IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 6, June 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006106 |

Detection of Concept Drift in Telecommunication Stream Data

Anusha RV[1], Fareena S[2], Chandumani S[3], Zaiba Muskan[4], Prof Abdul Razak MS[5].

BE Student, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India[1]

BE Student, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India[1]

BE Student, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India[1]

BE Student, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India[1]

Assistant Professor, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India [5]

ABSTRACT: Predictive services are a big part of every business these days. But "concept drift," which happens when data streams change over time, makes it hard for machine learning models that are already being used. This thing can make it hard for models to make good predictions. So, most of the time, retraining the model fixes concept drift. But the research we have right now doesn't tell us how to choose which data to use to retrain the machine learning model. Here, we'll use what we've learned about concept drift in a use case in telecommunications. The proposed model is meant to learn and adapt on its own to changes in telecommunication data.

.

KEYWORDS: Predictive Services, Telecommunication, Drift Adaptation, Human Intervention.

I. INTRODUCTION

In telecommunication, a drift is a change that takes a long time to happen in an attribute, value, or operational parameter of a system or piece of equipment. The drift should have a name, like "daily frequency drift" or "output level drift." Drift is usually bad and moves in one direction, but it can also move in both directions, go in cycles, or last so long and move so slowly that it doesn't matter.

Concept drift

In predictive analytics and machine learning, the term "concept drift" is used. It means that the target variable, whose statistical properties the model is trying to predict, changes in ways the model did not expect. This is a problem because the predictions get less accurate over time.

The amount that needs to be guessed is called a "concept." It can also mean things other than the target concept that are interesting, like an input, but in the context of concept drift, it usually means the target variable.

"High variability" refers to the problem of "concept drift," which happens when data isn't always the same and the environment changes. Streaming data can be sent in a lot of different ways because the way communications work is always changing.

The first problem with finding drift is that there are many different kinds of concept drift, such as gradual, sudden, and repeating drifts. The second problem with finding concept drift is that many things can cause it.

Stream Data

Streaming data is data that comes in from different sources all the time. Stream processing techniques should be used to process these kinds of data in small steps, without having access to all of the data. It is usually used in the context of "big data," which is a lot of data coming in quickly from many different sources.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006106 |

Data Stream Mining, which is also called "stream learning," is the process of getting knowledge structures from records of fast, continuous data. Using machine learning techniques, this prediction task can be learned automatically from labelled examples.

II. LITERATURE REVIEW

Concept drift usually falls into one of the following groups [1]: There are three types of concept drift: abrupt or sudden drifts, which happen when data structures change quickly (like when a sensor fails), gradual and incremental drifts, which happen when customers' buying habits change, and seasonal and recurring drifts (like when air conditioners sell more in the summer). A more detailed taxonomy [2] takes things like how big the drift is into account. Many different ideas have been put forward about how to deal with concept drifts [3]. But most methods rely on an explicit drift detection that looks for changes in how the data is spread out and then makes the necessary changes. Two of the most common algorithms are Page-Hinkley [4] and ADWIN [5]. Page-Hinkley works because it always keeps an eye on an input variable (e.g. the input data or the prediction accuracy). A change is flagged when the variable is very different from its average in the past. ADWIN, on the other hand, looks for changes by comparing two windows. As soon as the means of the two windows are different enough, a change alert is sent out and the older window is closed.
Li et al. [6] came up with the EDTC (Ensemble Decision Trees for Concept Drifting Data Streams), which is an incremental method that sets cut-points in a growing tree using three different random feature selections. Each growing node splits features at random when an instance appears so that it doesn't make branches that aren't needed. To find drift, EDTC looks at two thresholds and the way local data is spread out [6].

Harel et al. [7] came up with a way to look at how losses are spread out in the real world. The statistics for this method come from taking more than one sample of the data. Random permutations are used in the method to find drift, and the stream is used to make more than one set of train-test data. If there wasn't any drift, when the algorithm is stable, the prediction for the ordered data shouldn't be too different from the prediction for the shuffled data. When concept drift happens, the method gets time indices and uses them to change the size of the windows, start a training phase, and give information to the ensemble learners. Instances are supposed to be independent of time, but the authors say it can use tricks to keep exchangeability [7].

Wadewale and Desai [8] divide the different types of target concept drift into sudden, gradual, repeated, blip, and noise drifts. The sudden drift means that the data changes all at once and doesn't change back. Small steps and slow drifts make changes happen slowly. "Incremental drift" is the change in data values over time, while "gradual drift" is the change in the distribution of classes.
Some writers [9] say that there are two types of drift: real concept drift and virtual drift. If you think of concept drift as a change in how the data is spread out, then the real concept drift happens when the conditional distribution changes on the output but the input distribution stays the same. In the literature, virtual drift is described in different ways, such as when the way incoming data are spread out changes [9].

