Volume 3, Issue 9, September-2016, pp. 514-520 ISSN (O): 2349-7084
International Journal of Computer Engineering In Research Trends
Available online at: www.ijcert.org

Classification of Concept Drifting Data Streams Using Adaptive Novel-Class Detection

Ms. Aparna Yeshwantrao Ladekar*, Dr. Mrs. M.Y. Joshi**

Department of Computer Science and Engineering, MGM's College of Engineering, SRTMUN University, Nanded
*(Email: aparnaladekar@gmail.com)
** (Email: manisha.y.joshi@gmail.com)

Abstract— In data stream classification there are many problems observed by the data mining community. Four major problems are addressed, such as, concept-drift, infinite length, feature-evolution and concept-evolution. Concept-drift occurs when underlying concept changes which is common in data streams. Practically it is not possible to store and use all data for training purpose whenever required due to infinite length of data streams. Feature evolution frequently occurs in many text streams. In text streams new features like words or phrases may occur when stream progresses. New classes evolving in the data stream which occurs concept-evolution as a result. Most existing classification techniques of data stream consider only the first two challenges, and ignore the latter two. Classification of concept-drifting data stream using adaptive novel-class detection approach is used to solve concept-drift and concept-evolution problem where novel-class detector is maintained with classifier. Novel-class detector is more adaptive to the dynamic and evolving data streams. It enables to detect more than one novel-class simultaneously. This approach solves feature-evolution problem by using feature set homogenization technique. Experiments done on Twitter data set and got reduced ERR rate and increased detection rate as a result. This approach is very effective as compared with existing data stream classification techniques.

Index Terms— Concept-drift, concept-evolution, data streams, novel-class, outlier.

1 INTRODUCTION

In recent days, data stream classification is a research problem which has been widely studied problem. Data stream classification becomes more difficult due to its dynamic changing nature as compared to static data, so it requires efficient and effective techniques [13]. Data stream classification posses many problems which are observed by data mining community. Data streams are huge in ammount so it is not possible to store and use all data for training purpose. For this issue multi-pass learning algorithms are not applicable. Incremental learning approach is well suited for this problem. When underlying concept changes over time, concept drift occurs. Various techniques have been proposed to solve this problem [5, 6]. Classification model always updated with recent data to deal with concept drift.

New classes evolving in the data stream which occur concept-evolution as a result which is another property of data stream i.e. concept evolution. To deal with concept evolution problem, classification model should be able to detect novel-classes when they appear in the stream [1]. For example, network traffic stream for intrusion detection which considers class labels as an each type of attack. When new kind of attack occurs in the traffic stream, cocept evolution occurs. One other example is Twitter in which new matters may frequently evolves in stream of text messages. Most important property is feature evolution in which new words which represent features evolve and old features remove from consideration. Existing data stream classification techniques having some drawbacks [2, 3]. Masud et al. [3] did not solve the feature evolution problem. Existing data stream classification techniques falsely detected novel-class instances which are actually belong to an existing class for some data sets. Therefore, false detection rate becomes high. If there are more novel-classes than one novel-class occured in chunk then all classes are considered as single novel-class. Classification of concept-drifting data stream using adaptive novel-class detection approach uses improved technique for outlier detection and novel-class detection for reducing false alarm rate and increasing detection rate. This approach distinguishes among more than one novel-class.

First, Classification of concept-drifting data stream using adaptive novel-class detection approach allows a slack space beyond each hypersphere and derives a flexible decision boundary for outlier detection process. The slack space is controlled by using a threshold value, and it is adapted according to situation continuously for reducing the false detection rate and missed novel-class instances. Second, this

approach applies discrete gini-coefficient approach for detecting novel-class instances. This approach able to differenciate between different reasons for the appearance of the outliers i.e. noise, concept-drift, or concept-evolution. This approach derives a threshold value analytically for the gini-coefficient that identifies the case where a novel-class presents in the stream. Then use a graph-based approach for detecting the presence of multiple novel-classes. Finally, this approach uses effective feature selection technique to solve the feature evolution problem. Experiments done on Twitter data set and got reduced ERR rate and increased detection rate as a result. This approach is very effective as compared with existing data stream classification techniques.

