UTRECHT UNIVERSITY

Department of Mathematics

Applied Data Science Master thesis

Unsupervised drift detection in accounting data

First examiner:
Sjoerd Dirksen

Candidate:
Linda Ilic

Second examiner:
Kees Oosterlee

In cooperation with:
Fré Vink (Auditdienst Rijk)

July 7, 2023

Abstract

As machine learning becomes more popular and available to all sectors ML models have to be maintained and their performance has to be monitored. Large-scale disruptive events such as the COVID-19 pandemic have a big influence on society and possibly also on the data that reflects it. As a result, the performance of an ML model might decrease substantially. This change in data is difficult to monitor in the absence of labels. As this project is in collaboration with the Auditdienst Rijk and labels are not readily available in their data environment this paper proposes the use of the STUDD method to detect drift in an unsupervised way. The hypothesis was that drift should be detected in new unseen accounting data around early 2020 when new COVID-19-related policies were implemented and affected the budget and spending patterns of the Dutch government. Here we show that the STUDD method successfully detected drift in the new unseen data early on in the pandemic year. However, this change can not be attributed to the spread of COVID-19 as policies were implemented a substantial time after the first drift was detected. This might indicate other reasons for the changes in the data such as time or some external events that occurred in the previous year and already induced drift.

Contents

1. Introduction

Nowadays, private and public sectors are incorporating Machine Learning (ML) in their processes. Often, well-trained ML models are deployed and expected to maintain stable performance in the real world. However, changes in the data might occur over time and the model has to detect and adapt to these in order to maintain its original performance. Different types of changes can occur in the data. A change in the distribution of the input data (X) is defined as *data drift*, whereas a change in the joint distribution of the input X and the target y is classified as *concept drift*. In the latter case, the relationship between the input X and target y has changed. Both types of drift might lead to deterioration of the model's performance. These changes in the data are a reflection of changes in the environment the model is deployed and can be gradual or abrupt [1]. The COVID-19 pandemic is an example of a big and abrupt event that affected all layers of public life. Thus, ML models deployed in sectors that were highly impacted by the pandemic have to account for possible changes in the data in order to maintain their performance. The Auditdienst Rijk (ADR), a branch of the Dutch Ministry of Finance, was aware of this and wanted to implement a method that can monitor these changes and asses if their model is still able to perform adequately when confronted with new unseen data. They provided a pre-trained CNN model, the data it was trained on, and new unlabelled data from 2020. The main hypothesis of this paper was that there should be drift detected that coincides with the first large-scale COVID-19 policies implemented by the Dutch government. To perform the drift detection the STUDD method developed by Cerqueira, Gomes, Bifet & Torgo (2022) was selected.

The current paper presents the methodology and results of the project and is structured as follows. First, some more in-depth information about concept and data drift is presented in this section, followed by a review of the

current state-of-the-art methods available in drift detection (section 1). The second section (2) elaborates on the data that has been included in the analysis. This is followed by an explanation of the proposed method, why it was selected and how it was implemented (section 3). Subsequently, the results (section 4) and conclusions are presented and discussed in light of possible limitations (section 5).

1.1 Concept vs Data drift

To understand the concept of *data drift* and *concept drift* more clearly it is useful to define these two terms formally as they have been used inconsistently in the past. There has been significant confusion within the scientific community over the definition of concept and data drift and there have been several attempts to reach a universal definition [1]. When referring to concept drift in this paper I am defining it as a change in the joint probability distribution of X and y, formally defined as $P(y|X)$. In a machine learning context, X would be a (set of) feature(s) that might be used as input to an ML model to predict the target y. During training, the classifier "learns" to capture the relationship between X and y i.e. it approximates the true function which maps the input X to the output y [2]. But when there is a change in the joint probability distribution of X and y, the relationship between X and y might change and the ML model that is unaware of these changes might not be able to predict y as well as before. A good example of concept drift is the changes in spam emails. A model might initially learn what features or content within an email are indicative of a spam email. When the way spam emails are generated and their content changes the spam filter cannot recognise these new spam emails as spam because it did not learn what features and content the new spam emails are characterised by. In other words, new features are introduced or changed and the definition of a spam email has to be adapted. Thus concept drift is likely to impact the performance of a model in a negative way.