People usually use a sliding window, which only keeps a small number of the most recent data instances. Classifiers for data streams can be made using traditional learning algorithms and sliding windows. When someone uses a fixed-size sliding window, they have to choose between two options. The small window can quickly pick up on sudden changes, but when things are stable, it loses accuracy. Changes in data streams that happen quickly can't be found with a large size window [11]. In the drift detectors group, the Drift Detection Method is a popular algorithm (DDM). DDM keeps track of how often it makes mistakes and learns from the online classifier. The alarm goes off if the error rate goes below the bound level. DDM only works well with sudden drift. The ensemble is one of the most common ways to learn from data streams, along with windowing and drift detectors.

Recent interesting research in the field of concept drift has been focusing on more difficult problems, such as how to accurately detect concept drift in unstructured and noisy datasets [12, 13], how to quantitatively understand concept drift in a way that can be explained [14, 15], and how to effectively respond to drift by adapting related knowledge [16, 17]. When you figure out how to solve these problems, you give prediction and decision-making the ability to deal with a world that is uncertain. Concept drift techniques in data science and AI in general, and pattern recognition and data stream mining in particular, have made traditional research on machine learning much better. In a world that is always changing, these new studies make it easier for people to use analogy and knowledge reasoning. During this change, a new topic comes up: data-driven adaptive prediction/decision systems. Concept drift is a very important and well-known problem, especially in the age of "big data." This is because big data is based on the fact that the types of data and how they are spread out are unknown.

Gaps Identified
The current system has the following problems:

- Models' ability to predict the future gets worse over time.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006106 |

- The models we have now can't keep up with the amount and speed of data being made.
- Models are made with fixed data and can't work with real-time data.
- Takes up a lot of memory to store information.
- You can only do things in batches.

Problem Statement

Static data can have gaps, outliers, or wrong information, and models built on static data lose their ability to predict over time. The data coming from the telecommunications field is streaming data, which changes all the time. This makes it hard to predict how many customers will leave. The problem is that there is a lot of data, and it's hard to keep up with it and analyse it, and there isn't a model built to handle the large amount of data and the concept drift in the data.

Proposed solution

The ADWIN algorithm and several built-in functions of scikit-multiflow will be used to handle the concept drift in the proposed solution. Standard machine learning models use batch processing and build the model by processing the data and then training it with the same static data. The model is then tested and proven to work. But in our proposed solution, we build a model and process each piece of data separately. We then send the data to our model to predict, and if the prediction is right, we train the model with the predicted data and update the model. So, every time we try to predict the results, our model will be updated. Since it only looks at one sample at a time and processes it only once, it uses a small amount of memory, works quickly, and is always ready to predict. This will eventually slow down how fast the model is falling apart. Our model's results are based on Hoeffding Tree, which gives us graphs and an idea of how accurate they are.

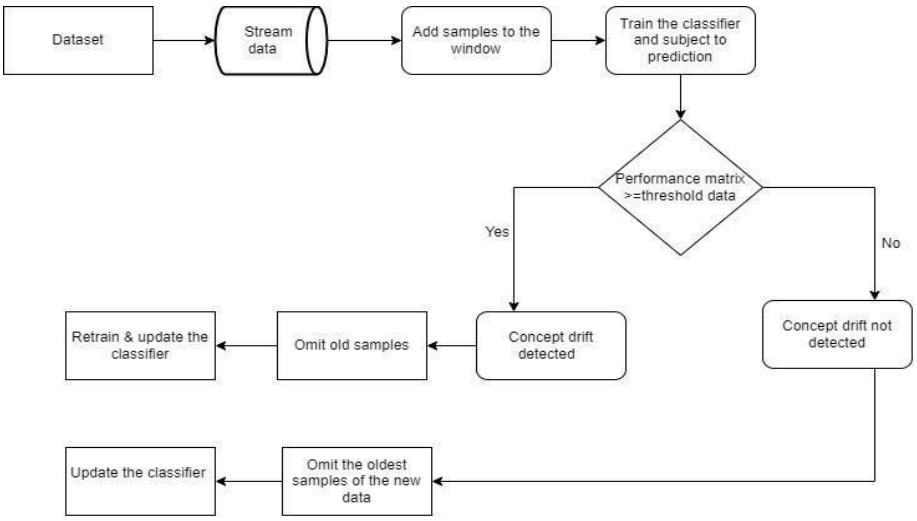
Objectives:

The main objectives are:

- To find representative instances over stream data.
- To update representative instances in case new arrival of data.
- To develop a classification model to identify drift in stream data.
- To measure the performance of the model.

