DriftWatch: A Tool that Automatically Detects Data Drift and Extracts Representative Examples Affected by Drift

Myeongjun Erik Jang[1,2] **Antonios Georgiadis**[2] **Yiyun Zhao**[2] **Fran Silavong**[2]
[1]Department of Computer Science, University of Oxford
[2]J.P. Morgan Chase
myeongjun.jang@cs.ox.ac.uk antonios.georgiadis@jpmchase.com
yiyun.zhao@jpmchase.com fran.silavong@jpmchase.com

Abstract

Data drift, which denotes a misalignment between the distribution of reference (i.e., training) and production data, constitutes a significant challenge for AI applications, as it undermines the generalisation capacity of machine learning (ML) models. Therefore, it is imperative to proactively identify data drift before users meet with performance degradation. Moreover, to ensure the successful execution of AI services, endeavours should be directed not only toward detecting the occurrence of drift but also toward effectively addressing this challenge. In this work, we introduce a tool designed to detect data drift in text data. In addition, we propose an unsupervised sampling technique for extracting representative examples from drifted instances. This approach bestows a practical advantage by significantly reducing expenses associated with annotating the labels for drifted instances, an essential prerequisite for retraining the model to sustain its performance on production data.

1 Introduction

The recent advancements in machine learning (ML) and deep learning (DL) have propelled the emergence of diverse natural language processing (NLP) AI solutions featuring cutting-edge ML and DL models. Nonetheless, their exclusive proficiency in inductive reasoning has given rise to substantial challenges when applied in practical business contexts. One such challenge is a *data drift*, an inconsistency between reference (i.e., training) and production data distributions (Madaan et al., 2023). As the alterations in data distribution violate the fundamental assumption of ML, the IID condition that posits an identical distribution between training and test data, the occurrence of data drift has the potential to aggravate the accuracy of previously-trained models and ultimately damage the quality of AI services. Consequently, it is crucial to detect data drift and provide an updated model before customers experience a degradation in performance.

The ML community has classified the data drift into two principal categories (Moreno-Torres et al., 2012; Gama et al., 2014; Mallick et al., 2022). Assume an input X, target Y, and the ground-truth relationship between X and Y as f, so that $Y = f(X)$. The first type of data drift is *covariate drift* (Shimodaira, 2000), which implies the change in the input feature distributions (i.e., $X \rightarrow X^{'}$). The second category is *concept drift* (Widmer and Kubat, 1996), where the underlying relationship f changes (i.e., $Y = f(X) \rightarrow Y = g(X)$). These two types of data drift readily occur in practical applications, such as introducing instances with unseen target labels or emerging new words/phrases under existing target labels. However, previous studies regarding data drift detection solely focused on a singular drift type, either covariate drift (Feldhans et al., 2021; Khaki et al., 2023; Chang et al., 2023; Madaan et al., 2023) or concept drift (Ackerman et al., 2020; Tahmasbi et al., 2021; Ackerman et al., 2021; Rabinovich et al., 2023). Furthermore, these studies primarily centred on the identification of drift, but from a practical viewpoint, it is equally crucial to effectively address the challenge of upholding the model's performance and the quality of AI services. The conventional and straightforward approach involves annotating the drifted instances and incorporating them in a training batch for model retraining. However, employing human annotators for labelling a substantial volume of data points constitutes a resource-intensive undertaking.

To this end, we propose a system called *DriftWatch*, which detects both covariate and concept drift in text data. Regarding the detector for covariate drift, we ascertained that using both semantic and syntactic features is beneficial over the exclusive reliance on either. Regarding the detector for concept drift, we investigated multiple approaches, including the incorporation of large language mod-

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), pages 335–346

els (LLMs), but found that conventional and simpler methods outperform the LLM-based approach in practical applications. In addition to this, we built a sampling methodology that autonomously extracts representative drifted instances, along with their corresponding importance rankings. This unsupervised approach can significantly reduce the effort to annotate labels for drifted instances, a necessity in the re-training ML model.

The main contributions of this paper can be summarised as follows: 1) We introduce an auditor capable of detecting both covariate and concept drift, 2) We propose an effective sampling approach for the extraction of representative samples from drifted instances, which offers the practical advantages by significantly reducing the effort required for annotating labels, 3) Our sampling methodology provides importance rankings for the drifted instances, facilitating prioritising annotation orders in the situation of limited resources, 4) We ascertain that contemporary LLMs may not necessarily outperform traditional approaches when implemented in practical applications.

2 Components of Proposed Solution

The overall process of *DriftWatch* solution is illustrated in Figure 1. First, models consisting of our covariate and concept drift detectors are trained using the reference dataset. Next, production instances affected by both covariate and concept drift are predicted using the trained models. Finally, representative sampling is introduced to address a practical issue where enough human labourers to annotate drifted instances are unavailable. Finally, newly annotated instances are integrated into the reference data.