Data drift, on the other hand, might or might not impact the performance of an ML model negatively. This type of drift is characterised by a change in

the probability distribution of the input data X. For example, an ML model is supposed to predict the probability of a visitor clicking on certain content on a website based on their age and gender. The model might not be able to predict the probability accurately if the distribution of these two features changes significantly. However, there are also instances where changes in the distribution of the input might not impact the performance of a model. The drift detection method used in this project monitors data drift as labels are not readily available. The underlying assumption proposed by the researcher who developed the method is that data drift that is accompanied by a drip in performance might be an indication of concept drift [2].

1.2 State of the art

Many drift detection methods have been developed but there is no one-fit-all method that can be generically used to detect drift. The selection of a method is based on the characteristics of the data and the trained ML model. One of the biggest limitations is label availability. In an ideal scenario, the real-life behaviour or ground truth would be available after the ML model has generated the predicted labels. In practice, this means that if an ML model is trained to predict or detect an outcome (or behaviour) and it does so in a specific instance then in a supervised setting there would be (immediate) feedback on whether or not the prediction was accurate. In this case concept drift can be detected through monitoring of a chosen performance measure over time. Various performance-based and other methods have been developed in the past [3][1]. This is (mostly) not possible when trying to detect drift without access to labels (unsupervised method). When there is no direct feedback available the only possibility is to monitor data drift. As previously mentioned the assumption often used is that if data drift is detected this might indicate a possible concept drift [2]. Some unsupervised methods compute the difference or distance in distributions [4] whereas others use statistical tests such as the Kolmogorov-Smirnov test which can be applied to different aspects of a model (e.g. input attributes, final decision, or predicted probability) [5] [6]. A third possibility is to use a

semi-supervised method for detecting drift where there is limited access to labels such as the MD3 model [7].
Another point to be considered is the method-specific requirements that the model has to meet. Many sophisticated and well-designed drift detection methods are limited in their applicability because they require the use of a specific classifier or have some other restrictions pertaining to the model the drift detection will be applied on. An example of such limitations was the initial version of the Margin Density Drift Detection (MD3) method which required a SVM classifier to be part of the ML model [8].
An additional aspect that should be considered is the type of data and how data is gathered. There are different methods available for batch and online data. In some environments, data has to be processed quickly and memory retention might be limited. Thus, this should also be taken into account. To summarise, these three aspects might be considered when choosing an appropriate drift detection method:

1. Label availability
2. Model specification
3. Online vs batch data

These aspects have been considered in this project and the considerations made are described in section 3.1.

2. Data

The CNN model provided by the ADR was trained to classify transactions having one text feature as input. Transactions belonged to one of five classes (see table 1). The goal of the model was to aid auditors in their work. The model would make probabilistic predictions as to what classes a transaction belonged to. The class with the highest probability for a transaction was selected. The (top 50) transactions that had the lowest probabilities (i.e. high uncertainty) on their predicted class were presented to auditors who investigated if any misstatements were made. The data set used to train the model is accounting data used by the Dutch Central Government Audit Service (CGAS). The CGAS uses this data to conduct the external audit of the Dutch ministries. The data set consists of transactions recorded in 2017 and 2018 in the general ledgers of two Dutch ministries. It has been pre-processed by a third party (who also trained the model) to only include the ministries' expenses. The original raw data set was not accessible anymore. The shape was (761694,119) where each row represented a transaction with categorical, numerical and text features offering different kinds of information about the transaction (see table 2.1). The provided model used the text variable 'text' which was constructed by combining four text features into one larger text feature as input. The new unseen data used for the drift detection were transactions recorded in 2020. This year was of particular interest due to the spread of COVID-19 in Europe and its influence on public and private structures. The 2020 data set can be considered to be fairly similar to the 2017/2018 and additional processing was done to establish an adequate level of comparability. There were variables, however, which were present in the 2020 data set but not in the older one and vice versa. Additionally, some of the variable names have been altered to some degree and the 2020 data set was considerably smaller (see table 2.1). As the goal of this project was to detect drift in the first half of 2020, the 2020 data was