2 RELATED WORK

A data stream is a sequence of instances which can be reading only one time using less computing and storage capabilities. Examples of data streams are computer-network traffic, ATM transactions, phone conversations, web searches, and sensor data. Data stream mining is included in data mining, knowledge discovery and machine learning. Data stream mining extracted knowledge from continuous data records i.e. rapid data [7]. The main consideration of data stream processing is to train instances and inspect them one time only which contained in streams having high speed and then discarded them to make space for particular instances. The algorithm used to process the stream which has no control over the sequence of the instances and update model incrementally with each example. Another property needs that the model is open at any time to apply on training instances [4].

The problems occured in data stream classification are solved by different researchers in different ways. Most of the existing data stream classification techniques handle concept drift and infinite length problem [6]. Those techniques follow some method of incremental technique. There are two variations of this technique: Single model approach and hybrid-batch incremental approach [11]. In single model approach, model is updated with new data which is dynamically maintained by this approach. For example, incrementally modify with new data in a decision tree. Second is hybrid batch incremental approach which uses batch learning technique to build a model. Old model is replaced by new model when required. To update a model, the hybrid approaches needs much simpler operations.

Another part of data-stream classification technique is cluster based approach which addresses the problem of concept evolution with concept drift and infinite length and detects novel-classes in data streams [10]. It defines hyper-sphere for all clusters and updated with stream progresses continuously. When cluster found out of this hyper-sphere and that cluster have some density, declared a novel-class. This approach considers only one normal class and other classes consider as a one novel-class. That's why it is not useful in multiclass data stream classification. Feature selection for data streams having dynamic feature space, this type of data stream classification techniques solves the feature evolution problem with the concept drift and infinite length problems. It includes of feature ranking incrementally method in which whenever new document arrives, first it is checked for new word in the document, if found it is added to library and update frequency statistics accordingly. New list of words i.e. ranking are calculated based on these frequency statistics and select top N words to update classifier. [2] FAE method also apply incremental feature selection technique. Classify instances using votes among different models in ensemble. This method has better performance than above approaches. But this approach uses Lossy-L conversion and doesn't detect novel-class.

Ensemble of models is used to classify unlabeled data by Multiclass Classifier and Novel-class Detector technique. During training phase decision boundaries are build. If any instance of test found outside the boundary considered as outliers. If enough outliers are found and it satisfies cohesion between outliers and separate from existing data then considered as novel-class instances. This approach does not solve feature evolution problem [3]. When more novel-classes found at a time, approach cannot differenciate between them [12]. In existing system, act miner is used which uses an ensemble classification technique for data problem and solving the other three problem which reduces the cost. Act miner is extended version of mine class. Act miner addresses four major problem concept evolution, concept drift [14], limited labeled data instances, and novel-class detection. In this method, dynamic feature selection problem and multi class classification in data stream classification based on clustering methods for collecting potential novel instances so memory is required to store. Another disadvantage is that using clustering method first find centroid which also incremental so time overhead occurs. And also not possible classify streamed data continuously. Because of continuous flow of streamed data and classification become continuous task. Streaming data mining is the most recent challenge in data mining. Characteristics of data streams is a large amount of data arrived at rate i.e. rapid and needs efficient processing method.

Existing data stream classification techniques have two main drawbacks [2, 5]. First, false detection rate (i.e. existing classes detected as a novel) is high for some data. Second, they are unable to distinguish among more than one novel-class. For reducing false detection rate and increasing true detection rate, a classification of concept drifting data streams using adaptive novel-class detection approach uses improved technique for novel-class detection and outlier detection. This approach detects multiple novel-classes simultaneously and reduces drawbacks of existing techniques.

3 FEATURE SPACE CONVERSION

Data Stream is a set of instances so it is extremely large in size. Data stream having incoming data continuously, it do not have any fixed feature space. It will have different feature spaces for different models. As incoming data is huge in ammount, new features may appear and new type of a class normally holds new set of features. Different chunks select

different set of features. A classification of concept drifting data streams using adaptive novel-class detection approach uses effective feature selection technique to solve the feature-evolution problem. The feature space of the classification models may be different from the feature space of test instance. So it needs homogeneous feature space for the test instance and model when need to classify an instance. There are three conversion methods [1]:

1) Lossy fixed conversion (or Lossy-F conversion)
2) Lossy local conversion (or Lossy-L conversion)
3) Lossless homogenizing conversion (or Lossless)

3.1 Lossy Fixed Conversion (Lossy-F Conversion)

In this conversion, the feature set is selected for the first n data chunks. That feature set is used for the whole stream which means the feature set fixed and projected all training and test instances to this fixed feature set.