2.1 Covariate Drift Detector

Syntactic Drift Detector. Following the work of Chang et al. (2023), we employed vocabulary drift to detect syntactic changes in input features. To elaborate, content words [1] were extracted from the training corpus and the frequency of each word was calculated. Subsequently, the likelihood of an instance x ($\mathcal{L}_x$) is defined as the logarithmic summation of the frequencies of content words contained in x:

$$\mathcal{L}_x = \frac{1}{|x_c|} \sum_{w \in x_c} \log F(w), \tag{1}$$

where x_c refers to the content words existing in x and $F(w)$ denotes the frequency of the word w. The low likelihood indicates that an instance contains many content words absent from the training corpus, signifying dissimilar input features. Consequently, instances are deemed drifted instances when their likelihood falls below a predefined threshold.

On top of the likelihood, the syntactic drift detector offers the contribution score of each content word to covariate drift. Assume that an input x is identified as a drifted instance owing to a low likelihood. As words with higher frequency have less influence on covariate drift, we defined the contribution score of a content word w in x as follows, where higher values imply a greater contribution to the drift.:

$$c_w = \frac{\hat{c}_w}{\sum_{k \in x_c} \hat{c}_k}, \hat{c}_w = \frac{\mathcal{L}_x}{\log F(w)}, \tag{2}$$

Semantic Drift Detector. We referred to the variational auto encoder (VAE) based density modelling approach to identify semantic alterations in input features, which, as demonstrated in the study of Madaan et al. (2023), exhibited superior performance over alternative approaches. During the training phase, sentence vectors are generated from S-BERT to train a VAE. In the inference phase, VAE generates the loss value for an instance x, which is then employed to compute the similarity score: $s_x = e^{-loss}$. The low similarity score indicates a failure of the VAE to reproduce the input vector representation, signifying its dissimilarity with training instances. Consequently, instances with a similarity score below a predefined threshold are regarded as drifted instances.

Our semantic drift detector also offers the contribution of each word to covariate drift. Consider an input sentence x consisting of n words is given. First, n masked sentences are generated by masking a single word at a time. Next, VAE generates similarity scores for x and masked sentences. Finally, the contribution of ith word is computed as follows:

$$D_i = \frac{s_i - s}{\sigma}, c_i = \frac{e^{D_i}}{\sum_{k=1}^{n} e^{D_k}}, \tag{3}$$

where σ denotes the standard deviation of training similarity scores, s and s_i refer to the similarity score of x and a masked sentence where ith word is masked, respectively. If $D_i > 0$, it means that the similarity is increased after making ith word,

[1] Noun, verb, adverb, and adjectives.

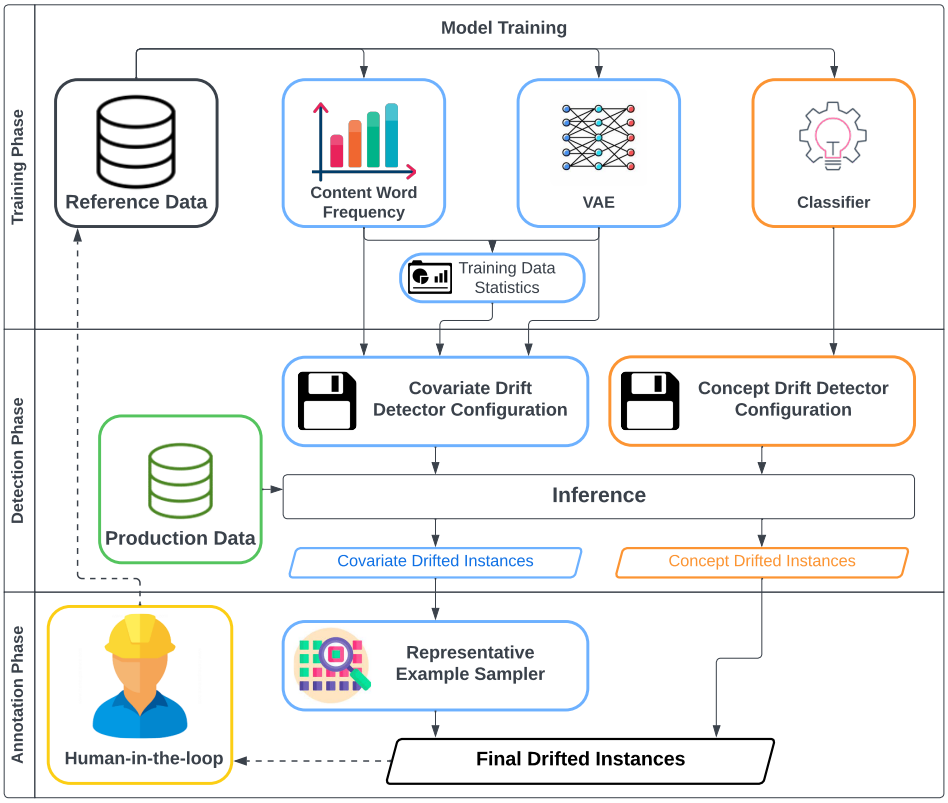


Figure 1: Overall process of DriftWatch solution.

signifying that the word negatively influences the similarity score and, hence, the word contributes more significantly to covariate drift.