sorted by the date of booking, and the first 4 months (first 160 000 transactions) were selected for drift detection. All the preprocessing done within this project can be viewed in the Jupyter Notebook.

Year	Shape	Selected Features (X)	Target (y)	Usage
2020	(457039, 112)	4 features merged into 1 text feature	5 classes	Stage 2
2017/2018	(761694,119)	4 features merged into 1 text feature	5 classes	Stage 1

Table 2.1: Overview of data used

3. Methodology

3.1 Selection of method

The data and the CNN model to be monitored for drift required an unsupervised approach as labels were not available at all or were extremely delayed. This has to do with the fact that acquiring labels in a financial audit setting is very time intensive. Thus, an unsupervised method was needed. As the classifier was trained within a CNN infrastructure and the stakeholder might change or use a different model in the future, a model-independent method was preferred. This ensures that the proposed method can be applied within different projects and settings. Additionally, there is no memory limitation so the data is presented all at once. A batch oriented method is the first choice. Additionally, the drift detection method should not have a high false positive rate as each alarm that is triggered requires new labels to re-train the model which is quite time and cost-intensive in a financial audit setting if performed very often.

3.2 STUDD

Taking into account all these aspects, the recently developed STUDD method was selected [2]. STUDD, short for "Student–Teacher approach for Unsupervised Drift Detection", is using a student-teacher paradigm to detect drift and offers a way to detect drift in an unsupervised setting i.e., without labels. It consists of two stages: the student-teacher training (see Fig. 3.1) and the change detection stage (see Fig. 3.2).
During the first stage a teacher model (T) is trained on training data (X_{tr}, y_{tr}) and is afterwards used to generate predictions $\hat{y}_{tr}$. These predictions are used to create a new training set consisting of the original X_{tr} and the newly generated $\hat{y}_{tr}$ thereby replacing the original labels y. This new training set

is then used to train a student model S. Having X_{tr} and $\hat{y}_{tr}$ as its input, the new student model "learns" to mimic the behaviour of the teacher model. It is important to use the same X_{tr} for the training of the two models. Thus, in this stage of the method the same data set is required. Additionally, it has to be a labelled data set as labels are necessary to train the teacher model (T).

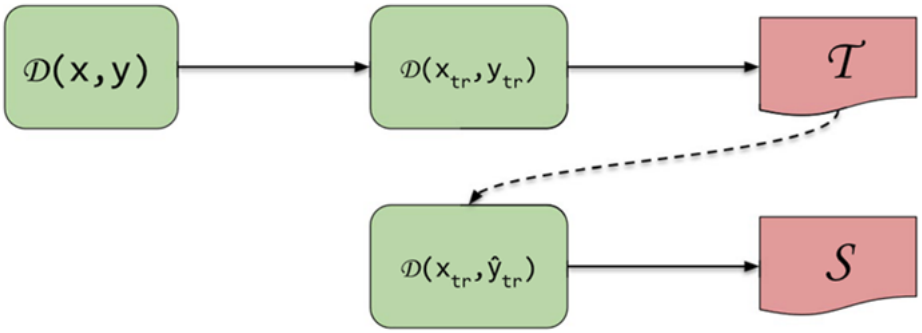

Figure 3.1: Student–teacher training (Stage 1)

During the second stage, the actual drift detection takes place. Here, new, unseen data which has no labels and only consists of X_i is passed to both models. Both models then generate predictions based on X_i, namely $\hat{y}_{i,T}$ and $\hat{y}_{i,S}$. These predictions are then compared and a loss function L_i is calculated based on the error rate. It should be noticed that this loss is attributed to the student model as its performance is being evaluated by comparing predictions $\hat{y}_{i,T}$ and $\hat{y}_{i,S}$. This loss function is then passed to the drift detector of choice that monitors the loss over time and detects changes that deviate from the norm in some way.