3.2 Lossy Local Conversion (Lossy-L Conversion)

In this conversion, training chunk and the model built from the chunk having own feature set selected using a feature extraction and selection technique. A test instance is classified using a model in which the instance is projected to the feature set of the model.

Both Lossy conversions lose some important features due to the conversion, hence the name Lossy [1].

3.3 Lossless Homogenizing Conversion (Lossless)

To ignore the loss of information, Masud used the Lossless conversion in [2], in which each classification model having set of features selected by it i.e. own set. A test instance is classified using a model in which the model and the test instance preserve their feature dimentions and take union of feature dimensions. This conversion is called as "lossless homogenizing" because both the test instance and model saves their features and feature space i.e. converted is homogeneous for the model and the test instance. Useful features are not lost by using this conversion.

A Classification of concept-drifting data stream using adaptive novel-class detection approach chooses the Lossless conversion between these three conversions because useful features are not lost. Lossless conversion is more efficient than the Lossy conversions [1].

4 CLASSIFICATION OF CONCEPT-DRIFTING DATA STREAMS USING ADAPTIVE NOVEL-CLASS DETECTION

In this section the overall process of classification of concept-drifting data streams using adaptive novel-class detection processs is discussed. The data stream is broken down into chunks of same size i.e. equal. The latest data chunk consists of data instances which need to classify by using the ensemble. The data instances in the latest chunk got labeled by experts and now that chunk is useful for training purpose. A classification and novel-class detection method contains main steps which are as follows:

1. Adaptive threshold method for Outlier detection: utilizes decision boundary of models
2. Gini Coefficient approach for Novel-class detection
3. Multiple novel-class detection at a time.

Outlier detection process checks each instance for assuring whether it is an outlier. If we confirmed about an instance is not an outlier, then it is considered as an existing class instance by the classification model. Otherwise, we confirmed about an instance is an outlier and stored in a buffer for temporary.

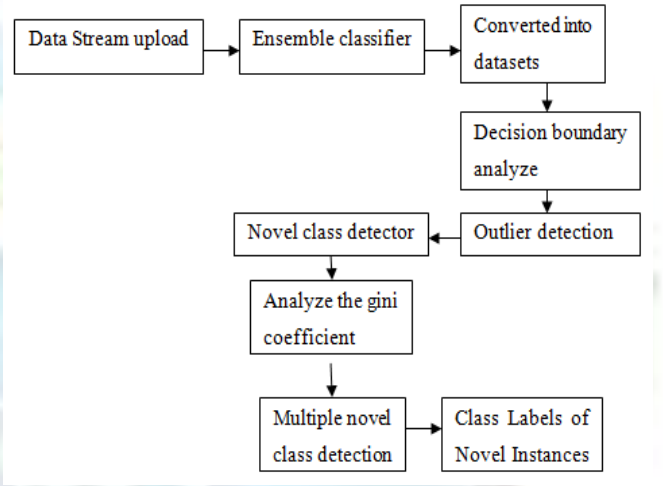


Fig. 4.1 System Architecture of novel-class detection

The novel-class detection module is called when the buffer contains sufficient number of instances. In this module, the instances of the novel-class are confirmed when a novel-class is found. If the instances in the buffer are not belonging to a novel class then considered as an existing class and use model to classify. The decision boundary decided in training is used in outlier detection module to decide about an instance is outlier or not. The novel-class detection process calculated the cohesion between the outliers in the buffer and outliers are separated from the existing classes to confirm about a novel-class.

4.1 Training Phase

The training data in which some labels are already known is used for training purpose and that trained data is used for training a k-NN based classifier [1]. In this method, the semi supervised k-means clustering is used to make k clusters and save the summary of clusters. The classification model made by storing summaries which contains the radius, centroid and frequency of the instances belongs to each class. The radius of the pseudopoint is the distance between the farthest instance in the cluster and centroid. The existing models presented in the ensemble, one model is replaced when a new model is trained. Using the latest training data model is built and selects the replacement model which is selected with the worst prediction error. At the given time by doing this we sure about exactly L models in the ensemble. Therefore requires a constant memory for storing the ensemble which automatically solves the infinite length problem. By keeping the model up-to-date with the recent data, concept-drift problem got solved.