Both drifted detectors require a predetermined threshold for decision-making. We defined the threshold as $\mu_{tr} - 3\sigma_{tr}$ following the Six Sigma method for quality control. The μ_{tr} and σ_{tr} represent the average and standard deviation derived from the likelihood and similarity score training distribution.

2.2 Concept Drift Detector

Predictive Entropy Approach. Concept drift denotes an alteration in the relationship between X and Y. As a classifier M is trained to formulate an empirical relationship between X and Y, i.e., $Y = M(X)$, the predictive distribution generated by the classifier has been conventionally employed for detecting concept drift. Building upon the work of Winter et al. (2023), we employed the entropy of the predictive distribution as a metric for identifying the concept drift:

$$\mathcal{H}_x = -\sum_{k \in C} p_M(y = k|x) \log p_M(y = k|x), \quad (4)$$

where $\mathcal{H}_x$ denotes the entropy of an instance x, $p_M(y = k|x)$ refers to the predictive probability of x having the label k generated by M. The higher entropy implies that the predictive distribution closely approximates a uniform distribution, suggesting an increased likelihood of concept drift. The drift detection performance can be further enhanced by employing the ensemble method, incorporating distributions generated by multiple classifiers. (Lakshminarayanan et al., 2017):

$$p_E(y = k|x) = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} p_m(y = k|x), \quad (5)$$

where $\mathcal{M}$ is the set of pre-trained classifiers.

2.3 Representative Example Sampler

Our sampling methodology consists of two components: a feature extraction module and a clustering-based sample extraction module.

Feature Extraction Module. We transformed text data into numerical vectors through the proposed feature extraction module. First, a sentence embedding model was employed to generate sentence vectors for each input text. Subsequently, the dimension of sentence vectors underwent reduction through a dimensionality reduction

methodology for efficient clustering. We used S-BERT (Reimers and Gurevych, 2019) for generating sentence vectors (all-MiniLM-L6-v2) and UMAP algorithm (McInnes et al., 2018) for the dimensionality reduction, where the size of reduced dimension was set as 10.

Sample Extraction Module. We employed the K-means clustering-based sampling approach (Chang et al., 2021) to extract representative examples. Assume that we have a total of n data instances, with the designated number of samples for extraction denoted as N. Utilising the output of the aforementioned feature extraction module, K-means clustering was performed where K is set to N to minimise the sum of the squared errors (SSE):

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{N} w_{i,j} ||x_i - \mu_j||^2, \qquad (6)$$

where μ_i is the centroid of the jth cluster, x_i is the embedding vector of ith instance, $w_{i,j}$ is 1 if x_i belongs to the jth cluster and 0 otherwise. The clustering process was iterated 10 times with different initial centroids, and the outcome yielding the minimum SSE was selected. Finally, N data points closest to each cluster's centre were extracted as representative examples. In addition to the representative samples, our solution also provides their respective importance scores. This information proves valuable for prioritising the annotation order, especially when annotation resources are constrained. Given that a cluster with smaller SSE implies instances within the cluster are densely concentrated, the centroid of such a cluster encompasses more similar instances. Also, clusters containing fewer instances have lower SSE values. Therefore, the SSE of each cluster divided by their size served as an importance score, where lower values indicate higher importance.

3 Experiments and Results

3.1 Publicly Available Dataset

We first assessed our proposed solution on publicly available datasets to ascertain the basic performance of our proposed drift detectors.

Covariate Drift Experiment. We employed Insurance company review [2] as a reference data. For the production set, we constructed two sets where the Insurance company review was mixed with Fashion item review (Agarap, 2018) and Restaurant review [3]. The details of the training are described in Appendix A.1.1. Table 1 displays the experimental results. The same evaluation metric proposed by Madaan et al. (2023) was used for the evaluation. We ascertained that our solution, which leverages both semantic and syntactic drift detectors, outperforms DetAIL (Madaan et al., 2023), a practical service that is currently operating. Also, it was observed that employing both detectors exhibits better performance compared to using only one type of detector, signifying the benefit of utilising multiple distinct features.

