-life that can result in lost characteristics. Values of features that are beyond certain dynamic minimum and maximum limit of the data caused by utmost noise, signal saturation, data corruption, etc. can also be considered as missing features. The simplest way to deal with such lost data is to neglect those instances that have missing attributes. When large portions of the data is facing problem of missing features or lost attributes filtering or deletion (list wise) are suboptimal (and impractical due to conditions applied on them to work well) approaches and generally known as filtering [85]. A more realistic approach is imputation where missing values are filled with a meaningful calculation [86-89] like mean or finding k-nearest neighbor value etc. to get correct estimates of missing values training data should be dense with enough number of features.

These techniques tends to produce biased results. Polynomial regression is one more method to handle missing data but regression technique is suitable to some specific kind of applications. Approaches with good performance guarantees have also been discovered. Many of these approaches are based on model based estimation, like Bayesian estimation [90-92], these method

calculate posterior and prior probabilities by uniting the missing feature space. Such methods also need enough dense data distribution and data distribution with parameters should be known in advance. Such prior knowledge in difficult to obtain in most of the cases. Alternatively Expectation Maximization (EM) [93-95; 91] algorithm is theoretically proven iterative method which is even simple to construct. EM suffers from two limitations: 1) Slow convergence if large part of data has missing values; and 2) Maximization step may be pretty hard if a closed form of the data distribution is not known, or if different examples have lost different features. So in such cases, EM does not work well in real world [2]. EM also needs information in advance about the underlying distribution that is not available usually.

If the distribution calculation is wrong then it may lead to inconsistent outputs, whereas unavailability of enough dense data may cause accuracy loss. To overcome such difficulties, rigorous approaches have been suggested such as using Gaussian mixture models [96, 97]; or Expected conditional Maximization [86]. There are several other approaches such as neural network based methods [98, 99], neuro –fuzzy algorithms [100]. Algorithms based on the common fuzzy min-max neural network or ARTMAP and fuzzy c-means clustering [101] are some of the examples of the method. Ensemble based approaches have also been launched such as Melville et al. [102] proved that the algorithm DECORATE that produces artificial data (with lack of missing values) from already present data (with the presence of missing values) is pretty robust to lost features. Juszczak and Duin [103] suggested uniting an ensemble of one class classifiers, where each classifier trained on one feature. This method is able to handle any type of combination of missing data, with less number of classifiers. This technique can be much powerful as long as single characteristic is able to calculate the underlying decision boundaries. But that is not often reasonable. Polikar et al. proposed a new way that belongs to Learn++ family known as Learn++.MF [84] produces sufficient frequency of classifiers whereby each one is trained with randomly chosen feature subset.

An instance with lost values is classified by majority voting of those classifiers that didn't employ the lost values in the period of the training. Therefore, this method can be distinguished by other methods in fundamental aspect. Learn++.MF attempts to extract the most unfair classification information stipulated by the present (old) data, so that it can take most benefit of duplication in the feature set. In other words, Learn++.MF do not have any concern with the estimation prediction of missing data values. Therefore, Learn++.MF prevents many of the demerits of estimation and imputation based approaches. This is a novel way to accommodate crucial amount of lost data values and without slow decline in performance as the volume of data grows. A detailed analysis can be found in [84] with simulation results and conditions in which the given algorithm performs at its best.

7. CONCLUSION AND FUTURE WORK

In this paper, we arranged our survey in terms of the four main challenges of research in handling stream data. We commented on the major areas of research in mining data streams that exhibit concept class, concept drift, class imbalance, and missing features. We discussed adaptive approach to handle concept class, combine problem of class imbalance and concept drift, and various techniques to solve such problems including ensemble technique. We observed that Learn++ family of algorithm play in important role in this problem domain along with adaptive framework known as ADAIN.

In spite of the increasing number of attempts, there is still much work to be done in data streams that constitute concept drift, class imbalance and concept class. We note that "ADAIN" solution of incremental learning is quite intelligent but does not handle concept drift and imbalance. Learn++.NIE algorithm is elegant technique for addressing class imbalance in non-stationary

environment. The work in this area with handling of this issues jointly is sparse and have not been evaluated thoroughly on large and real world applications. Thus we propose future work leading towards an intelligent approach that will overcome limitations of the existing research body and also rigorously assessing these techniques on large scale, real world examples.