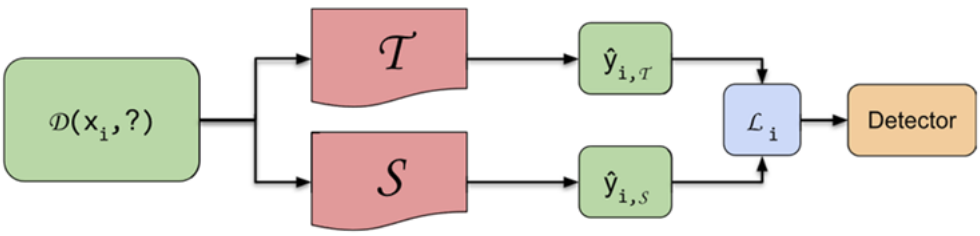

Figure 3.2: Change detection (Stage 2)

3.3 Assumptions

Although the method is model-independent and allows flexibility in how the teacher and student model are trained, it does have one assumption that should be met. Both models should have good predictive performance. For the teacher model, this might be intuitive as it is supposed to do its task reasonably well. The same is required from the student model as its good predictive performance is crucial for the STUDD method to work adequately.

3.4 Implementation and Changes

As the developers of the STUDD method made the code available online it was possible to use it and adapt it to this project. However, it had to be somewhat modified as some parts of the code were concerned with the evaluation of the STUDD model which was outside the scope of this project. In the end, only the code for the STUDD method was used and no post-drift adaptation (e.g. re-training of the model) was employed. (preprocessing) Originally, the method only used the same Datastream (X,y) for both stages. In the first stage, the first n observations from the Datastream were used for the training of the teacher and student model. The rest of the Datastream was used for the detection phase (stage 2). During this project, instead of training a new teacher model T with the first n observation in the Datastream (X,y), a CNN model that was trained on 2017/18 data (and provided by the ADR) was used in the first stage of the method. Accordingly, the student model was created based on the same CNN structure as the teacher model. Furthermore, for the detection stage, the 2020 data set was used to create a Datastream (X,y). As the research question was mostly interested in detecting drift in the first half of the year (initial spread of COVID-19), only the first 160 000 observations from the Datastream were included in drift detection whereas the original method included the entire data set in the drift detection (except for the initial n observation making up the training data set). Apart from these changes, most parts of the original code concerning

the two stages were adopted. The same drift detector was used, namely the Page-Hinkley test, and the same parameters like the δ value were chosen.

3.5 Visualising Drift

To visualise the possible results the performance of the student model was plotted against time and drift events were represented by vertical lines. Two plots were created with different performance metrics, namely Accuracy and Cohen's Kappa. The accuracy metric is a Binary or Class-specific classification agreement measure that is computed for each class of interest and averaged over k classes, whereas Cohen's Kappa is a multi-class classification agreement measure. Both have 0 as the lowest value and 1 as the highest value while Cohen's Kappa could technically also be negative. A drift event is expected to be accompanied by a drip in performance on both metrics. This would indicate a discrepancy between the predictions of the teacher and student model. This discrepancy between predictions would be a result of the data drift which causes the teacher model to deviate from its usual behaviour and generate (by the student model) unexpected predictions.