		+Fashion	+Restaurant
DriftWatch	Semantic	94.15±0.7	73.23±0.6
	Syntactic	82.08	73.62
	Both	**96.32**±0.2	**82.08**±0.3
DetAIL (Reported)		96.18	81.57

Table 1: Experimental results on the review datasets. The best results are formatted in bold. The average and standard deviation of five repetitions are reported.

Additionally, we employed the contribution scores to identify words that highly influenced the data drift and found that the syntactic drift detector scores were more intuitive than those of the semantic drift detector. The examples can be found in Figure 4 in Appendix.

Concept Drift Experiment. We used AG-News dataset (Zhang et al., 2015) for the concept drift detection experiments. We investigated four scenarios where one of the classes is removed from the training data. Test instances with the removed class as labels were considered drifted examples. Regarding the single approach, an average of five repetitions was reported. For the ensemble approach, the predictive distribution of the five single models was merged by Equation 5. The results are summarised in Table 2. It was observed that the ensemble method generated a significantly higher AUROC score than the single model approach, even performing better than the best-performing single model. The results signify that the ensemble method, which produces more stable and consistent performance, would be a safer approach in practical applications.

[2] Kaggle insurance company review data

[3] Kaggle restaurant review data

Removed Class	World	Sports	Business	Sci/Tech
Single Avg.	.810±.03	.649±.02	.821±.01	.791±.01
Ensemble	**.850**	**.693**	**.840**	**.802**

Table 2: Experimental results on the AG-News dataset for concept drift experiment. The best results are formatted in bold.

3.2 Real Practical Scenario

Next, we applied *DriftWatch* to a real-world industrial scenario by using our internal customer complaint dataset (ICD). This dataset consists of two textual components: summaries of customer complaints and the resolutions provided by our customer service agents, along with their corresponding 3-level hierarchical categories, which were labelled by human annotators. We concatenated the two texts to create an input, where the category served as the prediction label. The dataset spans all days of 2022. We partitioned Jan data as a reference set and constructed 11 production batches based on the respective months in which the data was collected.

Covariate Drift Detection Results. We devised an indirect experiment for evaluating performance due to the complexity of labelling covariate drifted instances in real data. This involved segregating the data into training, validation, and test sets for each month, excluding Jan, the reference data. Subsequently, our proposed solution was applied to the training set of each month to detect instances affected (D) and unaffected (¬D) by covariate drift. Next, two auxiliary training sets were formulated for each month: D+Rand(¬D), where all the drifted instances were used, and additional examples were randomly sampled from ¬D, and Rand(D+¬D), where all the examples were randomly sampled. The size of the two sets was identical to 10K. Finally, a classifier was trained for each month, utilising both the reference set and the auxiliary training set, and the performance on the test set was compared. We used the 1st level category as a target label (20 classes) and fine-tuned Electra-small model (Clark et al., 2020). Appendix A.1.1 describes more details regarding the training settings. Experimental results are summarised in Table 3. The findings indicate that the incorporation of all the drifted instances yields statistically significant improvements in performance across 8 out of 11 months, suggesting that the identified drifted instances exhibit distinctive features that impede the generalisation effect. Furthermore, we analysed the word contribution scores and ascertained that typos and abbreviations largely influenced to the covariate drift. The examples are not included in the manuscript due to the security issue.

Representative Sampling Results. Through the application of our representative sampling method, we selectively extracted 50% of examples affected by covariate drift, subsequently integrating them with the reference data for training a classifier for each month. For comparative analysis, all instances affected by covariate drift were integrated with the reference data. Table 3 shows that classifiers trained with sampled examples, despite being trained on a reduced dataset, demonstrated no statistically significant performance degradation overall and even exhibited superior performance in the datasets corresponding to June and July.

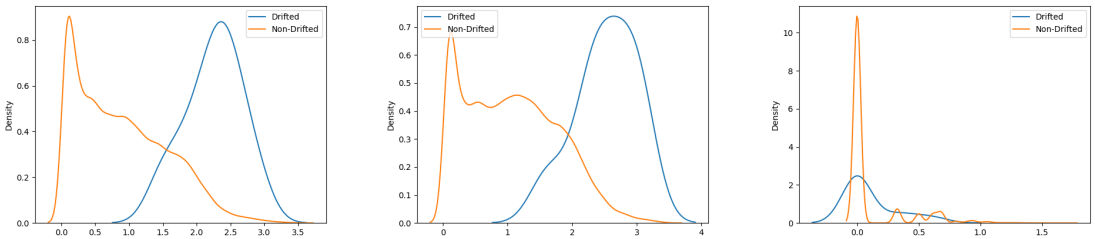
We additionally trained the classifiers on two variations to ascertain whether the importance score conveyed meaningful information. Specifically, we split the sampled representative examples into two groups: half of the examples with the highest importance (H-Imp) and the others (L-Imp). It was found that groups with higher importance produced superior performance in general, supporting the benefit of the proposed importance score.