4.2 Overall Novel-class Detection Process

Algorithm for novel-class detection shows the Classification of concept-drifting data stream using adaptive novel-class detection approach. Passing the M i.e. ensemble and the Buf i.e. buffer which consisting the outlier instances as input to the algorithm. This approach uses K-means to create K_0 clusters from the instances contained in buffer (line 2), where K_0 is similar to K i.e. the number of cluster points per chunk (line 1). Increase the speed of calculating the q-NSC value by using clustering. Now, requires constant time to calculate the q-NSC value of the outlier instance. The q-NSC value of an outlier instance h is the approximate average of the $q-NSC$ value of each instance in h [12]. Then, the $q-NSC(x)$ value [3] of outlier instance x is given by using following formule:

$$q-NSC(x)=\frac{\overline{D}_{c_{min},q(x)}-\overline{D}_{c_{out},q(x)}}{\max(\overline{D}_{c_{min},q(x)},\overline{D}_{c_{out},q(x)})} \qquad (4.1)$$

Here, $\overline{D}_{c_{min},q(x)}$ is the distance of an outlier x from closest existing class neighborhood of x i.e. minimum among all (i.e. $c_{min,q(x)}$) and $\overline{D}_{c_{out},q(x)}$ is the mean distance of an outlier x to its q-nearest outlier neighbors (i.e.$c_{out,q(x)}$).

Algorithm for novel-class detection:
Detect-Novel-Class (M, Buf)
Input: *M: Current ensemble of best L models*
Buf: Buffer temporarily holding F-outlier instances
Output: *The novel-class instances identified, if found*

1: $K_0 \longleftarrow (K^*|Buf|/S)$ //S = chunk size K = clusters per Chunk
2: $H \longleftarrow$ K-means (Buf,K_0) //create K_0 O-pseudopoints
3: for each model $M_i \in M$ do
4: tp $\longleftarrow 0$
5: for each cluster $h \in H$ do
6: h.sc $\longleftarrow$ q-NSC (h) // (equation 3.1)
7: if (h.sc > 0) then
8: tp+= h.size //total instances in the cluster
9: for each instance $x \in$ h.cluster do
x.sc $\longleftarrow$ max (x.sc,h.sc)
10: end if
11: end for
12: if (tp > q) then vote++
13: end for
14: if (vote == L then //found novel-class, identify novel Instances
15: $X_{nov} \longleftarrow$ all instance x with x.sc > 0
16: for all $x \in X_{nov}$ do
17: x.ns $\longleftarrow$ Nscore(x) //equation 3.3
18: if x.ns > $Gini_{th}$ then N_list $\longleftarrow$ N_list $\cup$ x
19: end for
20: Detect-Multinovel (N_list) //algorithm for multiple novel-class Detection (from 3.7)
21: end if

The expression $q-NSC(x)$ is a measure of cohesion and separation, and value of $q-NSC(x)$ is between -1 and +1. When the value of $q-NSC(x)$ is positive that means x is closer to the outlier instances (more cohesion) and far away from existing class instances (more separation). When the value of $q-NSC(x)$ is negative that means x is closer to the existing class instances and not an outlier. If there are q' (> q) outliers having positive $q-NSC(x)$, new class is tagged.

Now, take instances having positive $q-NSC(x)$ and calculate the $Nscore(x)$ value for them (line 17). The novel-class instance is declared when $Nscore(x)$ value is greater than threshold value ($Gini_{th}$) and save novel class in the novel class instances list (N_list) (line 18). Now, this approach computes whether multiple novel-classes appear in the stream (line 20). If the new data (i.e., S number of instances) appears in the stream, consider that the instances labels in the recent chunk will be available.

Next, briefly describe main steps in classification of concept-drifting data stream using adaptive novel-class detection approach.

1) Adaptive threshold method for Outlier detection

Outlier detection is an important part of the novel-class detection technique. A test instance is considered an F-outlier when it is outside the decision boundary of models which is decided in training phase. Take the union of feature spaces represents clusters which is the decision boundary [13, 14]. Each pseudopoint represents a hyperspherical region in the feature space which is represented by its centroid and radius. According to [3], if test instance falls outside the radius of all the pseudopoints in the ensemble of models then it is considereed as an outlier. So, test instance will be an outlier when it is close to surface of hypersphere and outside the hypersphe. Actually, this happens frequently because of concept-drift or noise, i.e. existing class instances falls outside but nearest to the surface of the hypersphere. The false detection rate (i.e., falsely detecting novel classes which are actually existing classes) becomes high. To avoid false detection problem, Classification of concept-drifting data streams using adaptive novel-class detection approach uses an adaptive threshold approach for detecting the outliers which is adapted according to situations.