III. METHODOLOGY PRESENTED

Methodology Diagram



Methodology Description

DataStream:

A data stream is a group of information that has been taken from a data provider. It has raw data that was collected from how users act. Data streams can be used to process and look at data. In our model, we've used data from telecommunications.

Sliding Window:

A sliding window moves along a data stream, so it shows the most recent examples and gets rid of the older ones when

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006106 |

concept drift is found. The sliding window is not fixed; it changes based on what is happening. When the concept drift is found, the sliding window is changed.

Classifier:
A classifier is a type of machine learning algorithm that puts a label on a set of data based on what kind of group it belongs to. It is a set of rules that tell how the data should be arranged. It automatically looks at the data, speeds up processes, and gathers useful information. Classifier automatically sorts or groups data into one or more classes.

Concept drift:
Concept drift in machine learning is when the relationships between the data that went in and the data that came out of the problem change over time. The statistical properties of the target variable, which the model is trying to predict, change in unexpected ways over time.

The data from the stream are fed into the sliding window. The samples are taken by the window and sent to the classifier. The classifier is trained and tested by giving it samples from the sliding window. A threshold value is taken into account. Based on what the classifiers say, the score of the performance matrix is checked. Based on what the classifiers say, the score of the performance matrix is checked. If the score on the performance matrix is the same as or higher than the threshold data, concept drift won't be found. If the performance matrix score is lower than the threshold data, concept drift is found. Old data samples are thrown out and new data is added to the sliding window. The classifier is then retrained and updated. Even if the drift is not found, the data samples are thrown away and the classifier is updated with new data samples.

Training Process
Random Forest Classifier: A random forest is a meta-estimator that uses averaging to improve the accuracy of predictions and stop over-fitting. It does this by fitting a number of decision tree classifiers to different sub-samples of the dataset.

Random Forest Algorithm: Random Forest is a supervised Machine Learning Algorithm that is often used for Classification and Regression problems. It makes decision trees from different samples and uses the majority vote for classification and the average for regression.
How the random forest algorithm works:
Step 1: In Random Forest, out of a set of data with k records, n random records are chosen.
Step 2: A different decision tree is made for each sample.
Step 3: The outcome of each decision tree is different.
Step 4: The final result for classification and regression is based on the majority vote or the average
.

IV. RESULTS AND DISCUSSION

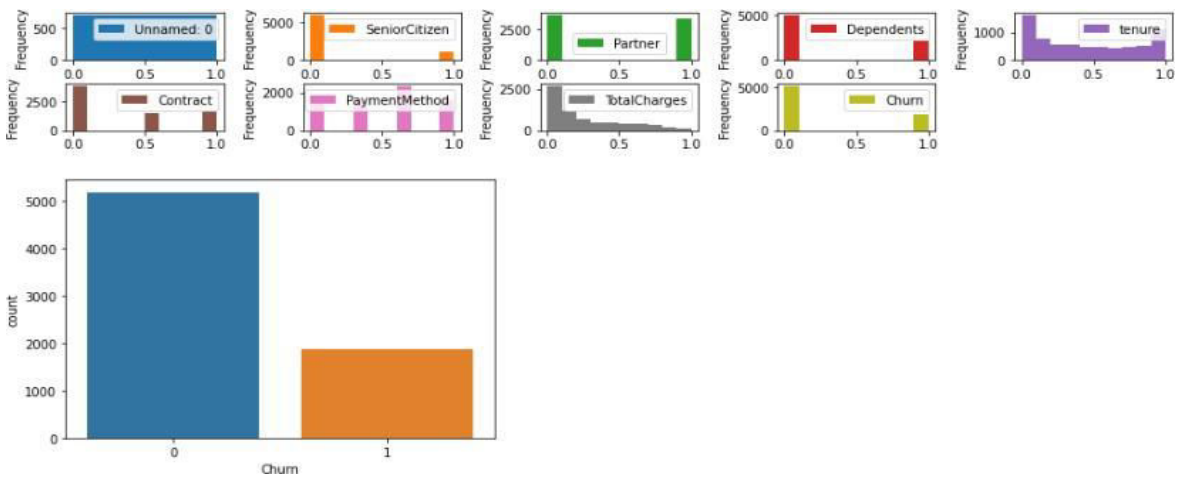


Fig: Count of data samples churn occurred or not (1 = churned, 0 = not churned)

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006106 |

	0	1
False Positive	1520	292
False Negative	292	1520
True Positives	4659	272
True Negatives	272	4659
Sensitivity	0.94	0.15
Specificity	0.15	0.94