REFERENCES

[1] R. Polikar, L. Udpa, S. S. Udpa, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," IEEE Trans. Syst., Man Cybern. Part C: Appl. Rev., vol. 31, no. 4, pp. 497– 508, Nov. 2001.

[2] R. Elwell and R. Polikar, "Incremental Learning of Concept Drift in Nonstationary Environments," IEEE Transactions on Neural Networks, vol. 22, no. 10, pp. 1517-1531, Oct.2011

[3] J. C. Schlimmer and R. H. Granger, "Incremental learning from noisy data," Machine Learning, vol. 1, no. 3, pp. 317-354, Sept.1986.

[4] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," Machine Learning, vol. 23, no. 1, pp. 69-101, 1996.

[5] F. H. Hamker, "Life-long learning Cell Structures--continuously learning without catastrophic interference," Neural Networks, vol. 14, no. 4-5, pp. 551-573, May2001.

[6] C. Alippi and M. Roveri, "Just-in-Time Adaptive Classifiers - Part I: Detecting Nonstationary Changes," IEEE Trans. Neural Networks, vol. 19, no. 7, pp.1145-1153, 2008.

[7] C. Alippi and M. Roveri, "Just-in-Time Adaptive Classifiers - Part II: Designing the Classifier," IEEE Transactions on Neural Networks, vol. 19, no. 12,pp. 2053-2064, Dec.2008.

[8] C. Alippi, G. Boracchi, and M. Roveri, "Just in time classifiers: managing the slow drift case," International Joint Conference on Neural Networks (IJCNN 2009), pp. 114-120, Atlanta, GA, 2009.

[9] C. Alippi, G. Boracchi, and M. Roveri, "Change Detection Tests Using the ICI Rule," World Congress on Computational Intelligence (WCCI 2010) - International Joint Conference on Neural Networks (IJCNN 2010), pp. 1190-1196, Barcelona, Spain, 2010.

[10] P. Vorburger and A. Bernstein, "Entropy-based Concept Shift Detection," International Conference on Data Mining (ICDM '06.), pp. 1113-1118, 2006.

[11] S. Hoeglinger and R. Pears, "Use of Hoeffding trees in concept based data stream mining," International Conference on Information and Automation for Sustainability (CIAFS 2007), pp. 57-62, 2007.

[12] C. J. Tsai, C. I. Lee, and W. P. Yang, "Mining decision rules on data streams in the presence of concept drifts," Expert Systems with Applications, vol. 36, no. 2, Part 1, pp. 1164-1178, Mar.2009

[13] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," Conf. on Knowledge Discovery in Data, pp. 97-106, San Francisco, 2001

[14] L. Cohen, G. Avrahami, M. Last, and A. Kandel, "Info-fuzzy algorithms for mining dynamic data streams," Applied Soft Computing, vol. 8, no. 4, pp. 1283-1294, Sept.2008

[15] L. Cohen, G. Avrahami-Bakish, M. Last, A. Kandel, and O. Kipersztok, "Real-time data mining of non-stationary data streams from sensor networks," Information Fusion, vol. 9, no. 3, pp. 344-353, 2008

[16] M. Baena-Garcia, J. del Campo-Avila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Bueno-Morales, "Early drift detection method," ECML PKDD Workshop on Knowledge Discovery from Data Streams, pp. 77-86, 2006.

[17] L. I. Kuncheva, "Classifier ensembles for detecting concept change in streaming data: overview and perspectives," European Conference on Artificial Intelligence (ECAI), pp. 5-10, Patras, Greece, 2008.

[18] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," Int'l Conf. on Knowledge Discovery & Data Mining, pp. 377-382, 2001.

[19] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavalda, "New Ensemble Methods For Evolving Data Streams," 15th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 09), pp. 139-148, 2009

[20] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen, "Dynamic integration of classifiers for handling concept drift," Information Fusion, vol. 9, no. 1, pp. 56-68, Jan.2008.