This approach uses a slack space outside each hypersphere which is controled by a threshold i.e. OUTTH. The false detection rate decrease by setting threshold value too small. For that purpose, this approach adjusts the false alarm rate by using an adaptive method. Any test instance is considered as an existing when it is present in this slack space which is falsely detected as novel-class. To detect this misclassification, this approach defines inst_weight(x) as follows:

$$inst\_weight(x)=e^{r-d} \qquad (4.2)$$

Here x is a test instance, and h is the closest pseudopoint of x

in model with radius r. Consider d is the distance between x and the centroid of h. If the $inst\_weight(x)$ greater than or equal to 1, then x is inside (or on) the hypersphere Otherwise, x is outside the hypersphere. The $inst\_weight(x)$ value yields between (0, 1) when x is falls outside the surface of hypersphere. Following algorithm shows how the threshold value is adapted according to the situation.

Algorithm for adjusting threshold:
Adjust-threshold(x, OUTTH)
Input: *x: most recent labeled instance*
OUTTH: current outlier threshold
Output: *OUTTH: new outlier threshold*
1: if (false-novel(x) && OUTTH − inst_weight(x) <ϵ) then
2: {OUTTH - = δ //increase slack space}
3: else if (false-existing(x) && inst_weight(x) - OUTTH <ϵ) then
4: {OUTTH + = δ //decrease slack space}
5: end if

2) Gini Coefficient approach for Novel-class detection

In the outlier detection method, outliers detected because of many reasons. There are different reasons behind occurences of outliers i.e. concept drift, noise, or concept evolution. Calculate the gini-coefficient value which is discrete measure of the outlier instances to differenciate the outliers that falls because of concept evolution only. Gini-coefficient identifies the place where novel class presents in the stream. Gini-coefficient also called as measure of statistical dispersion. Gini-coefficient yields a value between [0, 1]. The dispersion is high when the value of gini-coefficient is high. We have confident about concept evolution at that time when the gini-coefficient is higher than a particular threshold value.

This approach calculates $q - NSC(x)$ using (4.1) for outliers which are detected in outlier detection module. Remove instance from consideration if $q - NSC(x)$ value is negative i.e. instance is an existing. Now, compute a $Nscore(x)$ for each outlier which is measure of novelty having positive $q - NSC(x)$, called Novelty score or Nscore using following formula:

$$Nscore(x) = \frac{1-inst\_weight(x)}{1-minweight} q - NSC(x) \qquad (4.3)$$

Where, $minweight$- is the minimum inst_weight among all outliers having positive $q - NSC$ value. The first part of $Nscore(x)$ shows the distance of an outlier from its closest existing class instance (higher value indicates greater distance). The second part of $Nscore(x)$ shows the closeness of the outlier with other outliers, and far away from the existing class instance. The $Nscore(x)$ value yields between [0, 1]. If the $Nscore(x)$ value is high then the instance is a novel-class instance. By examining the distribution of $Nscore(x)$, we can decide about the novelty of the outlier instance.

Discretize the $Nscore(x)$ values into n equal intervals, and construct a cumulative distribution function (CDF) of $Nscore$. Here y_i is the value of the CDF for the ith interval. Gini-coefficient $G(s)$ for a random sample of y_i :

$$G(s) = \frac{1}{n}(n + 1 - 2\,(\frac{\sum_{i=1}^{n} (n+1-i)\, y_i}{\sum_{i=1}^{n} y_i})) \qquad (4.4)$$

If the gini-coefficient value comes under wide ranges, then (4.4) able to differenciate among different cases where instance is present. The main important case occurs for $Nscore(x)$ if the outliers contains mix of data and that happens because outliers are existing class or mostly novel-class. First case occurs when all values of $Nscore(x)$ are very low. So, $y_i = 1$ for all i. This case happens because all outliers are existing classes. Second case occurs when all values of $Nscore(x)$ are very high. So, $y_i = 0$ for all $i < n$ and $y_n = 1$. This case happens because all outliers novel classes. Third case occurs when the values of $Nscore(x)$ are evenly distributed across all the intervals. Therefore, $y_i = i/n$ for all i. If data contains mixed data, this case occurs and might be some novel-class instances. Now, we got the value of threshold for gini-coefficient to identify a novel-class by using these three cases.