Concept Drift Detection Results. Instances characterised by labels absent in the reference data were deemed as examples influenced by concept drift. Given the absence of instances having the new label in the 1st level category, we employed the 2nd level category as the target class, encompassing 47 subcategories. The results are summarised in Table 3. It was observed that, while the single approach yielded a decent level of performance, the ensemble approach employing five distinct classifiers exhibited a superior and more stable performance, with an average AUROC of 0.883±0.06, far surpassing that of the single approach of 0.821±0.09. Figure 3 in the Appendix illustrates the ROC curve of both approaches.

On top of our proposed approach, we implemented a concept drift detection method that employs LLMs. The recent advancements in LLMs have opened avenues for zero-shot data drift detection. This involves querying LLMs whether a given input exhibits an abnormal state, with specific applications in autonomous driving (Elhafsi et al., 2023) and log anomaly detection (Qi et al., 2023). These methodologies, however, lack appli-

		Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Covariate Drift	# of Drifts (Detected)	765	981	850	830	878	733	874	982	879	720	752
	Rand(D+¬D)	.669	.669	.651	.649	.658	.665	.689	.661	.672	.678	.687
	D+Rand(¬D)	**.675***	**.675***	**.652**	**.656***	**.660**	**.671***	**.694***	**.667***	**.679***	**.687***	**.689**
Representative Sampling	All	.660	.662	.631	**.639**	.627	.638	**.662**	**.661**	.665	**.678**	.685
	Kmeans 50%	**.662**	**.665**	**.632**	.636	**.637***	**.646***	.659	.660	**.666**	.677	**.686**
	L-Imp	.663	.659	.626	.628	.629	.640	.661	.657	.663	.674	**.685**
	H-Imp	**.665**	**.661**	**.633***	**.637***	**.631**	**.644***	**.664**	**.662***	**.667***	**.676**	.684
Concept Drift	# of Drifts	5	13	17	6	12	14	31	6	12	3	4
	Single	.743	.814	.944	.883	.927	.729	.744	**.796**	.897	.867	.683
	Ensemble (n=5)	**.909***	**.830***	**.949**	**.890**	**.938***	**.839***	**.780***	.790	**.900**	**.953***	**.940***

Table 3: Experimental results on ICD. The best performance is highlighted in bold. The evaluation metric for concept drift is the AUROC, and the F1-score for the others. We reported an average of five repetitions for each test scenario. * denotes the performance showed a statistically significant difference at a p-value of 0.1 using a t-test. '# of Drifts' in covariate drift is driven from the identified drifts by our tool, while that of concept drift is calculated by using the ground-truth labels.



(a) Entropy-Single (b) Entropy-Ensemble (c) Prompt Consistency

Figure 2: Entropy distribution of (a) single, (b) ensemble, and (c) prompt consistency approach.

cability in certain business domains as they rely on general knowledge for defining abnormal states. Hence, we devised a novel approach that leverages prompt consistency (Zhou et al., 2022). In particular, we used diverse prompt designs to fine-tune a LLM to generate the target label. We assumed that non-drifted instances would exhibit robust generalisation on the fine-tuned LLM, resulting in the model generating consistent answers across various prompt designs. Consequently, the prompt consistency score was employed to identify concept drift, which is defined as an entropy (equation 4) of the following predictive distribution:

$$p(y = k|x) = \frac{\sum_{i \in P} \mathbb{1}(LLM(x, i) = k)}{|P|}, \quad (7)$$

where P is the set of different prompt designs and $LLM(x, i)$ denotes the predicted label of an input x and the prompt design i.

We designed 10 different prompts (See Appendix A.2) and fine-tuned FlanT5-XL (Chung et al., 2022) with LoRA adaptation (Hu et al., 2022), where the details of the training are described in the Appendix A.1.3. Due to the excessive duration of the training FlanT5-XL, our experiments were confined to the Apr dataset, where optimal performance was observed for both the Single and Ensemble models. Notably, the LLM-based prompt consistency method yielded AUROC of 0.518, despite its 2 days fine-tuning period compared to the 1.5-hour duration for the ensemble approach. Figure 2 displays the entropy distribution, revealing that the prompt consistency approach produced an indistinguishable difference between instances affected by concept drift and those unaffected. The results signify that the modern LLMs may not necessarily be superior to conventional approaches in practical applications. The experimental results also indicate that the modern LLMs contain inconsistency issues, which is in line with many recent studies (Jang and Lukasiewicz, 2023; Teng et al., 2023; Bonagiri et al., 2024).