[21] Tsymbal, A.: The problem of concept drift: definitions and related work. Tech. Rep. TCD-CS-2004-15, Departament of Computer Science, Trinity College (2004).
URL https://www.cs.tcd.ie/publications/techreports/reports

[22] Widmer, G., Kubat, M.: Effective learning in dynamic environments by explicit context tracking. In: ECML, pp. 227–243. Springer (1993)

[23] G. Widmer and M. Kubat, “Learning in the presence of concept drift and hidden contexts,” Mach. Learn., vol. 23, no. 1, pp. 69–101, 1996
[24] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Dataset Shift in Machine Learning. Cambridge, MA: MIT Press, 2009
[25] L. L. Minku, A. P. White, and Y. Xin, “The impact of diversity on online ensemble learning in the presence of concept drift,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 5, pp. 730–742, May 2010.
[26] L. I. Kuncheva, “Classifier ensembles for detecting concept change in streaming data: Overview and perspectives,” in Proc. Eur. Conf. Artif. Intell., 2008, pp. 5–10.
[27] L. I. Kuncheva, “Classifier ensembles for changing environments,” in Multiple Classifier Systems, vol. 3077. New York: Springer-Verlag, 2004, pp. 1–15
[28] M. Muhlbaier, A. Topalis, and R. Polikar, “Learn++.NC: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes,” IEEE Trans. Neural Netw., vol. 20, no. 1, pp. 152–168, Jan. 2009.
[29] R. Elwell and R. Polikar, "Incremental Learning of Concept Drift in Nonstationary Environments," IEEE Transactions on Neural Networks, vol. 22, no. 10, pp. 1517-1531, Oct.2011.
[30] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: an ensemble method for drifting concepts," Journal of Machine Learning Research, vol. 8, pp. 2755-2790, 2007
[31] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen, "Dynamic integration of classifiers for handling concept drift," Information Fusion, vol. 9, no. 1, pp. 56-68, Jan.2008.
[32] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: an ensemble method for drifting concepts," Journal of Machine Learning Research, vol. 8, pp. 2755-2790, 2007.
[33] J. Gao, W. Fan, and J. Han, "On appropriate assumptions to mine data streams: analysis and practice," International Conference on Data Mining (ICDM 2007), pp. 143-152, 2007.
[34] K. Nishida and K. Yamauchi, "Adaptive Classifiers-Ensemble System for Tracking Concept Drift," International Conf.on Machine Learning and Cybernetics, eds. K. Yamauchi, Ed., vol. 6, pp. 3607-3612, 2007.
[35] H. He and S. Chen, "IMORL: Incremental Multiple-Object Recognition and Localization," IEEE Transactions on Neural Networks, vol. 19, no. 10, pp. 1727-1738, 2008.
[36] J. Gao, B. Ding, F. Wei, H. Jiawei, and P. S. Yu, "Classifying data streams with skewed class distributions and concept drifts," IEEE Internet Computing, vol. 12, no. 6, pp. 37-49, 2008.
[37] H. Abdulsalam, D. Skillicorn, and P. Martin, "Classification Using Streaming Random Forests," IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 1, pp. 22-36, 2011.
[38] A. Bifet, E. Frank, G. Holmes, and B. Pfahringer, "Accurate ensembles for data streams: Combining restricted Hoeffding trees using stacking.," 2nd Asian Conference on Machine Learning in Journal of Machine Learning Research, vol. 13, Tokyo, 2010.
[39] A. Bifet, MOA: Massive Online Analysis, Available at: http://moa.cs.waikato.ac.nz/ . Last accessed:7/6/2011.
[40] T. R. Hoens, N. V. Chawla, and R. Polikar, "Heuristic Updatable Weighted Random Subspaces for Non-stationary Environments," Data Mining (ICDM), 2011 IEEE 11th International Conference on, pp. 241-250, 2011.
[41] M. Muhlbaier and R. Polikar, "An Ensemble Approach for Incremental Learning in Nonstationary Environments," Multiple Classifier Systems, pp. 490- 500, 2007.
[42] M. Karnick, M. Ahiskali, M. Muhlbaier, and R. Polikar, "Learning concept drift in nonstationary environments using an ensemble of classifiers based approach," Int'l Joint Conf.on Neural Netw., pp. 3455-3462, 2008.
[43] R. Elwell, R. Polikar, "Incremental Learning in Nonstationary Environments with Controlled Forgetting," International Joint Conference on Neural Net-works (IJCNN 2009), pp. 771-778, 2009
[44] R. Elwell and R. Polikar, "Incremental learning of variable rate concept drift," International Workshop on Multiple Classifier Systems (MCS 2009) in Lecture Notes in Computer Science, vol. 5519, pp. 142-151, Reykjavik, Iceland, 2009.
[45] M. Kubat, R. Holte, and S. Matwin, "Machine Learning for the Detection of Oil Spills in Satellite Radar Images ," Machine Learning, vol. 30, pp. 195- 215, 1998.
[46] H. He and E. A. Garcia, "Learning from Imbalanced Data," IEEE Tran. on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, Sept.2009.
[47] P. E. Hart, "The condensed nearest neighbor rule," IEEE Transactions on Information Theory, vol. 14, no. 3, pp. 515-516, 1968.
[48] I. Tomek, "Two Modifications of CNN," IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 6, no. 11, pp. 769-772, 1976.