3) Multiple novel-class detection process

It is possible that more novel-classes may appear at the same time in the same chunk. It is important to detect novel-class and also to detect more than one such novel-class at a time. It is common in text streams i.e. Twitter messages in which new trends emerges frequently [1]. Detecting there are more than one novel-class in chunk is a challenging problem, so requires to do it in an unsupervised fashion.

To detecting multiple novel-classes main thing behind this is to construct a graph, and determine the connected nodes in the graph. Number of connected nodes determines the number of novel-classes. In determining the multiple novel-classes, main consideration follows property which shows an instance should be closer to the instance of its own class (cohesion) and far away from the instance of other classes (separation). For example, if there are two novel-classes, then the separation between different novel-class instances should be higher than the cohesion between the same-class instances.

5 EXPERIMENTS

User gives data stream as input to the system to classify and system classify data stream given by the user and gives back the classification result to the user. Twitter data set from sentiment for academics is used for experiment. This data set contains 3256 messages (tweets) of 6 different trends (classes) and 6 attributes. Apply pre-processing on dataset to get a useful dataset. Parsing and pre-processing done on text document i.e. twitter messages. The text is divided into words i.e. tokens. The features extracted forms a feature space for a particular text.

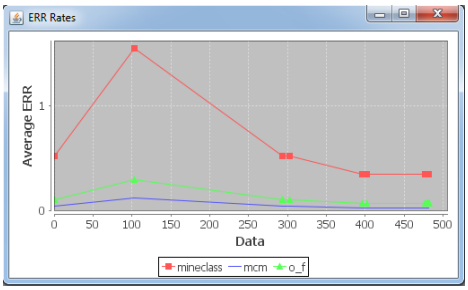


Fig. 5.1: ERR rates in Twitter

In result, MineClass is the existing approach proposed in [12], MCM is multiclass miner which is used in this work and O-F is a combination of the OLINDDA [10] with FAE [13] approach. Fig. 5.1 show the ERR rates for each approach. At X-axis show the stream, the Y-axis show the average ERR rate of each approach in Twitter data set. The ERR rate of MineClass is 17.2, MCM is 1.3 and O-F is 3.3 percent.

The ROC curve is plotted by taking false detection rate against the true detection rate. Fig. 5.2 shows the ROC curves for the Twitter data set, and the area under the curve (AUC) for MineClass is 0.88, MCM is 0.94 and O-F is 0.56. In this work, AUC for this approach is very high as compared to existing approaches which means true class detection rate is increased. So, false detection rate got reduced.

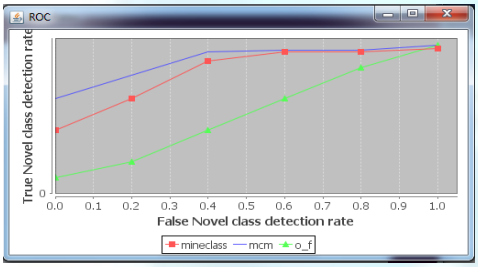
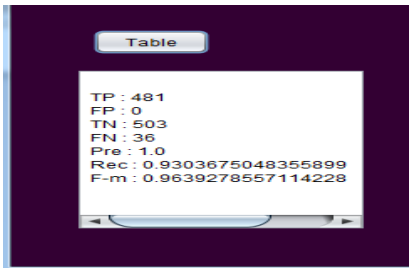


Fig. 5.2: ROC curves in Twitter

Table 5.1 shows the results of multiple novel-class detection process which shows accuracy and misclassification. The row "TP" report correctly detected number of type 1 novel-class instances. The row "FP" reports the number of incorrectly detected type 1 classes which are actually type 2 novel-class instances. The row "TN" report correctly detected number of type 2 novel-class instances and the row "FN" report the number of incorrectly detected type 2 classes which are actually type 1.

Table 5.1: Summary of Multinovel-class Detection Results



This approach also summarizes our findings by reporting the precision, recall, and F-measure which are based on the misclassification. The accuracy is simply the percentage of correctly classified instances. The precision is the number of instances correctly classified as its true class out of all the instances classified as that class. Again the recall represents the number correctly classified instances of a class out of all the instances of that class. F-measure gives a good indication of the overall performance of a classifier. This evaluation process focused on finding the optimum classification model for the dataset.