4 Related Works

Several studies have been conducted on covariate drift detection. Feldhans et al. (2021) generated sentence embeddings and performed statistical tests to detect changes in embedding vectors of reference and production data. Khaki et al. (2023) introduced a similar approach but used maximum mean discrepancy (MMD) test (Gretton et al., 2012). Ra-

binovich et al. (2023) employed an autoencoder, assuming that instances with high reconstruction errors are classified as outliers. They used change-point model (CPM) (Ross and Adams, 2012) to monitor whether a significant change in the reconstruction error of production data has occurred. Analogously, Madaan et al. (2023) proposed a framework named DetAIL, which leverages sentence embedding vectors for density modelling and detects covariate drift along with explanations. Chang et al. (2023) introduced a linguistic covariate drift detector that identifies changes in vocabulary usage, syntactic structure, and semantic meanings. Another line of works focused on identifying concept drift. A conventional approach is to use the confidence score of a winning label, which is generated by a pre-trained classifier, with a statistical testing (Ackerman et al., 2021) or CPM (Ackerman et al., 2020). Tahmasbi et al. (2021) implemented a supervised detection method that employs the performance of production data. Mallick et al. (2022) proposed an integrated framework for detecting and alleviating the data drift issue by finding a training batch that is the most similar to the production data and employing the model trained with the batch.

5 Conclusion

This paper introduces *DriftWatch*, a tool designed for the automated detection of data drift and the extraction of representative instances affected by such drift. The practical advantages of *DriftWatch* extend to industrial practitioners by facilitating proactive identification of data drift and reducing resources required for the annotation process for model re-training.

Limitations

As our representative sampling approach employs K-means clustering, the running time increases as the number of selected samples (i.e., K) grows. The duration can be regulated by employing smaller components for dimensionality reduction, but this may entail performance degradation. We conducted the LLM-based prompt consistency method only on the Apr dataset due to the excessive duration of fine-tuning LLMs, but our claim can be consolidated with more experimental results. Also, the proposed solution is applicable to text datasets, but it may not easily be adaptable to other types of data, which limits its generalisability.

Ethics Statement

The entire work presented in this manuscript adheres to the ACM Code of Ethics and Professional Conduct. Moreover, the internal review broadly assessed and approved the utilisation of the selected in-house dataset and the development of the proposed solution.

References

Samuel Ackerman, Eitan Farchi, Orna Raz, Marcel Zalmanovici, and Parijat Dube. 2020. Detection of data drift and outliers affecting machine learning model performance over time. *arXiv preprint arXiv:2012.09258*.

Samuel Ackerman, Orna Raz, Marcel Zalmanovici, and Aviad Zlotnick. 2021. Automatically detecting data drift in machine learning classifiers. *arXiv preprint arXiv:2111.05672*.

Abien Fred Agarap. 2018. Statistical analysis on e-commerce reviews, with sentiment classification using bidirectional recurrent neural network (rnn). *arXiv preprint arXiv:1805.03687*.

Vamshi Krishna Bonagiri, Sreeram Vennam, Priyanshul Govil, Ponnurangam Kumaraguru, and Manas Gaur. 2024. Sage: Evaluating moral consistency in large language models. *arXiv preprint arXiv:2402.13709*.

Ernie Chang, Xiaoyu Shen, Hui-Syuan Yeh, and Vera Demberg. 2021. On training instance selection for few-shot neural text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 8–13, Online. Association for Computational Linguistics.

Tyler Chang, Kishaloy Halder, Neha Anna John, Yogarshi Vyas, Yassine Benajiba, Miguel Ballesteros, and Dan Roth. 2023. Characterizing and measuring linguistic dataset drift. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8953–8967, Toronto, Canada. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Amine Elhafsi, Rohan Sinha, Christopher Agia, Edward Schmerling, Issa AD Nesnas, and Marco Pavone. 2023. Semantic anomaly detection with large language models. *Autonomous Robots*, pages 1–21.

Robert Feldhans, Adrian Wilke, Stefan Heindorf, Mohammad Hossein Shaker, Barbara Hammer, Axel-Cyrille Ngonga Ngomo, and Eyke Hüllermeier. 2021. Drift detection in text data with document embeddings. In *Intelligent Data Engineering and Automated Learning–IDEAL 2021: 22nd International Conference, IDEAL 2021, Manchester, UK, November 25–27, 2021, Proceedings 22*, pages 107–118. Springer.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Edward J. Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Myeongjun Jang and Thomas Lukasiewicz. 2023. Consistency analysis of ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15970–15985, Singapore. Association for Computational Linguistics.

Saeed Khaki, Akhouri Abhinav Aditya, Zohar Karnin, Lan Ma, Olivia Pan, and Samarth Marudheri Chandrashekar. 2023. Uncovering drift in textual data: An unsupervised method for detecting and mitigating drift in machine learning models. *arXiv preprint arXiv:2309.03831*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*.