4. Results

While the predictive accuracy of the pre-trained CNN model was quite low (below 0.8), the student model's performance was adequate for its purpose (above 0.8). This means that although the pre-trained CNN model might not predict the type of transactions well, the student model at least "mimics" the teacher model moderately well. To visualise the drift, plots were created as described in. The plots in this section depict the first 50 000 transactions of 2020 as drift was detected very early on and a plot with a smaller x-axis allows to zoom in on the first drift events. The performance plots with all 160 000 transactions can be found in the appendix (see Fig. 2 and Fig. 3). In the detection phase the method detected 33 drift events in total (see table 2) with the first being detected as early as January 2020 which was accompanied by a visible drop in the performance of the student model on both performance measures: accuracy (see Fig. 4.1) and Cohen's Kappa (see Fig. 4.2). After the first few drift events, additional drift events that are detected are not accompanied by significant performance drops (compared to the first few ones) with the performance seeming to stabilise after the first few drift events and remaining relatively low. The corresponding booking dates of the transactions, where data drift was detected, and the performance plots including all 160 000 transactions analysed can be found in the appendix.

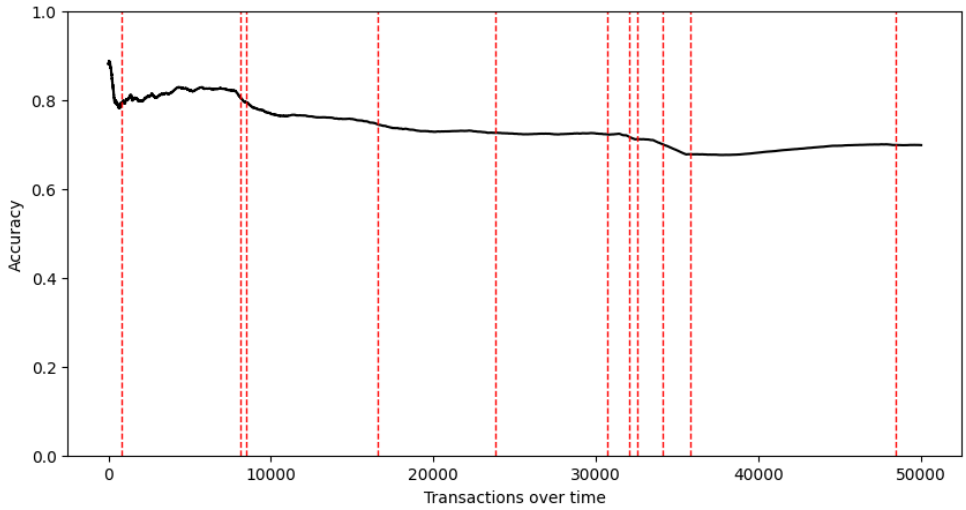

Figure 4.1: Accuracy of student model (black graph) and drift events (red lines) within the first 50 000 observations (corresponds to 10-02-2020)