Table: Results

Accuracy: 73.1277

FScore: 67.6069

Final accuracy: 73.1277, Elapsed time: 4.5181

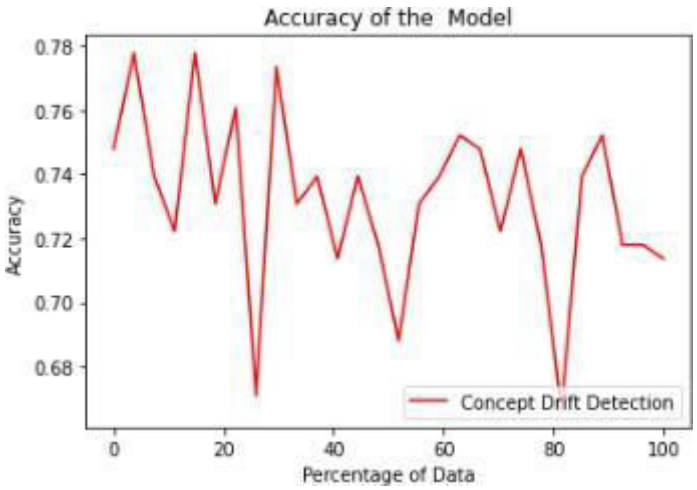


Fig: Graphical visualization of concept drift detection.

True positive: A true positive is an outcome where the model correctly predicts the positive class.

TP rate = TP / (TP+FN)

True negative: A true negative is an outcome where the model correctly predicts the negative class.
TN rate = TN / (TN + FP)

False positive: A false positive is an outcome where the model incorrectly predicts the positive class.

FP rate = FP / (TN + FP)

False negative: A false negative is an outcome where the model incorrectly predicts the negative class.

FN rate = FN / (FN + TP)

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006106 |

V. CONCLUSION

The growing use of telecommunication systems has made life easier for people, but it has also made it harder to collect and process large amounts of data from different service providers in telecommunication environments. Unlike traditional static data, telecom data is often large amounts of streaming data in environments that are not stable and change quickly. Adaptive ML methods are good solutions because they can handle IoT data streams that are always changing by adjusting to possible concept drifts.

VI. FUTURE WORK

- Getting more accurate by making changes or additions to the current model.
- Figure out why there are inconsistencies and false positives.
- Running more tests on datasets from other domains with different characteristics to prove the model works and improve its performance.

REFERENCES

1. Bifet, Albert, and Richard Gavalda. “Learning from time-changing data with adaptive Windowing.” In Proceedings of the 2007 SIAM international conference on data Mining, pp. 443-448. Society for industrial and Applied Mathematics, 2007.
2. N. L. A. Ghani, I. A. Aziz, and M. Mehat, “Concept Drift Detection on Unlabeled Data Streams: A Systematic Literature Review,” in 2020 IEEE Conference on Big Data and Analytics (ICBDA), 2020, pp. 61-65, doi: 10.1109/icbda50517.2020.9289802.
3. J. Lu, A. Liu, F. Dong, F. GU, J. Gama, and G. Zhang, “Learning under Concept Drift: A Review,” IEEE Trans. Knowl. Data Eng., vol. 31, no. 12, pp. 2346-2363, 2019, doi: 10.1109/TKDE.2018.2876857.
4. C. Manapragada, G. I. Webb, and M. Salehi, “Extremely fast decision tree,” Proc. ACM
5. SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 1953-1962, 2018, doi: 10.1145/3219819.3220005.
6. J. Montiel, J. Read, A. Bifet, and T. Abdessalem, “Scikit-multiflow: A Multi-output Streaming Framework,” J. Mach. Learn. Res., vol. 19, pp. 1-5, 2018, doi: 10.5555/3291125.3309634.
7. L. Yang and A. Shami, “A Lightweight Concept Drift Detection and Framework for IoT Data Streams,” IEEE Internet of Things Magazine, 2021, doi: 10.1109/IOTM.0001.2100012.
8. Bach, S. H. and Maloof, M. A. 2008. “Paired learners for concept drift.” Eighth IEEE International Conference on Data Mining IEEE.
9. Baena-Garcia, M. del Campo-Avila, J. Fidalgo, R. and Morales- Bueno, R. 2006, “Early drift detection method”. Fourth International Workshop on Knowledge Discovery from Data Streams 6: 77-86.
10. Bifet, A.and Gavalanda, R. 2007. “Learning from time changing data with adaptive Windowing” Proceedings of the 2007 SIAM international conference on data mining. Society for Industrial and Applied Mathematics.
11. Bifet, A. 2009. “Adaptive Learning and Mining for Data Streams and Frequent Patterns”, Doctoral Thesis.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.165

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details