Nishtha Madaan, Adithya Manjunatha, Hrithik Nambiar, Aviral Goel, Harivansh Kumar, Diptikalyan Saha, and Srikanta Bedathur. 2023. Detail: a tool to automatically detect and analyze drift in language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15767–15773.

Ankur Mallick, Kevin Hsieh, Behnaz Arzani, and Gauri Joshi. 2022. Matchmaker: Data drift mitigation in machine learning for large-scale systems. *Proceedings of Machine Learning and Systems*, 4:77–94.

L. McInnes, J. Healy, and J. Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*.

Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530.

Jiaxing Qi, Shaohan Huang, Zhongzhi Luan, Carol Fung, Hailong Yang, and Depei Qian. 2023. Loggpt: Exploring chatgpt for log-based anomaly detection. *arXiv preprint arXiv:2309.01189*.

Ella Rabinovich, Matan Vetzler, Samuel Ackerman, and Ateret Anaby Tavor. 2023. Reliable and interpretable drift detection in streams of short texts. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 438–446, Toronto, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Gordon J Ross and Niall M Adams. 2012. Two nonparametric control charts for detecting arbitrary distribution changes. *Journal of Quality Technology*, 44(2):102–116.

Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Ashraf Tahmasbi, Ellango Jothimurugesan, Srikanta Tirthapura, and Phillip B Gibbons. 2021. Driftsurf: A risk-competitive learning algorithm under concept drift. In *Proceeding of the International Conference on Machine Learning*. PMLR.

Zhiyang Teng, Ruoxi Ning, Jian Liu, Qiji Zhou, Yue Zhang, et al. 2023. Glore: Evaluating logical reasoning of large language models. *arXiv preprint arXiv:2310.09107*.

Gerhard Widmer and Miroslav Kubat. 1996. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23:69–101.

Anton Winter, Nicolas Jourdan, Tristan Wirth, Volker Knauthe, and Arjan Kuijper. 2023. An empirical study of uncertainty estimation techniques for detecting drift in data streams. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Chunting Zhou, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Prompt consistency for zero-shot task generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2613–2626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Appendix

A.1 Training Details

A.1.1 VAE for Covariate Drift

We trained a VAE consisting of two layers, i.e., a single input and output layer. The hidden dimension and latent dimension were set as 256 and 128, respectively. We applied Leaky Relu with a slope of 0.2. AdamW optimiser (Loshchilov and Hutter, 2019) was employed with the learning rate of $1e^{-4}$ and weight decay rate of 0.1. The model was trained for 100 epochs with a batch size of 64. An early stopping strategy was used to avoid overfitting if the validation loss did not decrease for three consecutive epochs. The same training setting was used for the experiments on publicly available datasets and our ICD. The models were trained by using a single Tesla T4 GPU.

A.1.2 Classifiers for Concept Drift and Sampling Experiments

For all experiments, we used the Electra-small model (Clark et al., 2020) as a backbone pre-trained language model. We set the maximum number of input tokens to 256. For the AG-News dataset, classifiers were trained for five epochs. When it comes to ICD, the training epoch was set to 10. Similar to VAE, AdamW optimiser (Loshchilov and Hutter, 2019) was used with the learning rate of $1e^{-4}$, the weight decay rate of 0.1, and a batch size of 64. The same early stopping strategy was adopted to avoid overfitting. A single Tesla T4 GPU was used for training the classifiers.

A.1.3 Fine-tuning LLM for Prompt Consistency

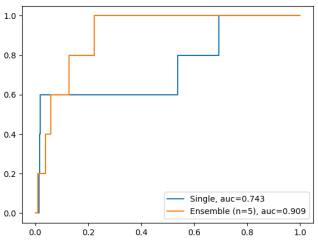
FlanT5-XL (Chung et al., 2022) was fine-tuned to generate the target label when a prompt containing an input sentence is given. The model was trained for one epoch with a batch size of four for each GPU. AdamW optimiser (Loshchilov and Hutter, 2019) was employed with the learning rate of $5e^{-6}$, weight decay rate of $1e^{-3}$, and warm-up ratio of 0.03. The number of maximum input tokes was set as 512. For efficient training, we applied LoRA adaptation technique (Hu et al., 2022). The LoRA hyperparameters r and α were set to 8 and 32, respectively. A dropout ratio of 0.1 is used. The model was trained by using four Tesla T4 GPUs.