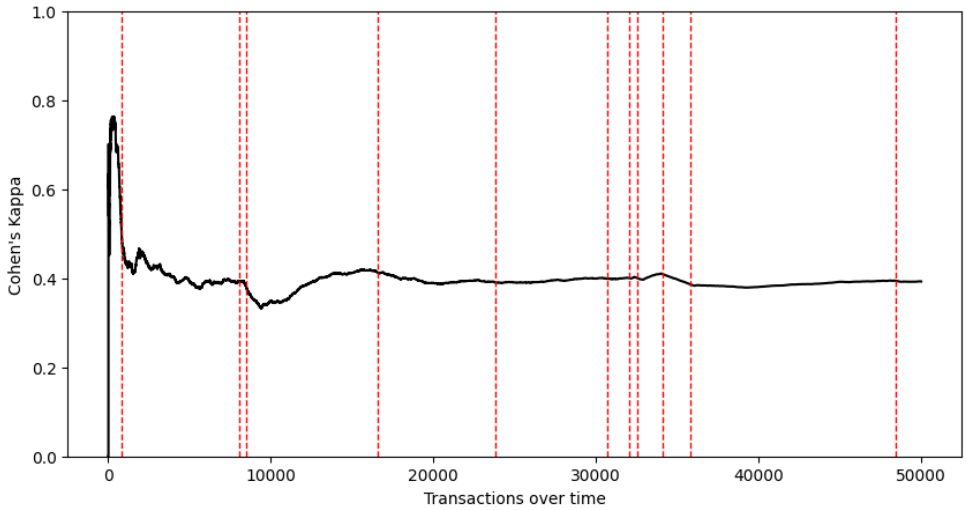

Figure 4.2: Cohen's Kappa (performance measure) of the student model (black graph) and drift events (red lines) within the first 50 000 observations (corresponds to 10-02-2020)

5. Discussion & Conclusion

The hypothesis raised at the start of this project was only confirmed partially. As hypothesised, the STUDD method detected several drift events. The first drifts occurred within the first 2 weeks of January and resulted in a drop in performance on both performance metrics. The co-occurrence of a drift event and a decrease in performance is in line with the content discussed in and the results of the developers of the STUDD method [2].
As the first drifts already occurred so early on, it seems unlikely that they could have been caused by or correlated with the spread of COVID-19 which only triggered a large-scale governmental reaction a few months later. Thus, there might be other causes for the observed drift in the data, besides COVID-19. One possibility is that the data drift already occurred before 2020 e.g. 2019 and the first drift in January might be attributed to a fundamental difference between the 2017/18 and the 2020 data sets, not connected to COVID-19. A more gradual drift with smaller changes accumulating during the year 2019 could also be a possible explanation.
The fact that drift has been detected early on in the data and that no kind of adaptation took place afterwards, might render the drift events that were detected later on less meaningful as the performance of the student model failed to reach its initial level again (see Fig. 4.1 and Fig. 4.2). In other words, the student model might not be able to 'mimic' the teacher's model's behaviour that closely and accurately anymore and additional drift events detected do not offer further meaningful information. Also, due to the decrease in performance, the assumption of a strong student model is violated. Thus, it can not be conclusively established whether the spread of COVID-19 did trigger additional drift later in the data.
Nevertheless, as data drift was detected this could be indicative of content drift as mentioned previously. Thus, some kind of adaptation should be implemented (e.g. re-training of the model), otherwise, the model might

not be able to capture the present data adequately and the performance will decrease. For this step, labels might be required.

5.1 Limitations

As there were time constraints placed upon this project, the performance of the STUDD method was not specifically evaluated for the data set used in the analysis nor accounting data in general. This presents a fundamental limitation that should be addressed before this method is further used in the specific setting of financial auditing. The performance evaluation should be conducted using a labelled data set (e.g. 2017/18) where the ground truth is known. This would allow the STUDD method to be compared to other supervised performance-based drift detection methods as described in the article by who developed the method.
Additionally, the assumptions of the method might not have been met as the performance accuracy of the pre-trained CNN model was not very high even on the 2017/18 data set. This clashes with the assumption of a strong teacher model as described by the developers of the method [2]. This project relied on the pre-trained CNN model due to time constraints put in place, but going forward it would be better to train the student and teacher model at the same time to meet the before-mentioned assumptions and guarantee adequate performance of the STUDD method.
Furthermore, to be able to establish more conclusively whether the COVID-19 pandemic might have caused drift in the data, 2019 data could be included in the analysis and drift adaptation should be performed when drift is detected to ensure adequate performance of the teacher model. This would make results more clear and the relationship between the possible occurrence of drift and the influence of the COVID-19 pandemic more interpretable. Additionally, there might be differences across ministries in the degree of influence the pandemic had on their financial processes e.g. the health ministry might have been affected more than the ministries considered in this analysis. This should also be taken into account going further.

5.2 Future possibilities

After addressing the above-mentioned limitations and evaluating the STUDD method on data used in a financial audit setting, the method can be used for different purposes beyond the one presented in this paper. The proposed method could be employed for anomaly detection which is of high interest in a financial audit setting. Furthermore, it could be integrated into the general process when building any kind of ML model that has limited access to labels. This would ensure the quality of the model deployed.