[49] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Oversampling Technique," Journal of Artificial Intelligence, vol. 16, pp. 321-357, 2002
[50] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, " SMOTEBoost: Improving Prediction of the Minority Class in Boosting," 7th European Conference on Principles and Practice of Knowledge Discoveryin Databases (PKDD), pp. 107-119, Dubrovnik, Croatia, 2003.
[51] C. Li, "Classifying imbalanced data using a bagging ensemble variation (BEV)," ACM Southeast Regional Conf., pp. 203-208, Winston-Salem, NC, 2007
[52] G. Ditzler, M. Muhlbaier, and R. Polikar, "Incremental Learning of New Classes in Unbalanced Datasets: Learn++.UDNC," Int. Workshop on Multiple Classifier Systems (MCS 2010) in Lecture Notes in Computer Science, vol. 5997, pp. 33-42, 2010.
[53] H. Guo and H. Viktor, "Learning form Imbalanced Data sets with Boosting and Data Generation: The DataBoost-IM Approach," ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 30-39, 2004.
[54] T. Yuchun, Z. Yan-Qing, N. V. Chawla, and S. Krasser, "SVMs Modeling for Highly Imbalanced Classification," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 39, no. 1, pp. 281-288, Feb.2009.
[55] R. Batuwita and V. Palade, "FSVM-CIL: Fuzzy Support Vector Machines for Class Imbalance Learning," Fuzzy Systems, IEEE Transactions on, vol. 18, no. 3, pp. 558-571, June2010
[56] Gao, J., Fan, W., Han, J., Yu, P.: A general framework for mining concept-drifting data streams with skewed distributions. In: SDM, pp. 3–14. Citeseer (2007)
[57] J. Gao, W. Fan, J. Han, and P. S. Yu, "A general framework for mining concept-drifting data streams with skewed distributions," SIAM International Conference on Data Mining, vol. 7, 2007.
[58] S. Chen and H. He, "SERA: Selectively recursive approach towards nonstationary imbalanced stream data mining," International Joint Conference on Neural Networks (IJCNN 2009), pp. 522-529, Atlanta, GA, 2009.
[59] S. Chen, H. He, L. Kang, and S. Desai, "MuSeRA: Multiple Selectively Recursive Approach towards imbalanced stream data mining," World Congress on Computer Intelligence (WCCI 2010) - International Joint Conference on Neural Networks (IJCNN 2010), pp. 1-8, Barcelona, Spain, 2010.
[60] S. Chen and H. He, "Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach," Evolving Systems,vol. 2, no. 1, pp. 35-50, 2011.
[61] E. S. Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas, "Dealing with concept drift and class imbalance in multi-label stream classification," International Joint Conference on Artificial Intelligence (IJCAI 2011), pp. 1583-1588, 2011.
[62] G. Ditzler, R. Polikar, and N. Chawla, "An Incremental Learning Algorithm for Non-stationary Environments and Class Imbalance," 20th InternationalConference on Pattern Recognition (ICPR 2010), pp. 2997-3000, Istanbul, Turkey, 2010
[63] Ditzler, Gregory, and Robi Polikar. "Incremental learning of concept drift from streaming imbalanced data." Knowledge and Data Engineering, IEEE Transactions on 25.10 (2013): 2283-2301.
[64] Muhlbaier, M., Topalis, A., Polikar, R.: Incremental learning from unbalanced data. In: Proc. of Int. Joint Conference on Neural Networks (IJCNN 2004), Budapest, Hungary, July 2004, pp. 1057–1062 (2004)
[65] Ditzler, Gregory, Gail Rosen, and Robi Polikar. "Incremental learning of new classes from unbalanced data." Neural Networks (IJCNN), the 2013 International Joint Conference on. IEEE, 2013.
[66] Y. Freund and R. Schapire, “A decision theoretic generalization of on-line learning and an application to boosting,” Comput. Syst. Sci.,vol. 57, no. 1, pp. 119–139, 1997
[67] C. Bishop, Neural Networks for Pattern Recognition. Oxford. Oxford, U.K.: Oxford Univ., 1995
[68] Razavi-Far, Roozbeh, Piero Baraldi, and Enrico Zio. "Dynamic weighting ensembles for incremental learning and diagnosing new concept class faults in nuclear power systems." Nuclear Science, IEEE Transactions on 59.5 (2012): 2520-2530.
[69] He, Haibo, et al. "Incremental learning from stream data." Neural Networks, IEEE Transactions on 22.12 (2011): 1901-1914.
[70] J. Fu, H. He, and X. Zhou, “Adaptive learning and control for MIMO system based on adaptive dynamic programming,” IEEE Trans. Neural Netw., vol. 22, no. 7, pp. 1133–1148, Jul. 2011.
[71] Ade, Ms RR, et al. "METHODS FOR INCREMENTAL LEARNING: A SURVEY."International Journal of Data Mining & Knowledge Management Process 3.4 (2013).
[72] H. He, Self-Adaptive Systems for Machine Intelligence. New York: Wiley, 2011