Prompt Designs
(1) Define the categories for the given text below.\n{sentence}
(2) What is the topic of the given text below?\n{sentence}
(3) You will be provided with a customer's complaint and how it is addressed. Classify the given text into a primary category. \n{sentence}
(4) What would be the best category for the following customer complaint and resolve note?\n{sentence}
(5) For the following customer complaint and resolving note, what would have been the best category?\n{sentence}
(6) Which label best describes the following text?\n{sentence}
(7) We'll provide you with information on the customer complaint and how to deal with it. Indicate that the text is to be classified as a primary category.\n{sentence}
(8) The following sentence was most accurately described by what label?\n{sentence}
(9) The customer complaint and how it is addressed shall be provided to you. Classify the text in question as a primary category.\n{sentence}
(10) What label best describes the given text below?\n{sentence}

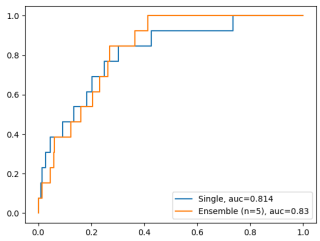
Table 4: Prompt designs for fine-tuning a LLM for prompt consistency approach.

A.2 Prompt Designs for LLM-based Prompt Consistency Approach

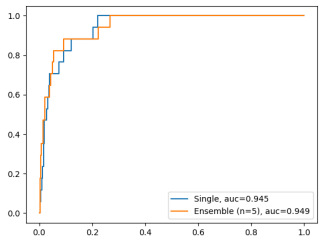
Table 4 describes the prompt designs we used for the prompt consistency approach.



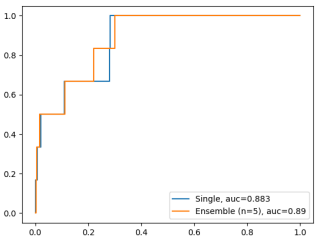
(a) Feb



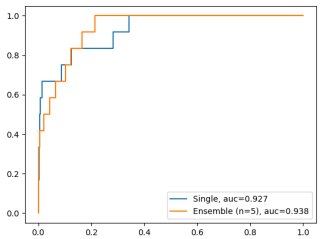
(b) Mar



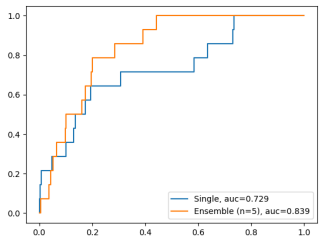
(c) Apr



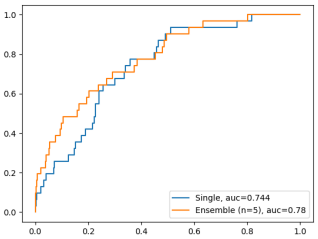
(d) May



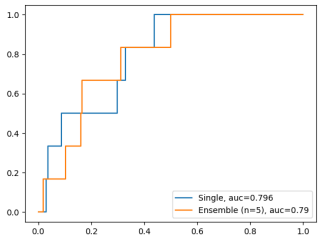
(e) Jun



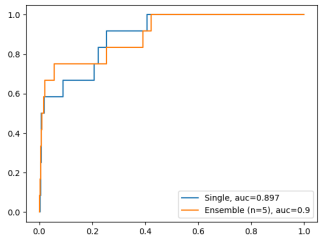
(f) Jul



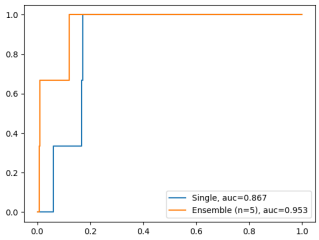
(g) Aug



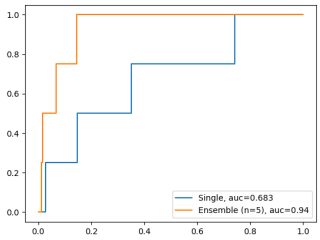
(h) Sep



(i) Oct



(j) Nov



(k) Dec

Figure 3: ROC curve of concept drift detection on ICD for each month.

Restaurant Review: The selection on the menu was great and so were the prices												
Words	**The**	**selection**	**on**	**the**	**menu**	**was**	**great**	**and**	**so**	**were**	**the**	**prices**
Sytenthic Contribution Score	0	0.264	0	0	0.413	0	0.129	0	0	0	0	0.194
Semantic Contribution Score	0.068	0.049	0.104	0.096	0.089	0.127	0.015	0.129	0.127	0.039	0.108	0.05
Fashion Review: Absolutely wonderful! silky and sexy and comfortable												
Words	**Absolutely**	**wonderful**	**!**	**silky**	**and**	**sexy**	**and**	**comfortable**				
Sytenthic Contribution Score	0	0	0	0.5	0	0.5	0	0				
Semantic Contribution Score	0.097	0.065	0.156	0.11	0.164	0.189	0.115	0.104				

Figure 4: Examples of the contribution scores on the review datasets.