Acknowledgements

I want to thank my supervisors Sjoerd Dirksen, Kees Oosterlee and Fré Vink for offering support and guidance throughout this project. Moreover, I want to thank all my friends and family who have continuously encouraged and motivated me. Thank you for the frequent check-ups and the emotional support. Lastly, special thanks go out to my mum who was patient with me and supported me throughout my studies and made this project possible.

References

[1] A. K. Firas Bayram Bestoun S. Ahmed, "From concept drift to model degradation: An overview on performance-aware drift detectors," *Knowledge-Based Systems*, 2022. DOI: 10.1007/s10994-022-06188-7.

[2] V. Cerqueira, H. M. Gomes, A. Bifet, and L. Torgo, "Studd: A student–teacher method for unsupervised concept drift detection," *Machine Learning*, Jun. 2022. DOI: 10.1007/s10994-022-06188-7.

[3] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, pp. 1–37, 2014.

[4] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*, pp. 443–448. DOI: 10.1137/1.9781611972771.42.

[5] D. M. dos Reis, P. Flach, S. Matwin, and G. Batista, "Fast unsupervised online drift detection using incremental kolmogorov-smirnov test," 2016. DOI: 10.1145/2939672.2939836.

[6] I. Žliobaité, "Change with delayed labeling: When is it detectable?," pp. 843–850, 2010. DOI: 10.1109/ICDMW.2010.49.

[7] T. S. Sethi and M. Kantardzic, "On the reliable detection of concept drift from streaming unlabeled data," *Expert Systems with Applications*, vol. 82, pp. 77–99, 2017. DOI: 10.1016/j.eswa.2017.04.008.

[8] R. N. Gemaque, A. F. J. Costa, R. Giusti, and E. M. dos Santos, "An overview of unsupervised drift detection methods," *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 6, 2020. DOI: 10.1002/widm.1381.

Appendices

Data

The following two figures were generated by Stijn Uijen who built the CNN model.

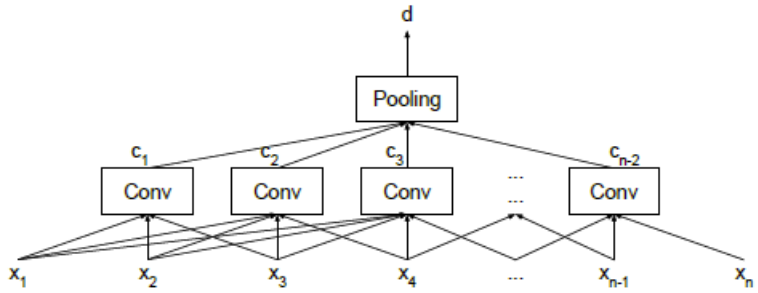

Figure 1: CNN-based Text Representation

	Overall		Ministry A		Ministry B	
Class Label	Frequency	%	Frequency	%	Frequency	%
Staff Costs	331,822	43.6	151,520	40.2	180,302	46.9
Purchase of Goods and Services	284,275	37.3	80,005	21.2	204,270	53.1
Program Expenses	139,045	18.3	139,045	36.9	0	0.0
Depreciation and Impairment	4195	0.6	4195	1.1	0	0.0
Interest Costs	2357	0.3	2357	0.6	0	0.0
Total	761,694	100.0	377,122	100.0	384,572	100.0

Table 1: Distribution of the transactions across classes in the 2017/18 data

Results

Transaction	Date of booking
793	06.02
8090	10.01
8455	11.01
16549	20.01
23806	25.01
30724	31.01
32045	01.02
32517	01.02
34124	01.02
35829	01.02
48406	07.02
60078	17.02
64924	21.02
66042	22.02
76931	28.02
78008	29.02
86847	06.03
91431	09.03
101452	13.03
103846	16.03
111987	23.03
116426	25.03
119412	27.03
121568	30.03
125089	01.04
129583	06.04
138733	09.04
141038	13.04
147970	20.04
149887	23.04
151830	23.04
153156	24.04
154346	24.04