[73] G. G. Yen and P. Meesad, "An effective neuro-fuzzy paradigm for machinery condition health monitoring," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., vol. 31, no. 4, pp. 523–536, Aug. 2001

[74] M. Pardowitz, S. Knoop, R. Dillmann, and R. D. Zollner, "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., vol. 37, no. 2, pp. 322–332, Apr. 2007

[75] C. Ji and S. Ma, "Performance and efficiency: Recent advances in supervised learning," Proc. IEEE, vol. 87, pp. 1519–1535, Sept. 1999

[76] M.D. Muhlbaier, R. Polikar, An ensemble approach for incremental learning in nonstationary environments, in: 7th International Workshop on Multiple Classifier Systems, Prague, Lecture Notes in Computer Science, vol. 4472, 2007,pp. 490–500

[77] Hoens, T. Ryan, Robi Polikar, and Nitesh V. Chawla. "Learning from streaming data with concept drift and imbalance: an overview." Progress in Artificial Intelligence 1.1 (2012): 89-101.

[78] Polikar, Robi. "Ensemble based systems in decision making." Circuits and Systems Magazine, IEEE 6.3 (2006): 21-45.

[79] D. H.Wolpert, "Stacked generalization," Neural Netw., vol. 5, no. 2, pp. 241–259, 1992

[80] A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," Neural Comput., vol. 3, pp. 79–87, 1991

[81] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," Neural Comput., vol. 6, no. 2, pp. 181–214, 1994

[82] J. Grim, J. Kittler, P. Pudil, and P. Somol, "Information analysis of multiple classifier fusion," 2nd Int. Workshop on Multiple Classifier Systems, in Lecture Notes in Computer Science, J. Kittler and F. Roli, Eds., vol. 2096, pp. 168–177, 2001

[83] L.I. Kuncheva, Combining Pattern Classifiers, Methods and Algorithms. New York, NY: Wiley Interscience, 2005

[84] Polikar, Robi, et al. "Learn++. MF: A random subspace approach for the missing feature problem." Pattern Recognition 43.11 (2010): 3817-3832.

[85] Pallavi Kulkarni and Roshani Ade. Article: Prediction of Student's Performance based on Incremental Learning. International Journal of Computer Applications 99(14):10-16, August 2014

[86] Roshani Ade, P. R. Deshmukh, An incremental ensemble of classifiers as a technique for prediction of student's career choice, IEEE International conference on network and soft computing, July 2014

[87] Roshani Ade, P. R. Deshmukh, Classification of Students using Psychometric Tests with the help of Incremental Naïve Bayes Algorithm, International Journal of Computer Application, Vol 89, No. 14, pp. 27-31March,2014