6 CONCLUSIONS

A classification of concept drifting data streams using adaptive novel-class detection technique solves concept evolution, feature evolution, infinite length and concept-drift which are not considered by most of the existing techniques. This approach allows a slack space outside decision boundary by using improved technique for outlier detection and this slack space adapted according to situations based on evolving data. By using this technique reduced false alarm rate which is high in existing technique by increasing True novel-class detection rate i.e. 0.94, which is 0.56 in existing system and reduced ERR rate i.e. 1.3, which is 17.2 in existing system. This approach predicts changes over time and maintains accuracy of prediction by detecting changes over time. This approach achieves better results compared to existing techniques.

A classification of concept drifting data streams using adaptive novel-class detection technique applied on offline dataset only. In future, it may be implemented this technique on online application. It can be applied this technique on network traffic stream for intrusion detection which considers class labels as an each type of attack. When new kind of attack occurs in the traffic stream, cocept evolution occurs. Another one future work is to identify the case where one class split into many classes then they occupy same feature space as before split is the same as the union of feature spaces covered after split.

ACKNOWLEDGMENT

I would like to acknowledge all the people who have been helped and assisted me throughout my work. First of all, I would like to thank my guide **Dr. Mrs. M.Y. Joshi** for her valuable guidance, constant encouragement and inspiring efforts for completion of this work. Without her efforts this research work would not have been completed.

I would also like to thank my family members for their sacrifices and time-to-time advice, moral support and encouragement for making this work successful. The acknowledgement would be incomplete without mentioning the blessing of the almighty, which helped me in keeping high moral during difficult period.

REFERENCES

[1] Mohammad M. Masud, Member, IEEE, Qing Chen, Member, IEEE, Latifur Khan, Senior Member, IEEE, Charu C. Aggarwal, Fellow, IEEE, Jing Gao, Member, IEEE, Jiawei Han, Fellow, IEEE, Ashok Srivastava, Senior Member, IEEE, and Nikunj C. Oza, Member, IEEE," Classification and Adaptive Novel-class Detection of Feature-Evolving Data Streams," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 7, July 2013.

[2] M.M. Masud, Q. Chen, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Classification and Novel-class Detection of Data Streams in a Dynamic Feature Space," Proc. European Conf.Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pp. 337-352, 2010.

[3] M.M. Masud, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Integrating Novel-class Detection with Classification for Concept-Drifting Data Streams," Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pp. 79-94, 2009.

[4] A. Bifet and R. Kirkby. Data stream mining – a practical approach. http://moa.cs.waikato.ac.nz/downloads/.

[5] M.M. Masud, Q. Chen, L. Khan, C. Aggarwal, J. Gao, J. Han, and B.M. Thuraisingham, "Addressing Concept-Evolution in Concept-Drifting Data Streams," Proc. IEEE Int'l Conf. Data Mining (ICDM), pp. 929-934, 2010.

[6] G. Hulten, L. Spencer, and P. Domingos, "Mining Time-Changing Data Streams," Proc. ACM SIGKDD Seventh Int'l Conf. Knowledge Discovery and Data Mining, pp. 97-106, 2001.

[7] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütz, "Introduction to Information Retrieval," e, 2008.

[8] "Stemming", http://en.wikipedia.org/wiki/Stemming.

[9] M.F.Porter, "An algorithm for suffix stripping," Computer Laboratory, Cambridge.

[10] E.J.Spinosa, A.P. de Leon F. de Carvalho, and J. Gama, "Cluster-Based Novel Concept Detection in Data Streams Applied to Intrusion Detection in Computer Networks,"Proc. ACM Symp. Applied Computing (SAC), pp. 976-980, 2008.

[11] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Dynamic Feature Space and Incremental Feature Selection for the Classification of Textual Data Streams, " Proc. IntlWorkshop Knowledge Discovery from Data Streams (ECML/PKDD), pp. 102-116, 2006.

[12] M.M. Masud, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Classification and Novel-class Detection in Concept-Drifting Data Streams under Time Constraints," IEEE Trans. Knowledge and Data Eng., vol. 23, no. 6, pp. 859-874, June 2011.

[13] B.Wenerstrom and C.Giraud-Carrier, "Temporal Data Mining in Dynamic Feature Spaces," Proc. Sixth Int'l Conf. Data Mining (ICDM), pp. 1141-1145, 2006.

[14] W. Fan, "Systematic Data Selection to Mine Concept-Drifting Data Streams," Proc. ACM SIGKDD 10th Int'l Conf. Knowledge Discovery and Data Mining, pp. 128-137, 2004.