Table 2: Drift events detected by the STUDD method

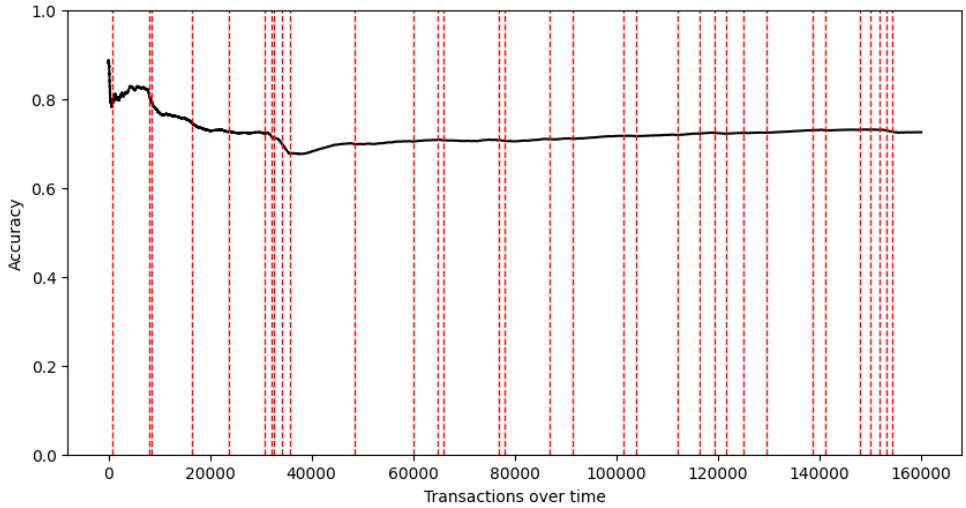

Figure 2: Accuracy of student model (black graph) and drift events (red lines) within the first 160 000 observations (corresponds to 01-05-2020)

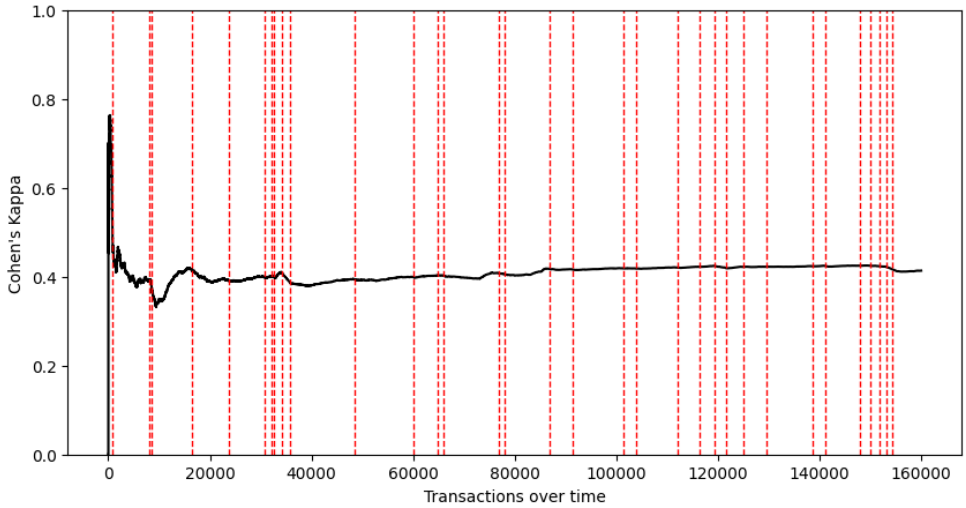

Figure 3: Cohen's Kappa (performance measure) of student model (black graph) and drift events (red lines) within the first 160 000 observations (corresponds to 01-05-2020)